IMA Journal of Numerical Analysis (2021) **00**, 1–31
https://doi.org/10.1093/imanum/drab066

Projection-free nonconvex stochastic optimization on Riemannian manifolds

Melanie Weber*
Princeton University, Princeton, NJ 08544, USA
*Corresponding author: mw25@math.princeton.edu

and

Suvrit Sra
LIDS, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
Email: suvrit@mit.edu

[Received on 26 March 2020; revised on 27 July 2021]

We study stochastic projection-free methods for constrained optimization of smooth functions on Riemannian manifolds, i.e., with additional constraints beyond the parameter domain being a manifold. Specifically, we introduce stochastic Riemannian Frank–Wolfe (Fw) methods for nonconvex and geodesically convex problems. We present algorithms for both purely stochastic optimization and finite-sum problems. For the latter, we develop variance-reduced methods, including a Riemannian adaptation of the recently proposed Spider technique. For all settings, we recover convergence rates that are comparable to the best-known rates for their Euclidean counterparts. Finally, we discuss applications to two classic tasks: the computation of the Karcher mean of positive definite matrices and Wasserstein barycenters for multivariate normal distributions. For both tasks, stochastic Fw methods yield state-of-the-art empirical performance.

Keywords: Riemannian optimization; Frank–Wolfe methods; nonconvex optimization; positive definite matrices; Karcher mean; Wasserstein barycenters.

1. Introduction

We study the following constrained (and possibly nonconvex) stochastic and finite-sum problems:

$$\min_{x\in\mathcal{X}\subset\mathcal{M}} \Phi(x) := \mathbb{E}_{\xi}[\phi(x,\xi)] = \int \phi(x,\xi)\ \mathrm{d}P(\xi), \tag{1.1}$$

$$\min_{x\in\mathcal{X}\subset\mathcal{M}} \Phi(x) := \frac{1}{m}\sum\nolimits_{i=1}^{m} \phi_i(x), \tag{1.2}$$

where $\mathcal{X}$ is compact and geodesically convex and $\mathcal{M}$ is a Riemannian manifold. Moreover, the component functions $\{\phi_i\}_{i=1}^m$ as well as Φ are (geodesically) Lipschitz smooth, but may be nonconvex. These problems greatly generalize their Euclidean counterparts (where $\mathcal{M} \equiv \mathbb{R}^d$), which themselves are of central importance in optimization and machine learning. In particular, finite-sum problems (Eq. 1.2) arise frequently in machine learning subroutines, such as empirical risk minimization, maximum likelihood estimation or the computation of M-estimators.

There has been an increasing interest in solving Riemannian problems of the above form, albeit without constraints (Bonnabel, 2013; Zhang & Sra, 2016; Zhang *et al.*, 2016; Tripuraneni *et al.*, 2018; Zhang *et al.*, 2018a; Kasai *et al.*, 2018b, 2019). This interest is driven by two key motivations: first,

that the exploitation of Riemannian geometry can deliver algorithms that are computationally superior to standard nonlinear programming approaches (Udriste, 1994; Absil *et al.*, 2008; Boumal *et al.*, 2014; Zhang *et al.*, 2016). Secondly, in many applications we encounter non-Euclidean data, such as graphs, strings, matrices and tensors, where using a forced Euclidean representation can be quite inefficient (Edelman *et al.*, 1998; Billera *et al.*, 2001; Zhang *et al.*, 2016; Nickel & Kiela, 2017; Sala *et al.*, 2018; Weber, 2020). These motivations have driven the recent surge of interest in the adaption and generalization of machine learning models and algorithms to Riemannian manifolds.

We solve problem (1.1) by introducing Riemannian stochastic Frank–Wolfe (Fw) algorithms. These methods are projection free (Frank & Wolfe, 1956), a property that has driven much of the recent interest in them (Jaggi, 2013). In contrast to projection-based methods, the Fw update requires solving a 'linear' optimization problem that ensures feasibility while often being much faster than projection. Fw has been intensively studied in Euclidean spaces for both convex (Jaggi, 2013; Lacoste-Julien & Jaggi, 2015) and nonconvex (Lacoste-Julien, 2016) objectives. Furthermore, stochastic variants have been proposed (Reddi *et al.*, 2016) that enable strong performance gains. As our experiments will show, our stochastic Riemannian Fw also delivers similarly strong performance gains on sample applications, outperforming the state-of-the-art.

1.1 *Summary of main contributions*

- We introduce three algorithms: (i) Stochastic Riemannian Frank–Wolfe (Srfw), a fully stochastic method that solves (1.1); (ii) Semi-stochastic variance-reduced Riemannian Frank–Wolfe (Svr-Rfw), a semi-stochastic variance-reduced version for (1.2); and (iii) Spider Riemannian Frank–Wolfe (Spider-Rfw), an improved variance-reduced variant that uses the recently proposed Spider technique for estimating the gradient. All three algorithms generalize various stochastic gradient tools to the Riemannian setting. For all methods, we establish convergence rates to first-order stationary points that match the rates of their Euclidean counterparts. Under the stronger assumption of geodesically convex objectives, we recover global sublinear convergence rates.
- In contrast to the study by Weber & Sra (2017), which considers Riemannian Fw, Stochastic Rfw does not require the computation of full gradients. Overcoming the need to compute the full gradient in each iteration greatly reduces the computational cost of each iteration as it removes a major bottleneck in Rfw. Moreover, Stochastic Rfw applies to problem 1.1, a crucial subroutine in many machine learning applications.
- We present an application to the computation of Riemannian centroids (Karcher mean) for positive definite matrices. This task is a well-known benchmark for Riemannian optimization and it arises, for instance, in statistical analysis, signal processing and computer vision. Notably, a simpler version of it also arises in the computation of hyperbolic embeddings.
- Furthermore, we present an application to the computation of Wasserstein barycenters for multivariate and *matrix-variate* Gaussians. For the latter, we prove the somewhat surprising property that the Wasserstein distance between two matrix-variate Gaussians is Euclidean convex. This result may be of independent interest.

The proposed Stochastic Rfw methods deliver valuable improvements, both in theory and experiment. Table 1 summarizes the complexity results for all variants in comparison with Rfw (Algorithm 1). For an analysis of Rfw 's complexity, see Weber & Sra (2017, Theorem 3). Our algorithms outperform state-of-the-art batch methods such as Riemannian LBFGS (Yuan *et al.*, 2016) and Zhang's majorization–minimization algorithm (Zhang, 2017). Moreover, we also observe performance gains over the

TABLE 1 *Oracle complexities of our Stochastic Riemannian Frank–Wolfe methods versus* RFW *(Weber & Sra, 2017) for nonconvex objectives. Note that we recover the best known rates of the Euclidean counterparts for each method. We consider three different oracle models, which we will define below in Section 2.4:* ***SFO/ IFO:*** *stochastic first-order oracle (for stochastic objectives) and incremental first-order oracle (for objectives with finite-sum form).* ***LO:*** *Riemannian linear optimization oracle*

Algorithm	RFW	SRFW	SVR-RFW	SPIDER-RFW
SFO/ IFO	$O\left(\frac{m}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon^4}\right)$	$O\left(m + \frac{m^{2/3}}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon^3}\right)$
RLO	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon^2}\right)$

deterministic RFW, which itself is known to be competitive against a wide range of Riemannian optimization tools (Weber & Sra, 2017). Importantly, our methods further outperform state-of-the-art stochastic Riemannian methods RSG (Kasai *et al.*, 2018b) and RSVRG (Zhang *et al.*, 2016; Sato *et al.*, 2017).

1.2 *Related work*

Riemannian optimization has recently witnessed a surge of interest (Bonnabel, 2013; Zhang & Sra, 2016; Huang *et al.*, 2018; Liu & Boumal, 2019). A comprehensive introduction to Riemannian optimization can be found in Absil *et al.* (2008). The MANOPT toolbox (Boumal *et al.*, 2014) implements many successful Riemannian optimization methods, serving as a benchmark.

The study of stochastic methods for Riemannian optimization has largely focused on projected-gradient methods. Bonnabel (2013) introduced the first Riemannian SGD. Zhang & Sra (2016) present a systematic study of first-order methods for geodesically convex problems, followed by a variance-reduced Riemannian SVRG (Zhang *et al.*, 2016; Sato *et al.*, 2017) that also applies to geodesically nonconvex functions. Kasai *et al.* (2018b) study gradient descent variants, as well as a Riemannian ADAM (Kasai *et al.*, 2019). A caveat of these methods is that a potentially costly projection is needed to ensure convergence. Otherwise, the strong (and often unrealistic) assumption that their iterates remain in a compact set is required In contrast, RFW (Algorithm 1) generates feasible iterates directly and therefore avoids the need to compute projections. This leads to a cleaner analysis and a more practical method in cases where the 'linear' oracle is efficiently implementable (Weber & Sra, 2017). We provide additional details on the comparison of projection-free and projection-based methods in Section 2.3. Riemannian optimization has also been applied in the ML literature, including for the computation of hyperbolic embeddings (Sala *et al.*, 2018), low-rank matrix and tensor factorization (Vandereycken, 2013) and eigenvector based methods (Journée *et al.*, 2010; Zhang *et al.*, 2016; Tripuraneni *et al.*, 2018).

2. Background and notation

We start by recalling some basic background on Riemannian geometry and introduce necessary notation. For a comprehensive overview on Riemannian geometry, see, e.g., Jost (2011).

2.1 *Riemannian manifolds*

A *manifold* $\mathcal{M}$ is a locally Euclidean space equipped with a differential structure. Its corresponding *tangent spaces* $T_x\mathcal{M}$ consist of tangent vectors at points $x \in \mathcal{M}$. We define an *exponential map*

Algorithm 1: Riemannian Frank-Wolfe (RFW)

1: Initialize $x_0 \in \mathcal{X} \subseteq \mathcal{M}$; assume access to the geodesic map $\gamma : [0,1] \to \mathcal{M}$
2: **for** $k = 0, 1, \ldots$ **do**
3: $\quad z_k \leftarrow argmin_{z \in \mathcal{X}} \langle \text{grad } \phi(x_k), \text{Exp}_{x_k}^{-1}(z) \rangle$
4: $\quad$ Let $\eta_k \leftarrow \frac{2}{k+2}$
5: $\quad x_{k+1} \leftarrow \gamma(\eta_k)$, where $\gamma(0) = x_k$ and $\gamma(1) = z_k$
6: **end for**

$\text{Exp} : T_x\mathcal{M} \to \mathcal{M}$ as follows: let $g_x \in T_x\mathcal{M}$; then $y = \text{Exp}_x(g_x) \in \mathcal{M}$ with respect to a geodesic $\gamma : [0,1] \mapsto \mathcal{M}$ with $\gamma(0) = x$, $\gamma(1) = y$ and $\dot{\gamma}(0) = g_x$. We will also use the *inverse* exponential map $\text{Exp}^{-1} : \mathcal{M} \to T_x\mathcal{M}$ that defines a diffeomorphism from the neighborhood of $x \in \mathcal{M}$ onto the neighborhood of $0 \in T_x\mathcal{M}$ with $\text{Exp}_x^{-1}(x) = 0$.

Riemannian manifolds are smooth manifolds with an inner product $\mathfrak{g}_x(u, v) = \langle u, v \rangle_x$ defined on $T_x\mathcal{M}$ for each $x \in \mathcal{M}$. The inner product gives rise to a norm $\|v\|_x := \sqrt{\mathfrak{g}_x(v, v)}$ for $v \in T_x\mathcal{M}$. We will further denote the geodesic distance of $x, y \in \mathcal{M}$ as $d(x, y)$. For comparing vectors of different tangent spaces, we use the following notion of *parallel transport*: let $x, y \in \mathcal{M}$, $x \neq y$. Then, the operator $\Gamma_x^y g_x$ maps $g_x \in T_x\mathcal{M}$ to the tangent space $T_y\mathcal{M}$ along a geodesic γ with $\gamma(0) = x$ and $\gamma(1) = y$. Note that the inner product on the tangent spaces is preserved under this mapping.

2.2 *Gradients, smoothness and convexity*

The *Riemannian gradient* $\text{grad } \phi(x)$ of a differentiable function $\phi : \mathcal{M} \to \mathbb{R}$ is defined as the unique vector in $T_x\mathcal{M}$ with directional derivative $D\phi(x)[v] = \langle \text{grad } \phi(x), v \rangle_x$ for all $v \in T_x\mathcal{M}$. For our algorithms we further need a notion of smoothness: let $\phi : \mathcal{M} \to \mathbb{R}$ be differentiable. We say that ϕ is *L-smooth*, if

$$\|\text{grad } \phi(y) - \Gamma_x^y \text{grad } \phi(x)\| \leq L d(x, y), \ \forall\, x, y \in \mathcal{M}, \tag{2.1}$$

or equivalently, if for all $x, y \in \mathcal{M}$, ϕ satisfies

$$\phi(y) \leqslant \phi(x) + \langle \text{grad } \phi(x), \text{Exp}_x^{-1}(y) \rangle_x + \tfrac{L}{2} d^2(x, y). \tag{2.2}$$

Another important property is *geodesic convexity* (short: g-convexity), which is defined as

$$\phi(y) \geqslant \phi(x) + \langle \text{grad } \phi(x), \text{Exp}_x^{-1}(y) \rangle_x \ \forall x, y \in \mathcal{M}. \tag{2.3}$$

2.3 *Projection-free vs. projection-based methods*

Classic Riemannian optimization has focused mostly on projection-based methods, such as *Riemannian Gradient Descent* (RGD) or *Riemannian Steepest Descent* (RSD) (Absil *et al.*, 2008). A convergence analysis of such methods typically assumes the gradient to be Lipschitz. However, the objectives typically considered in most optimization and machine learning tasks are not Lipschitz on the whole manifold. Hence, a compactness condition is required. Crucially, in projection-based methods, the

retraction back onto the manifold is typically not guaranteed to land in this compact set. Therefore, additional work (e.g., a projection step) is needed to ensure that the update remains in the compact region where the gradient is Lipschitz. On the other hand, Fw methods bypass this issue, because their update is guaranteed to stay within the compact feasible region. Further, for descent based methods it can suffice to ensure boundedness of the initial level set, but crucially, stochastic methods are *not* descent methods, and this argument does not apply. Finally, in some problems, the Riemannian 'linear' oracle can be much less expensive than computing a projection back onto the compact set. This is particularly significant for the applications highlighted in this paper, where the 'linear' oracle can even be solved in closed form.

2.4 *Oracle models*

We briefly review three oracle models, which are commonly used to understand the complexity of stochastic optimization algorithms.

1. *Stochastic first-order oracle* (short: *SFO*): Consider a stochastic function $\Phi(x) := \mathbb{E}[\phi(x, \xi)]$ with $\xi \sim \mathscr{P}$. For an input $x \in \mathscr{M}$, the SFO returns $(\phi(x, \xi'), \nabla\phi(x, \xi'))$ for a sample ξ' that is drawn i.i.d. from the distribution $\mathscr{P}$. For details, see Nemirovskiĭ & Yudin (1983).
2. *Incremental first-order oracle* (short: *IFO*): Consider a finite sum $\Phi(x) := \frac{1}{m}\sum_i \phi_i(x)$. For an input (i, x), where $i \in [n]$ is a function index and $x \in \mathscr{M}$, the IFO returns $(\phi_i(x), \nabla\phi_i(x))$. For details, see Agarwal & Bottou (2015).
3. *Riemannian linear optimization oracle* (short: *RLO*): For a set of constraints $\mathscr{X}$, a point $x \in \mathscr{X} \subseteq \mathscr{M}$ and a direction $g \in T_x\mathscr{M}$, the RLO returns $argmin_{z\in\mathscr{X}} \langle g, \operatorname{Exp}_x^{-1}(z)\rangle$.

Throughout the paper, we measure complexity as the number of SFO/ IFO and RLO calls made by the algorithm to obtain an ϵ-accurate solution.

3. Algorithms

In this section, we introduce three stochastic variants of RFW and analyze their convergence. Here and in the following x_k, x_{k+1} and y are as specified in Algorithm 2, 3 and 4, respectively. We further make the following assumptions: (1) Φ is L-smooth; and (2) in the stochastic case, the norm of the stochastic gradient is bounded as

$$\max_{\substack{x\in\mathscr{X}\\ \xi\in\operatorname{supp}(\mathscr{P})}} \|\operatorname{grad}\, \phi(x, \xi)\| \leqslant C$$

for some constant $C \geq 0$.

3.1 *Stochastic Riemannian Fw*

Our first method, SRFW (Algorithm 2), is a direct analog of stochastic Euclidean Fw. It has two key computational components: A stochastic gradient and a 'linear' oracle. Specifically, it requires access to the *stochastic* 'linear' oracle

$$y_k \leftarrow \operatorname*{argmin}_{y\in\mathscr{X}} \langle G(\xi, x_k), \operatorname{Exp}_{x_k}^{-1}(y)\rangle, \tag{3.1}$$

ALGORITHM 2 Stochastic Riemannian Frank-Wolfe (SRFW)

1: Initialize $x_0 \in \mathscr{X}$, assume access to the geodesic map $\gamma : [0, 1] \to \mathscr{M}$.

2: Set number of iterations K and minibatch sizes $\{b_k\}_{k=0}^{K-1}$.

3: **for** $k = 0, 1, \ldots K-1$ **do**

4: Sample i.i.d. $\{\xi_1, ..., \xi_{b_k}\}$ uniformly at random according to $\mathscr{P}$.

5: $y_k \leftarrow argmin_{y \in \mathscr{X}} \langle \frac{1}{b_k} \sum_{i=1}^{b_k} \text{grad } \phi(x_k, \xi_i), \text{Exp}_{x_k}^{-1}(y) \rangle$

6: Compute step size η_k and set $x_{k+1} \leftarrow \gamma(\eta_k)$, where $\gamma(0) = x_k$ and $\gamma(1) = y_k$.

7: $x^k \leftarrow x_k$

8: **end for**

9: Output $\hat{x}$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$.

where $G(\cdot, \cdot)$ is an unbiased estimator of the Riemannian gradient ($\mathbb{E}_\xi G(\xi, x) = \text{grad } \Phi(x)$). In contrast to Euclidean Fw, the oracle (3.1) involves solving a nonlinear, nonconvex optimization problem. Whenever this problem is efficiently solvable, we can benefit from the FW strategy. In Weber & Sra (2017), we analyze two instances where Eq. 3.1 can be solved in closed form, for positive definite matrices and for the special orthogonal group, respectively. Our experiments below will provide two concrete examples for the case of positive definite matrices.

We consider a minibatch variant of the oracle (3.1), namely

$$y_k \leftarrow \operatorname*{argmin}_{y \in \mathscr{X}} \left\langle \frac{1}{b_k} \sum_{i=1}^{b_k} \text{grad } \phi(x_k, \xi_i), \text{Exp}_{x_k}^{-1}(y) \right\rangle,$$

where $\xi_i \sim \mathscr{P}$ are drawn i.i.d., and thus the minibatch gradient is also unbiased. We first evaluate the goodness of this minibatch gradient approximation with the following (standard) lemma:

LEMMA 3.1 (Goodness of stochastic gradient estimate). Let $\Phi(x) = \mathbb{E}_{\boldsymbol{\xi}} \left[\phi(x, \xi_i) \right]$ with random variables $\{\xi_i\}_{i=1}^b = \boldsymbol{\xi} \sim \mathscr{P}$. Furthermore, let $g(x) := \frac{1}{b} \sum_{i=1}^b \text{grad } \phi(x, \xi_i)$ denote the gradient estimate from a batch $\boldsymbol{\xi}$. Assume that the norm of the gradient estimate is upper-bounded as $\max_{x \in \mathscr{X}, \xi \in \text{supp}(\mathscr{P})} \|\text{grad } \phi(x, \xi)\| \leqslant C$. Then, $\mathbb{E}_{\boldsymbol{\xi}} \left[\|g(x) - \text{grad } \Phi(x)\| \right] \leqslant \frac{C}{\sqrt{b}}$.

In the following, we drop the subscript $\boldsymbol{\xi}$ from the expectation for ease of notation, whenever its meaning is clear from context.

For the proof, recall the following fact, which we will use throughout the paper:

REMARK 3.1 For a set of n independent random variables $\{v_i\}_{1 \leq i \leq n}$ with mean zero, we have

$$\mathbb{E}\big[\|v_1 + \cdots + v_n\|^2\big] = \mathbb{E}\big[\|v_1\|^2 + \cdots + \|v_n\|^2\big]. \tag{3.2}$$

Proof. We have

$$\mathbb{E}\left[\|g(x) - \underbrace{\text{grad } \Phi(x)}_{=\mathbb{E}[g(x)]}\|^2\right] = \mathbb{E}\big[\|g(x)\|^2\big] - \underbrace{\|\mathbb{E}\,[g(x)]\|^2}_{\geqslant 0} \leqslant \mathbb{E}\big[\|g(x)\|^2\big]$$

$$= \mathbb{E}\left[\left\|\frac{1}{b}\sum_{i=1}^{b} \text{grad}\,\phi(x,\xi_i)\right\|^2\right] \overset{(1)}{\leqslant} \frac{1}{b^2}\mathbb{E}\left[\sum_{i=1}^{b}\underbrace{\|\text{grad}\,\phi(x,\xi_i)\|^2}_{\leqslant C^2}\right] \overset{(2)}{\leqslant} \frac{C^2}{b},$$

where (1) follows from Remark 3.1 and the fact that $\mathbb{E}\left[g(x) - \text{grad } \Phi(x)\right] = 0$, since $g(x)$ is assumed to be an unbiased gradient estimate; and (2) follows from the assumption that the norm of the gradient is upper-bounded by C. Furthermore, with Jensen's inequality:

$$\mathbb{E}\big[\|g(x) - \text{grad } \Phi(x)\|^2\big] \geqslant \big[\mathbb{E}\,(\|g(x) - \text{grad}\,\Phi(x)\|)\big]^2.$$

Putting both together and taking the square root on both sides gives the desired claim:

$$\mathbb{E}\left[\|g(x) - \text{grad } \Phi(x)\|\right] \leqslant \frac{C}{\sqrt{b}}.$$

□

With this characterization of the approximation error, we can perform a convergence analysis for both nonconvex and g-convex objectives. To evaluate convergence rates, consider the following criterion (*Fw gap*):

$$\mathcal{G}(x) = \max_{y\in\mathcal{X}} \langle \text{Exp}_x^{-1}(y),\, -\text{grad}\,\Phi(x)\rangle. \tag{3.3}$$

A similar criterion is used in theoretical analysis of Euclidean Fw methods (see, e.g., Reddi *et al.* (2016)). We define the *Stochastic Fw gap* as

$$\hat{\mathcal{G}}(x) = \max_{y\in\mathcal{X}} \langle \text{Exp}_x^{-1}(y),\, -g(x)\rangle.$$

Assuming that the Robbins–Monroe approximation $g(x)$ gives an unbiased estimate of the gradient $\text{grad}\,\Phi(x)$ (*), we have (by Jensen's inequality and the convexity of the *max*-function):

$$\mathbb{E}\big[\hat{\mathcal{G}}(x)\big] \geq \max_{y\in\mathcal{X}} \langle \text{Exp}_x^{-1}(y),\, -\mathbb{E}\,[g(x)]\rangle \overset{(*)}{=} \max_{y\in\mathcal{X}} \langle \text{Exp}_x^{-1}(y),\, -\text{grad}\,\Phi(x)\rangle = \mathcal{G}(x).$$

With this, we can show that SRFW converges at a sublinear rate to first-order stationary points:

THEOREM 3.1 (Convergence SRFW). With constant steps size $\eta_k = \frac{1}{\sqrt{K}}$ and constant batch sizes $b_k = K$, Algorithm 2 converges in expectation with a *sublinear* rate, i.e.,

$$\mathbb{E}_{\xi}\left[\mathscr{G}(\hat{x})\right] = O(1/\sqrt{K}).$$

To prove the theorem, we need a few additional auxiliary results. First, recall the definition of the *curvature constant* M_{Φ}, introduced in Weber & Sra (2017):

DEFINITION 3.2 (Curvature constant). Let $x, y, z \in \mathscr{X}$ and $\gamma : [0,1] \to \mathscr{M}$ a geodesic map with $\gamma(0) = x$, $\gamma(1) = z$ and $y = \gamma(\eta)$ for $\eta \in [0,1]$. Define

$$M_{\Phi} := \sup_{\substack{x,y,z \in \mathscr{X} \\ y=\gamma(\eta)}} \frac{2}{\eta^2}\big[\Phi(y) - \Phi(x) - \langle \operatorname{grad}\Phi(x), \operatorname{Exp}_x^{-1}(y)\rangle\big]. \tag{3.4}$$

We further recall two technical lemmas on M_{Φ}; the proofs can be found in Weber & Sra (2017):

LEMMA 3.3 (Weber & Sra (2017)). Let $\Phi : \mathscr{M} \to \mathbb{R}$ be L-smooth on $\mathscr{X}$; let $\operatorname{diam}(\mathscr{X}) := \sup_{x,y\in\mathscr{X}} \mathrm{d}(x,y)$. Then, the curvature constant M_{ϕ} satisfies the bound $M_{\Phi} \leq L \operatorname{diam}(\mathscr{X})^2$.

LEMMA 3.4 (Weber & Sra (2017)). Let $\mathscr{X}$ be a constrained set. There exists a constant $M_{\Phi} \geqslant 0$ such that for $x_k, x_{k+1}, y_k \in \mathscr{X}$ as specified in Algorithm 2, and for $\eta_k \in (0,1)$

$$\Phi(x_{k+1}) \leqslant \Phi(x_k) + \eta_k \langle \operatorname{grad}\ \Phi(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \tfrac{1}{2} M_{\Phi} \eta_k^2.$$

With this, we can now prove Theorem 3.1:

Proof. (Theorem 3.1) Let again

$$g_k(x_k) := \frac{1}{b_k} \sum_{i=1}^{b_k} \operatorname{grad}\ \phi(x_k, \xi_i) \tag{3.5}$$

denote the gradient estimate from the k^{th} batch. Then

$$\Phi(x_{k+1}) \overset{(1)}{\leqslant} \Phi(x_k) + \eta_k \langle \operatorname{grad}\ \Phi(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \frac{1}{2} M_{\Phi} \eta_k^2. \tag{3.6}$$

$$\overset{(2)}{\leqslant} \Phi(x_k) + \eta_k \langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \eta_k \langle \operatorname{grad}\ \Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \frac{1}{2} M_{\Phi} \eta_k^2 \tag{3.7}$$

Here, (1) follows from Lemma 3.4 and (2) follows from 'adding a zero' with respect to g_k. We then apply the Cauchy–Schwartz inequality to the inner product and make use of the fact that the geodesic distance between points in $\mathscr{X}$ is bounded by its diameter:

$$\left\langle \operatorname{grad}\ \Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\right\rangle \leqslant \|\operatorname{grad}\ \Phi(x_k) - g_k(x_k)\| \cdot \underbrace{\left\|\operatorname{Exp}_{x_k}^{-1}(y_k)\right\|}_{\leqslant \operatorname{diam}(\mathscr{X})}. \tag{3.8}$$

This gives (with $D := \operatorname{diam}(\mathscr{X})$)

$$\Phi(x_{k+1}) \leqslant \Phi(x_k) + \eta_k \langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \eta_k D \|\operatorname{grad}\ \Phi(x_k) - g_k(x_k)\| + \frac{1}{2} M_\Phi \eta_k^2.$$

Taking expectations and applying Lemma 3.1 to the third term on the right-hand side, we get

$$\mathbb{E}\left[\Phi(x_{k+1)}\right] \leqslant \mathbb{E}\left[\Phi(x_k)\right] - \eta_k \mathbb{E}\left[\hat{\mathscr{G}}(x_k)\right] + \eta_k D \frac{C}{\sqrt{b_k}} + \frac{1}{2} M_\Phi \eta_k^2,$$

where we have rewritten the second term in terms of the stochastic Fw gap

$$\mathbb{E}\left[\hat{\mathscr{G}}(x_k)\right] = -\mathbb{E}\left[\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle\right].$$

Summing over all k batches, telescoping and reordering terms gives

$$\sum_k \eta_k \mathbb{E}\left[\hat{\mathscr{G}}(x_k)\right] \leqslant \mathbb{E}\left[\Phi(x_0)\right] - \mathbb{E}\left[\Phi(x_K)\right] + \sum_k \eta_k D \frac{C}{\sqrt{b_k}} + \sum_k \frac{1}{2} M_\Phi \eta_k^2 \tag{3.9}$$

$$\leqslant \left(\Phi(x_0) - \Phi(x_K)\right) + \sum_k \eta_k D \frac{C}{\sqrt{b_k}} + \sum_k \frac{1}{2} M_\Phi \eta_k^2. \tag{3.10}$$

From Algorithm 2 we see that the output $\hat{x}$ is chosen uniformly at random from $\{x_1, ..., x_K\}$, i.e., $\mathbb{E}\left[\mathbb{E}\left[\hat{\mathscr{G}}(x_k)\right]\right] = \mathbb{E}\left[\mathscr{G}(\hat{x})\right]$, where we have used that, by construction, $\mathbb{E}\left[\hat{\mathscr{G}}(x)\right] = \mathscr{G}(x)$. Now, with constant step sizes $\eta_k = \eta$ and batch sizes $b_k = b$, we have

$$K\eta \mathbb{E}\left[\mathscr{G}(\hat{x})\right] \leqslant \left(\Phi(x_0) - \Phi(x_K)\right) + K\eta D \frac{C}{\sqrt{b}} + K \frac{1}{2} M_\Phi \eta^2.$$

Now, let $C_{x_0} > 0$ be an initialization-dependent constant, such that $C_{x_0} > \Phi(x_0) - \mathbb{E}\left[\Phi(x^\star)\right]$, where $x^\star$ is a first-order stationary point. From $\eta = \frac{1}{\sqrt{K}}$ and $b = K$ we see that

$$\mathbb{E}\left[\mathscr{G}(\hat{x})\right] \leqslant \frac{1}{\sqrt{K}} \left(C_{x_0} + DC + \frac{1}{2} M_\Phi\right),$$

which shows the desired sublinear convergence rate. □

COROLLARY 3.1 SRFW obtains an ϵ-accurate solution with SFO complexity of $O\left(\frac{1}{\epsilon^4}\right)$ and RLO complexity of $O\left(\frac{1}{\epsilon^2}\right)$.

Proof. It follows directly from Theorem 3.1 that SRFW achieves an ϵ-accurate solution after $O\left(\frac{1}{\epsilon^2}\right)$ iteration, i.e., its RLO complexity is $O\left(\frac{1}{\epsilon^2}\right)$. For the SFO complexity, note that

$$\sum_{k=0}^{K-1} b_k = Kb = K^2 \lesssim O\left(\frac{1}{\epsilon^4}\right).$$

□

For g-convex objectives, we can obtain a global convergence result in terms of the optimality gap $\Delta_k := \Phi(x_k) - \Phi(x^*)$. Here, SRFW converges at a sublinear rate to the global optimum $\Phi(x^*)$.

COROLLARY 3.2 If Φ is g-convex, then under the assumptions of Theorem 3.1 the optimality gap converges as $\mathbb{E}_{\boldsymbol{\xi}}\left[\Delta_k\right] = O(1/\sqrt{K})$.

Proof. In the proof of Theorem 3.1, Eq. 3.6, note that

$$\begin{aligned}\Phi(x_{k+1}) &\leqslant \Phi(x_k) + \eta_k\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \eta_k\langle \operatorname{grad}\Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \frac{1}{2}M_\Phi\eta_k^2\\ &\overset{(1)}{\leqslant} \Phi(x_k) + \eta_k\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(x^*)\rangle + \eta_k\langle \operatorname{grad}\Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \frac{1}{2}M_\Phi\eta_k^2,\end{aligned}$$

where (1) follows from y_k being the argmin as defined in Algorithm 2. Note that in the third term, the Cauchy–Schwartz inequality gives

$$\langle \operatorname{grad}\Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle \leqslant \|\operatorname{grad}\Phi(x_k) - g_k(x_k)\| \underbrace{\|\operatorname{Exp}_{x_k}^{-1}(y_k)\|}_{\leq \operatorname{diam}(\mathcal{X}) =: D}.$$

Inserting this above and taking expectations, we have

$$\begin{aligned}\mathbb{E}\left[\Phi(x_{k+1})\right] &\leqslant \mathbb{E}\left[\Phi(x_k)\right] + \eta_k\mathbb{E}\left[\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(x^*)\rangle\right] + \eta_k D\underbrace{\mathbb{E}\left[\|\operatorname{grad}\Phi(x_k) - g_k(x_k)\|\right]}_{\leq \frac{C}{\sqrt{b_k}}} + \frac{1}{2}M_\Phi\eta_k^2\\ &\overset{(2)}{\leqslant} \mathbb{E}\left[\Phi(x_k)\right] + \eta_k\mathbb{E}\left[\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(x^*)\rangle\right] + \eta_k D\frac{C}{\sqrt{b_k}} + \frac{1}{2}M_\Phi\eta_k^2,\end{aligned}$$

where (2) follows from Lemma 3.1. For the second term, we have

$$\mathbb{E}\left[\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(x^*)\rangle\right] = \left\langle \mathbb{E}\left[g_k(x_k)\right], \operatorname{Exp}_{x_k}^{-1}(x^*)\right\rangle \overset{(3)}{=} \langle \operatorname{grad}\Phi(x_k), \operatorname{Exp}_{x_k}^{-1}(x^*)\rangle \overset{(4)}{\leqslant} -\left(\Phi(x_k) - \Phi(x^*)\right),$$

since (3) $g_k(x_k)$ is an unbiased estimate of grad $\Phi(x_k)$ and (4) the Fw gap upper-bounds the optimality gap, which is a direct consequence of the g-convexity of Φ (see Eq. 2.3). Let $\Delta_k := \Phi(x_k) - \Phi(x^*)$

denote the optimality gap. Then, putting everything together and rewording terms, we get

$$\eta_k \mathbb{E}\left[\Delta_k\right] \leqslant \mathbb{E}\left[\Phi(x_k) - \Phi(x_{k+1})\right] + \eta_k D \frac{C}{\sqrt{b_k}} + \frac{1}{2} M_\Phi \eta_k^2.$$

Summing, telescoping and inserting the definition of the output ($\hat{x}$ with optimality gap $\Delta_{\hat{k}} = \Phi(\hat{x}) - \Phi(x^*)$), we have

$$\mathbb{E}\left[\Delta_{\hat{k}}\right]\left(\sum_k \eta_k\right) \leqslant \left(\Phi(x_0) - \Phi(x_K)\right) + DC \sum_k \frac{\eta_k}{\sqrt{b_k}} + \frac{1}{2}\sum_k \eta_k^2.$$

With the parameter choice $\eta_k = \eta = \frac{1}{\sqrt{K}}$ and $b_k = b = K$, the claim follows as

$$\mathbb{E}\left[\Delta_{\hat{k}}\right] \leqslant \frac{1}{\sqrt{K}}\left(\Delta_{x_0} + DC + \frac{1}{2} M_\Phi\right),$$

where Δ_{x_0} denotes the initial optimality gap, which is a constant whose value depends on the initialization only. □

A shortcoming of SRFW is its large batch sizes. We expect that choosing a non-constant, decreasing step size will reduce the required batch size.

3.2 *Stochastic variance-reduced Fw*

In addition to the purely stochastic SRFW method, we can obtain a stochastic FW algorithm via a (semi-stochastic) *variance-reduced* approach for problems with a *finite-sum structure* (1.2). Recall, that in problem (1.2), we assume that the cost function Φ can be represented as a finite sum $\Phi(x) = \frac{1}{m}\sum_{i=1}^m \phi_i(x)$, where the ϕ_i are L-smooth (but may be nonconvex). We will see that by exploiting the finite-sum structure, we can obtain provably faster FW algorithms.

We first propose SVR-RFW (Algorithm 3), which combines RFW with a classic variance-reduced estimate of the gradient. This resulting algorithm computes the full gradient at the beginning of each epoch and uses batch estimates within epochs. The variance-reduced gradient estimate guarantees the following bound on the approximation error:

LEMMA 3.5 (Goodness of variance-reduced gradient estimate). Consider the kth iteration in the sth epoch and the *stochastic variance-reduced gradient estimate* with respect to a minibatch $I_k = \left(i_1, \ldots, i_{b_k}\right)$

$$g_k\left(x_k^{s+1}\right) = \frac{1}{b_k}\sum_{j=i_1,\ldots,i_{b_k}} \operatorname{grad}\phi_j\left(x_k^{s+1}\right) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}}\left(\operatorname{grad}\phi_j(\tilde{x}^s) - \operatorname{grad}\Phi(\tilde{x}^s)\right),$$

with the $\{\phi_i\}$ assumed to be L-Lipschitz. Then the expected deviation of the estimate g_k from the true gradient $\operatorname{grad}\Phi$ is bounded as

$$\mathbb{E}_{I_k}\left[\left\|\operatorname{grad}\Phi\left(x_k^{s+1}\right) - g_k\left(x_k^{s+1}\right)\right\|\right] \leqslant \frac{L}{\sqrt{b_k}} d\left(x_k^{s+1}, \tilde{x}^s\right).$$

ALGORITHM 3 Semi-stochastic variance-reduced Riemannian Frank-Wolfe (SVR-RFW)

Initialize $\tilde{x}^0 \in \mathscr{X}$; assume access to the geodesic map $\gamma : [0,1] \to \mathscr{M}$.
Choose number of iterations S and size of epochs K and set minibatch sizes $\{b_k\}_{k=0}^{K-1}$.
for $s = 0, \dots S-1$ **do**
  Compute gradient at $\tilde{x}^s$: $\operatorname{grad} \Phi(\tilde{x}^s) = \frac{1}{N} \sum_{i=1}^{m} \operatorname{grad} \phi_i(\tilde{x}^s)$.
  for $k = 1, \dots K$ **do**
    Sample i.i.d. $I_k := \left(i_1, ..., i_{b_k}\right) \subseteq [m]$ (minibatches).
    $z_{k+1}^{s+1} \leftarrow \operatorname{argmin}_{z \in \mathscr{X}} \langle \frac{1}{b_k} \sum_{j=i_1,...,i_{b_k}} \operatorname{grad} \phi_j(x_k^{s+1}) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}} \left(\operatorname{grad} \phi_j(\tilde{x}^s) - \operatorname{grad} \Phi(\tilde{x}^s)\right), \operatorname{Exp}_{\tilde{x}^s}^{-1}(z) \rangle$
    Compute step size η_k and set $x_{k+1}^{s+1} \leftarrow \gamma(\eta_k)$, where $\gamma(0) = x_k^{s+1}$ and $\gamma(1) = z_{k+1}^{s+1}$.
  end for
  $\tilde{x}^{s+1} = x_K^s$.
end for
Output $\hat{x} = \tilde{x}_K^S$.

We again drop the subscript I_k, whenever it is clear from context.

Proof. Following Algorithm 3, let $I_k = \left(i_1, \dots, i_{b_k}\right)$ denote the sample in the kth iteration of the sth epoch. We introduce the shorthands

$$\zeta_k^{s+1} = \frac{1}{b_k} \sum_{l=1}^{b_k} \operatorname{grad} \phi_{i_l}\left(x_k^{s+1}\right) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}} \operatorname{grad} \phi_{i_l}(\tilde{x}^s)$$

$$\zeta_{k,i_l}^{s+1} = \operatorname{grad} \phi_{i_l}\left(x_k^{s+1}\right) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}} \operatorname{grad} \phi_{i_l}(\tilde{x}^s),$$

i.e., $\zeta_k^{s+1} = \frac{1}{b_k} \sum_{l=1}^{b_k} \zeta_{k,i_l}^{s+1}$. Then we have

$$\mathbb{E}\left[\left\|\operatorname{grad} \Phi\left(x_k^{s+1}\right) - g_k\left(x_k^{s+1}\right)\right\|^2\right] = \mathbb{E}\left[\left\|\zeta_k^{s+1} - \operatorname{grad} \Phi\left(x_k^{s+1}\right) + \Gamma_{\tilde{x}^s}^{x_k^{s+1}} \operatorname{grad} \Phi(\tilde{x}^s)\right\|^2\right]$$

$$\overset{(1)}{=} \mathbb{E}\left[\left\|\zeta_k^{s+1} - \mathbb{E}\left(\zeta_k^{s+1}\right)\right\|^2\right].$$

Here, (1) follows from the following argument:

$$\begin{aligned}\operatorname{grad}\Phi\left(x_k^{s+1}\right) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}} \operatorname{grad}\Phi(\tilde{x}^s) &\overset{(2)}{=} \mathbb{E}\left[\frac{1}{b_k}\sum_l \operatorname{grad}\phi_{i_l}\left(x_k^{s+1}\right) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}} \operatorname{grad}\phi_{i_l}(\tilde{x}^s)\right] \\ &= \mathbb{E}\left[\frac{1}{b_k}\sum_l \zeta_{k,i_l}^{s+1}\right] \\ &= \mathbb{E}\left(\zeta_k^{s+1}\right),\end{aligned}$$

where in (2) we used the assumption that the variance-reduced gradient is an unbiased estimate of the full Riemannian gradient. We further have

$$\begin{aligned}\mathbb{E}\Big[\Big\|\zeta_k^{s+1} - \mathbb{E}\big[\zeta_k^{s+1}\big]\Big\|^2\Big] &= \mathbb{E}\big[\|\zeta_k^{s+1}\|^2\big] - \underbrace{\Big\|\mathbb{E}\big[\zeta_k^{s+1}\big]\Big\|^2}_{\geqslant 0} \leqslant \mathbb{E}\Big[\|\zeta_k^{s+1}\|^2\Big] \\ &= \mathbb{E}\left[\Big\|\frac{1}{b_k}\sum_l \zeta_{k,i_l}^{s+1}\Big\|^2\right] \overset{(3)}{\leqslant} \frac{1}{b_k^2}\mathbb{E}\left[\sum_l \Big\|\zeta_{k,i_l}^{s+1}\Big\|^2\right] \\ &= \frac{1}{b_k^2}\mathbb{E}\left[\sum_l \underbrace{\Big\|\operatorname{grad}\ \phi_{i_l}\left(x_k^{s+1}\right) - \Gamma_{\tilde{x}_s}^{x_k^{s+1}} \operatorname{grad}\ \phi_{i_l}(\tilde{x}^s)\Big\|^2}_{\leqslant L d(x_k^{s+1},\tilde{x}^s)}\right] \\ &\overset{(4)}{\leqslant} \frac{b_k L^2 d^2\left(x_k^{s+1},\tilde{x}^s\right)}{b_k^2},\end{aligned}$$

where (3) follows from Remark 3.1 and (4) from the assumption that the ϕ_i are L-Lipschitz smooth. This shows

$$\mathbb{E}\Big[\Big\|\operatorname{grad}\Phi\left(x_k^{s+1}\right) - g_k\left(x_k^{s+1}\right)\Big\|^2\Big] \leq \frac{L^2}{b_k} d^2\left(x_k^{s+1},\tilde{x}^s\right).$$

Jensen's inequality gives

$$\mathbb{E}\Big[\Big\|\operatorname{grad}\Phi\left(x_k^{s+1}\right) - g_k\left(x_k^{s+1}\right)\Big\|^2\Big] \geq \mathbb{E}\Big[\Big\|\operatorname{grad}\Phi\left(x_k^{s+1}\right) - g_k\left(x_k^{s+1}\right)\Big\|\Big]^2,$$

and, putting everything together and taking the square root on both sides, the claim follows as

$$\mathbb{E}\Big[\Big\|\operatorname{grad}\Phi\left(x_k^{s+1}\right) - g_k\left(x_k^{s+1}\right)\Big\|\Big] \leq \frac{L}{\sqrt{b_k}} d\left(x_k^{s+1},\tilde{x}^s\right).$$

□

Using Lemma 3.5 we can recover the following sublinear convergence rate:

THEOREM 3.6 With steps size $\eta_k = \frac{1}{\sqrt{KS}}$ and constant batch sizes $b_k = K^2$, Algorithm 3 converges in expectation with $\mathbb{E}_{I_k}\left[\mathscr{G}(\hat{x})\right] = O\left(\frac{1}{\sqrt{KS}}\right)$. Here, $\mathscr{G}(x)$ again denotes the Fw gap as defined in Eq. 3.3.

Proof. (Theorem 3.6) Let again

$$g_k(x_k^{s+1}) = \frac{1}{b_k}\sum_j \operatorname{grad}\phi_j(x_k^{s+1}) - \Gamma_{\tilde{x}^s}^{x_k^{s+1}}\left(\operatorname{grad}\phi_j(\tilde{x}^s) - \operatorname{grad}\Phi(\tilde{x}^s)\right) \tag{3.11}$$

denote the variance-reduced gradient estimate in the kth iteration of the sth epoch. Then

$$\Phi(x_{k+1}^{s+1}) \overset{(1)}{\leqslant} \Phi(x_k^{s+1}) + \eta_k\langle\operatorname{grad}\ \Phi(x_k^{s+1}), \operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\rangle + \frac{1}{2}M_\Phi\eta_k^2 \tag{3.12}$$

$$\overset{(2)}{\leqslant} \Phi(x_k^{s+1}) + \eta_k\langle g_k(x_k^{s+1}), \operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\rangle \tag{3.13}$$

$$+ \eta_k\langle\operatorname{grad}\ \Phi(x_k^{s+1}) - g_k(x_k^{s+1}), \operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\rangle + \frac{1}{2}M_\Phi\eta_k^2. \tag{3.14}$$

Here, (1) follows from Lemma 3.4 and (2) follows from 'adding a zero' with respect to g_k. We then apply Cauchy–Schwarz to the inner product and make use of the fact that the geodesic distance between points in $\mathscr{X}$ is bounded by its diameter:

$$\langle\operatorname{grad}\ \Phi(x_k^{s+1}) - g_k(x_k^{s+1}), \operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\rangle \leqslant \|\operatorname{grad}\ \Phi(x_k^{s+1}) - g_k(x_k^{s+1})\| \cdot \underbrace{\|\operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\|}_{\leqslant \operatorname{diam}(\mathscr{X})}. \tag{3.15}$$

This gives (with $D := \operatorname{diam}(\mathscr{X})$)

$$\Phi(x_{k+1}^{s+1}) \leqslant \Phi(x_k^{s+1}) + \eta_k\langle g_k(x_k^{s+1}), \operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\rangle + \eta_k D\|\operatorname{grad}\ \Phi(x_k^{s+1}) - g_k(x_k^{s+1})\| + \frac{1}{2}M_\Phi\eta_k^2.$$

Taking expectations, we have

$$\mathbb{E}\left[\Phi(x_{k+1}^{s+1})\right] \leq \mathbb{E}\left[\Phi\left(x_k^{s+1}\right)\right] + \eta_k\mathbb{E}\left[\left\langle g_k\left(x_k^{s+1}\right), \operatorname{Exp}_{x_k^{s+1}}^{-1}(y_k)\right\rangle\right]$$

$$+ \eta_k D\mathbb{E}\left[\|\operatorname{grad}\ \Phi(x_k^{s+1}) - g_k(x_k^{s+1})\|\right] + \frac{1}{2}M_\Phi\eta_k^2 \tag{3.16}$$

$$\overset{(3)}{\leq} \mathbb{E}\left[\Phi(x_k^{s+1})\right] - \eta_k\mathbb{E}\left[\hat{\mathscr{G}}(x_k^{s+1})\right] + \eta_k D\frac{L}{\sqrt{b_k}}\mathbb{E}\left[d(x_k^{s+1}, \tilde{x}^s)\right] + \frac{1}{2}M_\Phi\eta_k^2, \tag{3.17}$$

where (3) follows from applying the definition of the stochastic Fw gap to the second term and Lemma 3.5 to the third term.

For the following analysis, define for $k = 1, \dots, K$ and a fixed epoch $s \in [S]$

$$R_k := \mathbb{E}\big[\Phi(x_k^{s+1}) + c_k d(x_k^{s+1}, \tilde{x}^s)\big] \tag{3.18}$$

$$c_k = c_{k+1} + \eta_k D \frac{L}{\sqrt{b_k}} \qquad (c_K = 0). \tag{3.19}$$

With that and inequality 3.16, we have

$$\begin{aligned}
R_{k+1} &= \mathbb{E}\big[\Phi\big(x_{k+1}^{s+1}\big)\big] + c_{k+1}\mathbb{E}\big[d\big(x_{k+1}^{s+1}, \tilde{x}^s\big)\big] \\
&\leqslant \mathbb{E}\big[\Phi\big(x_k^{s+1}\big)\big] - \eta_k \mathbb{E}\big[\hat{\mathscr{G}}\big(x_k^{s+1}\big)\big] + \eta_k D \frac{L}{\sqrt{b_k}} \mathbb{E}\big[d\big(x_k^{s+1}, \tilde{x}^s\big)\big] + \frac{1}{2} M_\Phi \eta_k^2 + c_{k+1} \underbrace{\mathbb{E}\big[d\big(x_{k+1}^{s+1}, \tilde{x}^s\big)\big]}_{\overset{(4)}{\leqslant} \mathbb{E}\left[d(x_{k+1}^{s+1}, x_k^{s+1}) + d(x_k^{s+1}, \tilde{x}^s)\right]} \\
&\leqslant \underbrace{\left(\mathbb{E}\big[\Phi\big(x_k^{s+1}\big)\big] + \underbrace{\left(c_{k+1} + \eta_k D \frac{L}{\sqrt{b_k}}\right)}_{=c_k} \mathbb{E}\big[d\big(x_k^{s+1}, \tilde{x}^s\big)\big] \right)}_{=R_k} - \eta_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + c_{k+1}\mathbb{E}\big[d(x_{k+1}^{s+1}, x_k^{s+1})\big] \\
&\quad + \frac{1}{2} M_\Phi \eta_k^2 \\
&\overset{(5)}{\leqslant} R_k - \eta_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + c_{k+1}\eta_k D + \frac{1}{2} M_\Phi \eta_k^2,
\end{aligned}$$

where (4) follows by adding a zero and applying the triangle inequality and (5) follows from the definition of R_k and the definition of the update step via the geodesic map γ (see Algorithm 3)

$$\mathbb{E}\big[d(x_{k+1}^{s+1}, x_k^{s+1})\big] \leq \eta_k \mathbb{E}\big[\|\mathrm{Exp}_{x_k}^{-1}(z_k)\|\big] \leq \eta_k D. \tag{3.20}$$

Telescoping within the epoch $s+1$ we get (with $\eta_k = \eta$ and $b_k = b$ for $k = 0, \dots, K-1$)

$$\begin{aligned}
R_K &\leq R_0 - \sum_k \eta_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + \frac{1}{2} M_\Phi \sum_k \eta_k^2 + D \sum_k \eta_k c_{k+1} \\
&= R_0 - \eta \sum_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + \frac{1}{2} M_\Phi \eta^2 K + D\eta \sum_k c_{k+1} \\
&= R_0 - \eta \sum_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + \frac{1}{2} M_\Phi \eta^2 K + \frac{\eta^2 D^2 L}{\sqrt{b}} \frac{K(K-1)}{2}.
\end{aligned}$$

This gives

$$\mathbb{E}\big[\Phi(x_K^{s+1})\big] \leqslant \mathbb{E}\left[\Phi(x_K^s)\right] - \eta \sum_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + \frac{1}{2} M_\Phi \eta^2 K + \frac{\eta^2 D^2 L}{\sqrt{b}} \frac{K(K-1)}{2}.$$

Finally, telescoping over all epochs $s = 0, \ldots, S-1$, we get

$$\mathbb{E}\big[\Phi(x_K^S)\big] \leqslant \mathbb{E}\left[\Phi(x_0)\right] - \eta \sum_s \sum_k \mathbb{E}\big[\hat{\mathscr{G}}(x_k^{s+1})\big] + \frac{1}{2} M_\Phi \eta^2 KS + \frac{\eta^2 DLS}{\sqrt{b}} \frac{K(K-1)}{2}.$$

Reordering terms and using the definition of the output in Algorithm 3 $\Big($and the fact that $\mathbb{E}\big[\mathbb{E}\big[\hat{\mathscr{G}}(x_K^S)\big]\big] = \mathbb{E}\left[\mathscr{G}(\hat{x})\right]\Big)$, this gives

$$KS\eta \mathbb{E}\left[\mathscr{G}(\hat{x})\right] \leqslant \Phi(x_0) - \mathbb{E}\big[\Phi(x_K^S)\big] + \frac{1}{2} M_\Phi \eta^2 KS + \frac{\eta^2 DLS}{\sqrt{b}} \frac{K(K-1)}{2},$$

from which the claim follows with $\eta = \frac{1}{\sqrt{KS}}$ and $b = K^2$ as

$$\mathbb{E}\left[\mathscr{G}(\hat{x})\right] \leqslant \frac{1}{\sqrt{KS}} \left(C_{x_0} + \frac{1}{2}(M_\Phi + D^2 L) \right),$$

where $C_{x_0} > 0$ is an initialization-dependent constant, such that $C_{x_0} > \Phi(x_0) - \mathbb{E}\left[\Phi(x^\star)\right] > \Phi(x_0) - \mathbb{E}\left[\Phi(x_K^S)\right]$, where $x^\star$ is a first-order stationary point. □

Choosing a suitable minibatch size is critical to achieving a good performance with variance-reduced approaches, such as SVR-RFW. In Algorithm 3 this translates into a careful choice of K with respect to m: If K is too small, the complexity of the algorithm may be dominated by the cost of recomputing the full gradient frequently. If K is too large, than computing the gradient estimates will be expensive too. We propose to set $K = \lceil m^{1/3} \rceil$, following a convention in the Euclidean FW literature. With that, we get the following complexity guarantees:

COROLLARY 3.3 SVR-RFW with $K = \lceil m^{1/3} \rceil$ obtains an ϵ-accurate solution with IFO complexity of $O\left(m + \frac{m^{2/3}}{\epsilon^2}\right)$ and RLO complexity of $O\left(\frac{1}{\epsilon^2}\right)$.

Proof. It follows directly from Theorem 3.2 that SVR-RFW has an LO complexity of $O\left(\frac{1}{\epsilon^2}\right)$. For the IFO complexity, note that

$$\sum_{s=0}^{S-1} \left(m + \sum_{k=1}^{K-1} b_k \right) = \sum_{s=0}^{S-1} (m + Kb) \lesssim O\left(m + \frac{K^2}{\epsilon^2}\right) = O\left(m + \frac{m^{2/3}}{\epsilon^2}\right),$$

where the last equality follows from setting $K = \lceil m^{1/3} \rceil$. □

Analogously to SRFW, SVR-RFW converges sublinearly to the global optimum, if the objective is g-convex. As before, we use $\Delta_k = \Phi(x_k) - \Phi(x^*)$.

COROLLARY 3.4 If Φ is g-convex, then in the setting of Theorem 3.2 the optimality gap converges as $\mathbb{E}_{I_k}\left[\Delta_k\right] = O(1/\sqrt{KS})$.

The proofs are very similar to that of Corollary 3.2.

A significant shortcoming of the semi-stochastic approach is the need for repeated computation of the full gradient which limits its scalability. In the following section, we introduce an improved version that circumvents these costly computations.

3.3 *Improved gradient estimation with* SPIDER

ALGORITHM 4 SPIDER-RFW

Initialize $x_0 \in \mathscr{X}$, number of iterations K, size of epochs n. Assume access to $\gamma : [0,1] \to \mathscr{M}$.
for $k = 0, 1, \ldots K-1$ **do**
  if $\mathrm{mod}(k, n) = 0$ **then**
    Sample i.i.d. $S_1 = \{\xi_1, ..., \xi_{|S_1|}\}$ (for SRFW) or $S_1 = \left(i_1, \ldots, i_{|S_1|}\right)$ (for SVR-RFW) with predefined $|S_1|$.
    Compute gradient $g_k \leftarrow \mathrm{grad}\, \Phi_{S_1}(x_k)$.
  else
    $|S_2| \leftarrow \left\lceil \min\left\{m, \frac{2nL^2 \|\mathrm{Exp}_{x_{k-1}}^{-1}(x_k)\|}{\epsilon^2}\right\}\right\rceil$
    Sample i.i.d. $S_2 = \{\xi_1, ..., \xi_{|S_2|}\}$ (for SRFW) or $S_2 = \left(i_1, \ldots, i_{|S_2|}\right)$ (for SVR-RFW).
    Compute gradient $g_k \leftarrow \mathrm{grad}\, \Phi_{S_2}(x_k) - \Gamma_{x_{k-1}}^{x_k}\left(\mathrm{grad}\, \Phi_{S_2}(x_{k-1}) - g_{k-1}\right)$.
  end if
  $z_{k+1} \leftarrow \mathrm{argmin}_{z \in \mathscr{X}} \langle g_k, \mathrm{Exp}_{x_k}^{-1}(z)\rangle$.
  $x_{k+1} \leftarrow \gamma(\eta_k)$, where $\gamma(0) = x_k$ and $\gamma(1) = z_{k+1}$.
end for
Output $\hat{x}$ chosen uniformly at random from $\{x^k\}_{k=0}^{K-1}$.

Recently, Nguyen *et al.* (2017) and Fang *et al.* (2018) introduced SPIDER (also known as SARAH) as an efficient way of estimating the (Euclidean) gradient in stochastic optimization tasks. Based on the idea of variance reduction, the algorithm iterates between gradient estimates with different sample size. In particular, it recomputes the gradient at the beginning of each epoch with a larger (constant) batch size; the smaller batch sizes within epochs decrease as we move closer to the optimum. This technique was studied for Riemannian gradient descent in Zhang *et al.* (2018a) and Zhou *et al.* (2018b). In the following, we will introduce an improved variance-reduced STOCHASTIC RFW using SPIDER. Let

$$\mathrm{grad}\, \Phi_S(x) = \begin{cases} \frac{1}{|S|}\sum_{i=1}^{|S|} \mathrm{grad}\, \phi(x, \xi_i), & \text{stochastic} \\ \frac{1}{|S|}\sum_{i=1}^{|S|} \mathrm{grad}\, \phi_i(x), & \text{finite-sum} \end{cases} \tag{3.21}$$

denote the gradient estimate with respect to a sample $S = \{\xi_1, \ldots, \xi_{|S|}\}$ (for stochastic objectives) or $S = \left(i_1, \ldots, i_{|S|}\right)$ (for objectives with finite sum form). Furthermore, we make the following parameter choice (K denoting the number of iterations):

$$\eta = \frac{1}{\sqrt{K}} \quad \text{(step size)} \tag{3.22}$$

$$n = \sqrt{K} = \frac{1}{\epsilon} \quad \text{(\# epochs)} \tag{3.23}$$

$$|S_1| = \begin{cases} \frac{2C^2}{\epsilon^2}, & \text{stochastic} \\ \frac{2L^2D^2}{\epsilon^2}, & \text{finite-sum.} \end{cases} \tag{3.24}$$

Here, ϵ characterizes the goodness of the gradient estimate. $|S_2|$ is recomputed in each iteration as given in Algorithm 4. Note that here m is determined by the number of terms in the finite-sum approximation or we set $m = \infty$ in the stochastic case.

We start by analyzing the goodness of the SPIDER gradient estimate g_k, which is central to our convergence analysis. For $\mathrm{mod}(k, n) = 0$ an upper bound is given by Lemmas 3.1 and 3.5. The critical part is to analyze the case $\mathrm{mod}(k, n) \neq 0$. Let $\mathscr{F}_k$ be the sigma-field generated by the x_k. First, we show that the differences $\left(g_k - \operatorname{grad}\Phi(x_k)\right)_k$ form a martingale with respect to $(\mathscr{F}_k)_k$ (Lemma 3.7). Then, using a classical property of L^2-martingales (Remark 3.2), we can prove the following bound on the approximation error:

LEMMA 3.6 (Goodness of SPIDER-approximation). The expected deviation of the estimate g_k from the true gradient $\operatorname{grad}\Phi$ as defined in Algorithm 4 ($\mathrm{mod}(k, n) \neq 0$) is bounded as $\mathbb{E}\left[\|g_k - \operatorname{grad}\Phi(x_k)\| | \mathscr{F}_k\right] \leqslant \epsilon$.

We first show that the differences form a martingale:

LEMMA 3.7 The differences of the gradient estimates g_k from the true gradients $\operatorname{grad}\Phi$, i.e., $\left(g_k - \operatorname{grad}\Phi(x_k)\right)_k$, form a martingale with respect to the filtration $\left(\mathscr{F}_k\right)_k$.

Proof.

$$\begin{aligned}
\mathbb{E}\left[g_k - \operatorname{grad}\Phi(x_k) | \mathscr{F}_k\right] &= \mathbb{E}\left[\operatorname{grad}\Phi_{S_2}(x_k) - \Gamma_{x_{k-1}}^{x_k}\left(\operatorname{grad}\Phi_{S_2}(x_{k-1}) - g_{k-1}\right) - \operatorname{grad}\Phi(x_k) | \mathscr{F}_k\right] \\
&= \underbrace{\mathbb{E}\left[\operatorname{grad}\Phi_{S_2}(x_k) - \operatorname{grad}\Phi(x_k) | \mathscr{F}_k\right]}_{=0} + \mathbb{E}\left[\Gamma_{x_{k-1}}^{x_k} g_{k-1} - \operatorname{grad}\Phi_{S_2}(x_{k-1}) | \mathscr{F}_k\right] \\
&\overset{*}{=} \Gamma_{x_{k-1}}^{x_k} g_{k-1} - \operatorname{grad}\Phi(x_{k-1}),
\end{aligned}$$

where (*) follows from $\mathbb{E}\left[\operatorname{grad}\Phi_{S_2}(x_k) | \mathscr{F}_k\right] = \operatorname{grad}\Phi(x_k)$, since $\operatorname{grad}\Phi_{S_2}(x_k)$ is assumed to be an unbiased estimate. □

REMARK 3.2 Let $M = (M_k)_k$ denote an L^2-martingale. The orthogonality of increments, i.e.,

$$\langle M_t - M_s, M_v - M_u \rangle = 0 \qquad (v \geqslant u \geqslant t \geqslant s),$$

implies that

$$\mathbb{E}\big[M_k^2\big] = \mathbb{E}\big[M_{k-1}^2\big] + \mathbb{E}\left[\big(M_k - M_{k-1}\big)^2\right] .$$

Therefore, we have recursively

$$\mathbb{E}\big[M_k^2\big] = \mathbb{E}\big[M_0^2\big] + \sum_{i=1}^{k} \mathbb{E}\left[\big(M_i - M_{i-1}\big)^2\right] .$$

We can now prove Lemma 3.6:

Proof. (Lemma 3.6) We consider two cases:

1. $\boxed{\mathrm{mod}(m,k) = 0}$ For stochastic objectives, we have

$$\mathbb{E}\big[\|g_k - \mathrm{grad}\, \Phi(x_k)\|^2 | \mathscr{F}_k\big] = \mathbb{E}\big[\|\mathrm{grad}\, \Phi_{S_1}(x_k) - \mathrm{grad}\, \Phi(x_k)\|^2 | \mathscr{F}_k\big] \overset{(1)}{\leqslant} \frac{C^2}{|S_1|} = \frac{C^2\epsilon^2}{2C^2} = \frac{\epsilon^2}{2} , \tag{3.25}$$

where (1) follows from Lemma 3.1. For objectives with finite-sum form, we have

$$\mathbb{E}\big[\|\mathrm{grad}\, \Phi_{S_1}(x_k) - \mathrm{grad}\, \Phi(x_k)\|^2 | \mathscr{F}_k\big] \overset{(2)}{\leq} \frac{L^2D^2}{|S_1|} = \frac{L^2D^2\epsilon^2}{2L^2D^2} = \frac{\epsilon^2}{2} , \tag{3.26}$$

where (2) follows from Lemma 3.5.

2. $\boxed{\mathrm{mod}(m,k) \neq 0}$ We have

$$\begin{aligned}
&\mathbb{E}\big[\|g_k - \mathrm{grad}\, \Phi(x_k)\|^2 | \mathscr{F}_k\big] \\
&\overset{(3)}{=} \mathbb{E}\left[\left\| \Gamma_{x_{k-1}}^{x_k}\big(g_{k-1} - \mathrm{grad}\, \Phi(x_{k-1})\big)\right\|^2 | \mathscr{F}_k\right] + \mathbb{E}\left[\left\| g_k - \mathrm{grad}\, \Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\big(g_{k-1} - \mathrm{grad}\, \Phi(x_{k-1})\big)\right\|^2 | \mathscr{F}_k\right] \\
&\overset{(4)}{=} \mathbb{E}\left[\left\| \Gamma_{x_{k-1}}^{x_k}\big(g_{k-1} - \mathrm{grad}\, \Phi(x_{k-1})\big)\right\|^2 | \mathscr{F}_k\right] \\
&\quad + \mathbb{E}\left[\left\| \mathrm{grad}\, \Phi_{S_2}(x_k) - \Gamma_{x_{k-1}}^{x_k}\big(\mathrm{grad}\, \Phi_{S_2}(x_{k-1}) - g_{k-1}\big) - \mathrm{grad}\, \Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\big(g_{k-1} - \mathrm{grad}\, \Phi(x_{k-1})\big)\right\|^2 | \mathscr{F}_k\right] \\
&= \mathbb{E}\left[\left\| \Gamma_{x_{k-1}}^{x_k}\big(g_{k-1} - \mathrm{grad}\, \Phi(x_{k-1})\big)\right\|^2 | \mathscr{F}_k\right] \\
&\quad + \mathbb{E}\left[\left\| \mathrm{grad}\, \Phi_{S_2}(x_k) - \mathrm{grad}\, \Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\big(\mathrm{grad}\, \Phi_{S_2}(x_{k-1}) - \mathrm{grad}\, \Phi(x_{k-1})\big)\right\|^2 | \mathscr{F}_k\right] .
\end{aligned}$$

In the chain of inequalities, (3) follows from Remark 3.2 and (4) from substituting g_k according to Algorithm 4.

In the following, we assume that Φ is a *stochastic function*. Analogous arguments hold, if Φ has a finite-sum structure. We introduce the shorthand

$$\zeta_i = \operatorname{grad}\phi(x_k,\xi_i) - \operatorname{grad}\Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\big(\operatorname{grad}\phi(x_{k-1},\xi_i) - \operatorname{grad}\Phi(x_{k-1})\big).$$

Then, we get for the second term

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{grad}\Phi_{S_2}(x_k) - \operatorname{grad}\Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\left(\operatorname{grad}\Phi_{S_2}(x_{k-1}) - \operatorname{grad}\Phi(x_{k-1})\right)\right\|^2 |\mathscr{F}_k\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{|S_2|}\sum_{i=1}^{|S_2|}\zeta_i\right\|^2 |\mathscr{F}_k\right] = \frac{1}{|S_2|^2}\mathbb{E}\left[\left\|\sum_{i=1}^{|S_2|}\zeta_i\right\|^2 |\mathscr{F}_k\right] \\
&\overset{(5)}{\leqslant} \frac{1}{|S_2|^2}\mathbb{E}\left[\left(\sum_{i=1}^{|S_2|}\|\zeta_i\|\right)^2 |\mathscr{F}_k\right] \overset{(6)}{=} \frac{1}{|S_2|^2}\mathbb{E}\left[\sum_{i=1}^{|S_2|}\|\zeta_i\|^2 |\mathscr{F}_k\right] \\
&= \frac{1}{|S_2|^2}\sum_{i=1}^{|S_2|}\mathbb{E}\left[\|\zeta_i\|^2|\mathscr{F}_k\right] \overset{(7)}{=} \frac{1}{|S_2|}\mathbb{E}\left[\|\zeta_i\|^2|\mathscr{F}_k\right] \\
&= \frac{1}{|S_2|}\mathbb{E}\left[\left\|\operatorname{grad}\phi(x_k,\xi) - \operatorname{grad}\Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\left(\operatorname{grad}\phi(x_{k-1},\xi\right) - \operatorname{grad}\Phi(x_{k-1})\right\|^2 |\mathscr{F}_k\right],
\end{aligned}$$

where (5) follows from the triangle-inequality, (6) from $\mathbb{E}\left[\zeta_i\right] = 0$, see Equation 3.1; and (7) from the ζ_i being i.i.d. Note that

$$\begin{aligned}
\mathbb{E}\left[\operatorname{grad}\phi(x_k,\xi)|\mathscr{F}_k\right] &= \operatorname{grad}\Phi(x_k) \\
\mathbb{E}\left[\Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_{k-1},\xi)|\mathscr{F}_k\right] &= \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\Phi(x_{k-1}).
\end{aligned}$$

With this, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\operatorname{grad}\phi(x_k,\xi) - \operatorname{grad}\Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\left(\operatorname{grad}\phi(x_{k-1},\xi\right) - \operatorname{grad}\Phi(x_{k-1})\right\|^2 |\mathscr{F}_k\right] \\
&= \mathbb{E}\left[\left\|\operatorname{grad}\phi(x_k,\xi) - \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_{k-1},\xi) - \underbrace{\left(\operatorname{grad}\Phi(x_k) - \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\Phi(x_{k-1})\right)}_{=\mathbb{E}\left[\operatorname{grad}\phi(x_k,\xi) - \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_k,\xi)|\mathscr{F}_k\right]}\right\|^2 |\mathscr{F}_k\right] \\
&= \mathbb{E}\left[\|\operatorname{grad}\phi(x_k,\xi) - \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_{k-1},\xi)\|^2\right] - \underbrace{\left\|\mathbb{E}\left[\operatorname{grad}\phi(x_k,\xi) - \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_k,\xi)|\mathscr{F}_k\right]\right\|^2}_{\geq 0} \\
&\leqslant \mathbb{E}\left[\|\operatorname{grad}\phi(x_k,\xi) - \Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_{k-1},\xi)\|^2|\mathscr{F}_k\right].
\end{aligned}$$

In summary, we have for the second term

$$\mathbb{E}\left[\left\|\operatorname{grad}\Phi_{S_2}(x_k)-\operatorname{grad}\Phi(x_k)-\Gamma_{x_{k-1}}^{x_k}\left(\operatorname{grad}\Phi_{S_2}(x_{k-1})-\operatorname{grad}\Phi(x_{k-1})\right)\right\|^2|\mathscr{F}_k\right] \tag{3.27}$$

$$\leqslant \mathbb{E}\left[\|\operatorname{grad}\phi(x_k,\xi)-\Gamma_{x_{k-1}}^{x_k}\operatorname{grad}\phi(x_{k-1},\xi)\|^2|\mathscr{F}_k\right]. \tag{3.28}$$

Putting everything together, we get

$$\begin{aligned}
&\mathbb{E}\left[\|g_k-\operatorname{grad}\Phi(x_k)\|^2|\mathscr{F}_k\right]\\
&\leqslant \mathbb{E}\left[\left\|\Gamma_{x_{k-1}}^{x_k}\left(g_{k-1}-\operatorname{grad}\Phi(x_{k-1})\right]\right)\right\|^2|\mathscr{F}_k\Big]+\frac{1}{|S_2|}\mathbb{E}\left[\left\|\operatorname{grad}\phi(x_k,\xi)-\Gamma_{x_{k-1}}^{x_k}\left(\operatorname{grad}\phi(x_{k-1},\xi)\right)\right\|^2|\mathscr{F}_k\right]\\
&\overset{(8)}{\leqslant} \mathbb{E}\left[\left\|\Gamma_{x_{k-1}}^{x_k}\left(g_{k-1}-\operatorname{grad}\Phi(x_{k-1})\right)\right\|^2|\mathscr{F}_k\right]+\frac{1}{|S_2|}L^2\|\operatorname{Exp}_{x_{k-1}}^{-1}(x_k)\||\mathscr{F}_k\\
&\overset{(9)}{=} \mathbb{E}\left[\left\|\Gamma_{x_{k-1}}^{x_k}\left(g_{k-1}-\operatorname{grad}\Phi(x_{k-1})\right)\right\|^2|\mathscr{F}_k\right]+\frac{\epsilon^2}{2mL^2\|\operatorname{Exp}_{x_{k-1}}^{-1}(x_k)\|}L^2\|\operatorname{Exp}_{x_{k-1}}^{-1}(x_k)\|\\
&= \mathbb{E}\left[\left\|\Gamma_{x_{k-1}}^{x_k}\left(g_{k-1}-\operatorname{grad}\Phi(x_{k-1})\right)\right\|^2|\mathscr{F}_k\right]+\frac{\epsilon^2}{2m},
\end{aligned}$$

where (8) follows from ϕ being L-Lipschitz and (9) follows from the choice of $|S_2|$ in Algorithm 4. Recursively going back to the beginning of the epoch (see Remark 3.2), we get (with $k_0=\lfloor\frac{k}{m}\rfloor m$):

$$\mathbb{E}\big[\|g_k-\operatorname{grad}\Phi(x_k)\|^2|\mathscr{F}_k\big]\leqslant \underbrace{\mathbb{E}\big[\|g_{k_0}-\operatorname{grad}\Phi(x_{k_0})\|^2|\mathscr{F}_{k_0}\big]}_{\leq\frac{\epsilon^2}{2}\ \text{Eq.}(3.25)}+m\frac{\epsilon^2}{2m}\leqslant\epsilon^2.$$

With Jensen's inequality, we have

$$\left(\mathbb{E}\left[\|g_k-\operatorname{grad}\Phi(x_k)\||\mathscr{F}_k\right]\right)^2\leqslant\mathbb{E}\big[\|g_k-\operatorname{grad}\Phi(x_k)\|^2|\mathscr{F}_k\big]\leqslant\epsilon^2,$$

which gives

$$\mathbb{E}\left[\|g_k-\operatorname{grad}\Phi(x_k)\||\mathscr{F}_k\right]\leqslant\epsilon.$$

Analogously, if Φ has a finite-sum structure with component functions ϕ_i that are L-Lipschitz, we get

$$\mathbb{E}\big[\|g_k-\operatorname{grad}\Phi(x_k)\|^2|\mathscr{F}_k\big]\leqslant \underbrace{\mathbb{E}\big[\|g_{k_0}-\operatorname{grad}\Phi(x_{k_0})\|^2|\mathscr{F}_{k_0}\big]}_{\leqslant\frac{\epsilon^2}{2}\ \text{Eq.}(3.26)}+m\frac{\epsilon^2}{2m}\leqslant\epsilon^2,$$

from which, again, with Jensen's inequality the claim follows as

$$\mathbb{E}\left[\|g_k - \operatorname{grad} \Phi(x_k)\| | \mathscr{F}_k\right] \leqslant \epsilon.$$

□

With this preparatory work, we arrive at the main result for this section: We show that SPIDER-RFW attains a global sublinear convergence rate for nonconvex objectives.

THEOREM 3.3 (Convergence SPIDER-RFW). With the parameter choices (3.22), Algorithm 4 converges in expectation with rate $\mathbb{E}\left[\mathscr{G}(\hat{x})\right] = O\left(\frac{1}{\sqrt{K}}\right)$.

Proof. We again have

$$\begin{aligned}\Phi(x_{k+1}) &\overset{(1)}{\leqslant} \Phi(x_k) + \eta_k \langle \operatorname{grad}\ \Phi(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \frac{1}{2} M_\Phi \eta_k^2 \\ &\overset{(2)}{\leqslant} \Phi(x_k) + \eta_k \langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \eta_k \langle \operatorname{grad}\ \Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \frac{1}{2} M_\Phi \eta_k^2,\end{aligned}$$

where, (1) follows from Lemma 3.4 and (2) follows from 'adding a zero' with respect to g_k. We again apply the Cauchy–Schwartz inequality to the inner product and make use of the fact that the geodesic distance between points in $\mathscr{X}$ is bounded by its diameter:

$$\langle \operatorname{grad}\ \Phi(x_k) - g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle \leqslant \|\operatorname{grad}\ \Phi(x_k) - g_k(x_k)\| \cdot \underbrace{\|\operatorname{Exp}_{x_k}^{-1}(y_k)\|}_{\leqslant \operatorname{diam}(\mathscr{X})}. \tag{3.29}$$

This gives (with $D := \operatorname{diam}(\mathscr{X})$)

$$\Phi(x_{k+1}) \leqslant \Phi(x_k) + \eta_k \langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle + \eta_k D \|\operatorname{grad}\ \Phi(x_k) - g_k(x_k)\| + \frac{1}{2} M_\Phi \eta_k^2.$$

Taking expectations, we get

$$\mathbb{E}\left[\Phi(x_{k+1})\right] \leqslant \mathbb{E}\left[\Phi(x_k)\right] + \eta_k \underbrace{\mathbb{E}\left[\langle g_k(x_k), \operatorname{Exp}_{x_k}^{-1}(y_k)\rangle\right]}_{= -\mathbb{E}\left[\hat{\mathscr{G}}(x_k)\right]} + \eta_k D \underbrace{\mathbb{E}\left[\|\operatorname{grad}\ \Phi(x_k) - g_k(x_k)\|\right]}_{\leq \epsilon} + \frac{1}{2} M_\Phi \eta_k^2.$$

With Lemma 3.6 and the definition of the stochastic Fw gap, this can be rewritten as

$$\mathbb{E}\left[\Phi(x_{k+1})\right] \leqslant \mathbb{E}\left[\Phi(x_k)\right] - \eta_k \mathbb{E}\left[\hat{\mathscr{G}}(x_k)\right] + \eta_k D \epsilon + \frac{1}{2} M_\Phi \eta_k^2.$$

Summing and telescoping gives

$$\begin{aligned}\mathbb{E}\left[\mathcal{G}(\hat{x})\right]\sum_k \eta_k &\leqslant \mathbb{E}\left[\Phi(x_0)\right] - \mathbb{E}\left[\Phi(x_K)\right] + D\epsilon\sum_k \eta_k + \frac{1}{2}M_\Phi\sum_k \eta_k^2 \\ &\leqslant \left(\Phi(x_0) - \mathbb{E}\left[\Phi(x_K)\right]\right) + D\epsilon\sum_k \eta_k + \frac{1}{2}M_\Phi\sum_k \eta_k^2,\end{aligned}$$

where we have again used the definition of the output in Algorithm 4; in particular, that $\mathbb{E}\big[\mathbb{E}\big[\hat{\mathcal{G}}(x_K)\big]\big] = \mathbb{E}\left[\mathcal{G}(\hat{x})\right]$. With $\eta_k = \eta = \frac{1}{\sqrt{K}}$, this becomes

$$\underbrace{K\eta}_{=\sqrt{K}}\,\mathbb{E}\left[\mathcal{G}(\hat{x})\right] \leq \left(\Phi(x_0) - \mathbb{E}\left[\Phi(x_K)\right]\right) + D\epsilon\,\underbrace{K\eta}_{=\sqrt{K}} + \frac{1}{2}M_\Phi\,\underbrace{K\eta^2}_{=1}\,.$$

Note, that $\epsilon = \frac{1}{n} = \frac{1}{\sqrt{K}}$. Dividing by $\sqrt{K}$ then gives the claim

$$\mathbb{E}\left[\mathcal{G}(\hat{x})\right] \leqslant \frac{1}{\sqrt{K}}\left(C_{x_0} + D\,\underbrace{\epsilon\sqrt{K}}_{=1} + \frac{1}{2}M_\Phi\right)\,, \tag{3.30}$$

where $C_{x_0} > \Phi(x_0) - \Phi(x^\star)$ depends on the initialization only and $x^\star$ is a first-order stationary point. □

COROLLARY 3.5 SPIDER-RFW obtains an ϵ-accurate solution with SFO/ IFO complexity of $O\left(\frac{1}{\epsilon^3}\right)$ and RLO complexity of $O\left(\frac{1}{\epsilon^2}\right)$.

Proof. It follows directly from Theorem 3.3 that SPIDER-RFW has an RLO complexity of $O\left(\frac{1}{\epsilon^2}\right)$. For the SFO complexity, consider a stochastic objective Φ. Then

$$SFO = \sum_{s=1}^{n}\left(|S_1| + \mathbb{E}\left[\sum_{k=2}^{n}|S_2|\right]\right)\,.$$

We have

$$\mathbb{E}\left[\sum_{k=2}^{n}|S_2|\right] = \mathbb{E}\left[\sum_{k=2}^{n}\frac{2nL\|\mathrm{Exp}_{x_{k-1}}^{-1}(x_k)\|}{\epsilon^2}\right] \overset{(1)}{\leqslant} \frac{2n^2L^2(D^2\eta^2)}{2\epsilon^2} \overset{(2)}{=} O\left(\frac{1}{\epsilon^2}\right)\,,$$

where (1) follows from $\|\mathrm{Exp}_{x_{k-1}}^{-1}(x_k)\| \leq \eta D$ (see Eq. 6) and (2) from $\eta = \frac{1}{n}$ by construction. This gives

$$SFO = O\left(n\left(\frac{1}{\epsilon^2} + \frac{1}{\epsilon^2}\right)\right) = O\left(\frac{1}{\epsilon^3}\right).$$

An analogous argument gives the IFO complexity, if Φ has a finite-sum structure. □

We again consider the special case of g-convex objectives for completeness. Here, we obtain a result on function suboptimality:

COROLLARY 3.6 If Φ is g-convex, one can show under the assumptions of Theorem 3.3 a similar convergence rate for the optimality gap, i.e., $\mathbb{E}\left[\Delta_k\right] = O(1/\sqrt{K})$.

The proof is analogous to the proof of Corollary 3.2 (for stochastic objectives) and of Corollary 3.4 (for objectives with finite-sum structure).

4. Experiments

(Stochastic) Riemannian optimization is frequently considered in the machine learning literature, including for the computation of hyperbolic embeddings (Sala *et al.*, 2018), low-rank matrix and tensor factorization (Vandereycken, 2013) and eigenvector based methods (Journée *et al.*, 2010; Zhang *et al.*, 2016; Tripuraneni *et al.*, 2018).

In this section we validate the proposed stochastic algorithms by comparison with the deterministic RFW (Weber & Sra, 2017) and state-of-the-art stochastic Riemannian optimization methods. All experiments were performed in MATLAB.

Our numerical experiments use synthetic data, consisting of sets of symmetric, positive definite matrices. We generate matrices by sampling real matrices of dimension d uniformly at random $M_i \sim \mathscr{U}(\mathbb{R}^{d\times d})$ and then multiplying each with its transpose $M_i \leftarrow M_i M_i^T$. To generate ill-conditioned matrices, we sample matrices with a rank deficit $U_i \sim \mathscr{U}(\mathbb{R}^{d\times d})$ (with $\operatorname{rank}(U) < d$) and set $B_i \leftarrow \delta I + U_i U_i^T$ (for a small $\delta > 0$).

Throughout the experiments, the hyperparameter choices (b, K) are guided by the specifications in Algorithm 2 (for SRFW) and Algorithm 3 (for SVR-RFW) and their theoretical analysis. All RFW methods are implemented with decreasing step sizes.

4.1 *Riemannian centroid*

The computation of the *Riemannian centroid* (also known as the *geometric matrix mean* or the *Karcher mean*) is a canonical benchmark task for testing Riemannian optimization methods (Zhang *et al.*, 2016; Kasai *et al.*, 2018a,b). Besides its importance as a benchmark, the Karcher mean is a fundamental subroutine in many machine learning methods, for instance, in the computation of hyperbolic embeddings (Sala *et al.*, 2018). Although the Karcher mean problem is nonconvex in Euclidean space, it is g-convex in the Riemannian setting. This allows for the application of RFW, in addition to the stochastic methods discussed above. RFW requires the computation of the full gradient in each iteration step, whereas the stochastic variants implement gradient estimates at a significantly reduced computational cost. This results in observable performance gains as shown in our experiments (Fig. 1).

Formally, the Riemannian centroid is defined as the mean of a set $M = \{M_i\}$ of $d \times d$ positive definite matrices (we write $|M| = m$) with respect to the Riemannian metric. This task requires solving

$$\min_{H \preceq X \preceq A} \sum_{i=1}^{m} w_i \delta_R^2(X, M_i) = \sum_{i=1}^{m} w_i \left\| \log\left(X^{-1/2} M_i X^{-1/2}\right) \right\|_F^2,$$

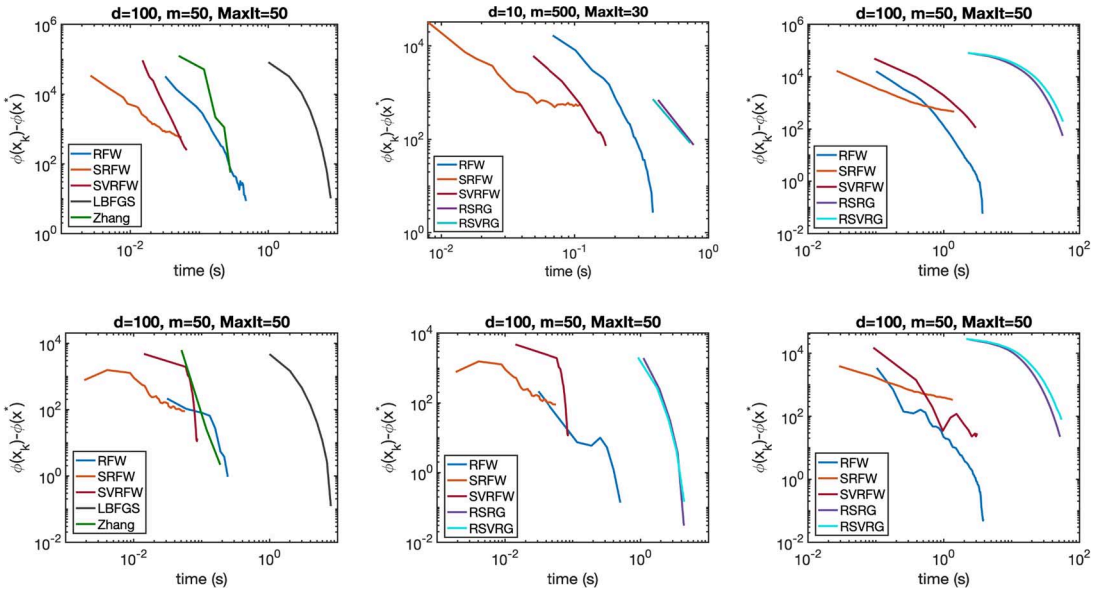

FIG. 1. **Riemannian centroid.** RFW and its stochastic variants in comparison with state-of-the-art Riemannian optimization methods (*parameters:d*, size of matrices; *m*, number of matrices; *MaxIt*, number of iterations). All experiments are initialized with the harmonic mean. Hereby, we compare against deterministic algorithms (LBFGS and ZHANG, left) as well as recent state-of-the-art stochastic Riemannian algorithms R-SRG and RSVRG (middle and right). The results in the top row are for well-conditioned matrices, the results in the bottom row are for ill-conditioned matrices.

where $\|\cdot\|_F$ denotes the Frobenius norm and $w_i \in [0,1]$ are weights, where $\sum_{i=1}^m w_i = 1$. The well-known *matrix means inequality* bounds the Riemannian mean from above and below with respect to the Löwner order: The *harmonic mean* $H := \left(\sum_i w_i M_i^{-1}\right)^{-1}$ gives a lower bound on the geometric matrix mean, while the arithmetic mean $A := \sum_i w_i M_i$ provides an upper bound (Bhatia, 2007). This allows for phrasing the computation of the Riemannian centroid as a constrained optimization task with interval constraints given by the harmonic and arithmetic means (though it could be solved as unconstrained task too). Writing $\phi_i(X) = w_i \delta_R^2(X, M_i)$, we note that the gradient of the objective is given by $\nabla \phi_i(X) = w_i X^{-1} \log(X M_i^{-1})$ (see, e.g., Bhatia, 2007, Ch.6), whereby the corresponding Riemannian 'linear' oracle reduces to solving

$$Z_k \leftarrow \operatorname*{argmin}_{H \preceq Z \preceq A} \left\langle X_k^{1/2} \nabla \phi_i(X_k) X_k^{1/2},\, \log\left(X_k^{-1/2} Z X_k^{-1/2}\right) \right\rangle . \tag{4.1}$$

Remarkably, (4.1) can be solved in closed form (Weber & Sra, 2017, Theorem 4.1), which we exploit to achieve an efficient implementation of RFW and STOCHASTIC RFW. For completeness, we recall the theorem below:

THEOREM 4.1 (Theorem 4.1 (Weber & Sra, 2017)). Let $L, U \in \mathbb{P}_d$ such that $L \prec U$. Let $S \in \mathbb{H}_d$ and $X \in \mathbb{P}_d$ be arbitrary. Then, the solution to the optimization problem

$$\min_{L \preceq Z \preceq U} \quad \operatorname{tr}(S \log(XZX)) \tag{4.2}$$

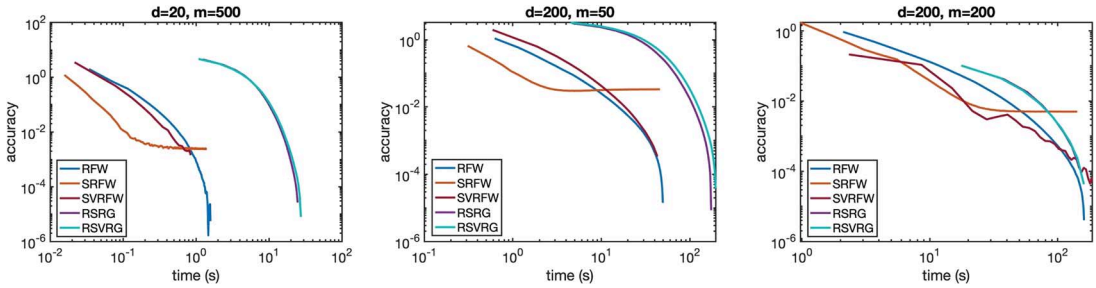

FIG. 2. **Riemannian centroids.** Accuracy of RFW and stochastic variants in comparison with RSRG and RSVRG for inputs of different size (d, size of matrices; m, number of matrices). All experiments are initialized with the arithmetic mean.

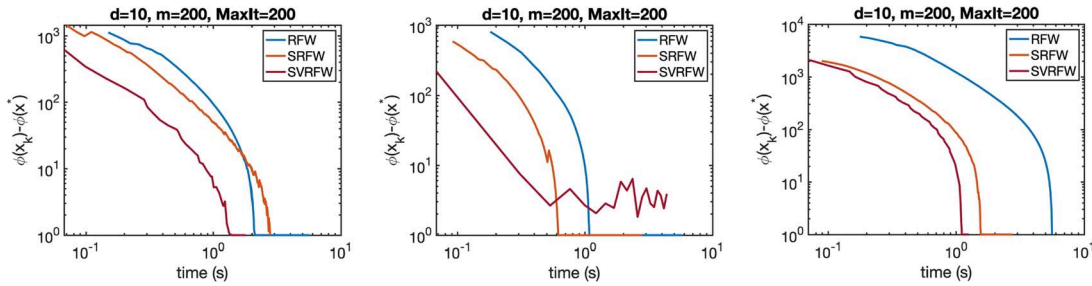

FIG. 3. **Wasserstein barycenters.** Performance of RFW and stochastic variants for well-conditioned inputs of fixed size (d, size of matrices; m, number of matrices; K, number of iterations) with different initializations: $X_0 \sim \mathscr{C}$ (left), $X_0 = \frac{1}{2}(\alpha I + A)$ (middle) and $X_0 = A$ (right). Here, A denotes the arithmetic mean of $\mathscr{C}$ and α the smallest eigenvalue over $\mathscr{C}$.

is given by $Z = X^{-1}Q\big(P^*[-\mathrm{sgn}(D)]_+P + \hat{L}\big)Q^*X^{-1}$, where $S = QDQ^*$ is a diagonalization of S, $\hat{U} - \hat{L} = P^*P$ with $\hat{L} = Q^*XLXQ$ and $\hat{U} = Q^*XUXQ$.

Setting $L = H$, $U = A$ and $S = X_k^{1/2}\nabla\phi_i(X_k)X_k^{1/2}$, this result gives a closed form solution to Eq. 4.1.

To evaluate the efficiency of our methods, we compare against state-of-the-art algorithms. First, Riemannian LBFGS, a quasi-Newton method (Yuan *et al.*, 2016), for which we use an improved limited-memory version of the method available in *Manopt* (Boumal *et al.*, 2014). Secondly ZHANG's method (Zhang, 2017), a recently published majorization–minimization method for computing the geometric matrix mean. Against both (deterministic) algorithms we observe significant performance gains (Fig. 1). In Weber & Sra (2017), RFW is compared with a wide range of Riemannian optimization methods and varying choices of hyperparameters. In those experiments, LBFGS and ZHANG's method were reported to be especially competitive, which motivates our choice. We further present two instances of comparing STOCHASTIC RFW against stochastic gradient-based methods (RSRG and RSVRG; Kasai *et al.*, 2018a), both of which are outperformed by our RFW approach. In all experiments, we assume uniform weights, i.e., $w_i = \frac{1}{m}\ \forall\ i \in [m]$.

In a second experiment, we compare the accuracy $\left(\text{i.e., } \frac{|\phi(x_{\text{final}}) - \phi(x^*)|}{|\phi(x^*)|}\right)$ of RFW and its stochastic variants with that of RSRG and RSVRG. Figure 2 shows that STOCHASTIC RFW reach a medium accuracy fast; however, ultimately RFW, as well as R-SRG and RSVRG reach a higher accuracy. STOCHASTIC RFW is therefore particularly suitable for data science and machine learning applications, where we encounter high-dimensional, large-scale data sets and very high accuracy is not required.

We note that the comparison experiments are not quite fair to our methods, as neither R-SRG nor RSVRG implement the noted projection operation (see discussion in Section 2.3) required to align their implementation with their theory.

4.2 *Wasserstein barycenters*

The computation of means of empirical probability measures with respect to the optimal transport metric (or *Wasserstein distance*) is a basic task in statistics. Here, we consider the problem of computing such *Wasserstein barycenters* of multivariate (centered) Gaussians. This corresponds to the following minimization task on the *Gaussian density manifold* (also known as *Bures manifold*):[1]

$$\min_{\alpha I \preceq X \preceq A} \sum_{i=1}^{M} d_W^2(X, \mathscr{C}) = \sum_i w_i \big[\operatorname{tr}(C_i + X) - 2\operatorname{tr}\big(C_i^{1/2} X C_i^{1/2}\big)^{1/2}\big], \tag{4.3}$$

where $\mathscr{C} = \{C_i\} \subseteq \mathbb{P}(n)$, $|\mathscr{C}| = m$ are the covariance matrices of the Gaussians, $w_i \in [0,1]$ weights ($\sum_i w_i = 1$) and α denotes their minimal eigenvalue over $\mathscr{C}$. Note that the Gaussian density manifold is isomorphic to the manifold of symmetric positive definite matrices considered in the previous section. This allows for a direct application of RFW to Eq. 4.3, albeit with a different set of constraints.

A closely related problem is the task of computing Wasserstein barycenters of *matrix-variate Gaussians*, i.e., multivariate Gaussians whose covariance matrices are expressed as suitable Kronecker products. Such models are of interest in several inference problems, see for instance Stegle *et al.* (2011). By plugging in Kronecker structured covariances into (4.3), the corresponding barycenter problem takes the form

$$\min_{X \succ 0} \sum_{i=1}^{n} \operatorname{tr}(A_i \otimes A_i) + \operatorname{tr}(X \otimes X) - 2\operatorname{tr}\big[(A_i \otimes A_i)^{1/2}(X \otimes X)(A_i \otimes A_i)^{1/2}\big]^{1/2}. \tag{4.4}$$

Remarkably, despite the product terms, problem (4.4) turns out to be (Euclidean) convex (Lemma 4.1). This allows one to apply (g-) convex optimization tools, and use convexity to conclude global optimality. This result should be of independent interest.

LEMMA 4.1 The barycenter problem for matrix-variate Gaussians (Eq. 4.4) is convex.

For the proof, recall the following well-known properties of Kronecker products:

LEMMA 4.2 (Properties of Kronecker products). Let $A, B, C, D \in \mathbb{P}^d$.

1. $(A \otimes A)^{1/2} = A^{1/2} \otimes A^{1/2}$;
2. $AC \otimes BD = (A \otimes B)(C \otimes D)$.

Furthermore, recall the Ando–Lieb theorem (Ando, 1979):

THEOREM 4.2 (Ando–Lieb). Let $A, B \in \mathbb{P}^d$. Then the map $(A, B) \mapsto A^\gamma \otimes B^{1-\gamma}$ is jointly concave for $0 < \gamma < 1$.

[1] Interestingly, this problem turns out to be Euclidean convex (more precisely, a nonlinear semidefinite program). However, a Riemannian approach exploits the problem structure more explicitly.

Equipped with those two arguments, we can prove the lemma.

Proof. (Lemma 4.1) First, note that

$$\operatorname{tr}(A_i \otimes A_i) = (\operatorname{tr} A_i)(\operatorname{tr} A_i) = (\operatorname{tr} A_i)^2 \qquad \forall\, i = 1, \dots n$$
$$\operatorname{tr}(X \otimes X) = (\operatorname{tr} X)(\operatorname{tr} X) = (\operatorname{tr} X)^2.$$

Next, consider the third term. We have

$$\begin{aligned}
\operatorname{tr}\left[\left((A_i \otimes A_i)^{1/2}(X \otimes X)(A_i \otimes A_i)^{1/2}\right)^{1/2}\right] &\overset{(1)}{=} \operatorname{tr}\left[\left((A_i^{1/2} X \otimes A_i^{1/2} X)(A_i \otimes A_i)^{1/2}\right)^{1/2}\right] \\
&\overset{(1)}{=} \operatorname{tr}\left[\left((A_i^{1/2} X A_i^{1/2}) \otimes (A_i^{1/2} X A_i^{1/2})\right)^{1/2}\right] \\
&\overset{(2)}{=} \operatorname{tr}\left[\left(A_i^{1/2} X A_i^{1/2}\right)^{1/2} \otimes \left(A_i^{1/2} X A_i^{1/2}\right)^{1/2}\right],
\end{aligned}$$

where (1) follows from Lemma 4.2(ii) and (2) follows from Lemma 4.2(i). Note that $X \mapsto A^{1/2} X A^{1/2}$ is a linear map. Therefore, we can now apply the Ando–Lieb theorem with $\gamma = \frac{1}{2}$, which establishes the concavity of the trace term. Its negative is convex and, consequently, the objective is a sum of convex functions. The claim follows from the convexity of sums of convex functions. □

One can show that the Wasserstein mean is upper bounded by the arithmetic mean A and lower bounded by αI, where α denotes the smallest eigenvalue over $\mathscr{C}$ (Bhatia *et al.*, 2018a,b). This allows for computing the Wasserstein mean via constrained optimization (though, again, one could use unconstrained tools too). For computing the gradient, note that the Riemannian gradient $\operatorname{grad} \phi(X)$ can be written as $\operatorname{grad} \phi(X) = X \nabla \phi(X) - \nabla \phi(X) X$, where $\nabla \phi$ is the Euclidean gradient (where ϕ denotes the objective in (4.3)). It is easy to show, that

$$\nabla \phi(X) = \sum_i w_i \left(I - \left(C_i X\right)^{-1/2} C_i\right),$$

which directly gives the gradient of the objective.

We evaluate the performance of our stochastic RFW methods against the deterministic RFW method for different initializations (Fig. 3). We again assume that $w_i = \frac{1}{m}\ \forall\ i \in [m]$. Our results indicate that all three initializations are suitable. This suggests, that (stochastic) RFW is not sensitive to initialization and performs well even if not initialized close to the optimum. In a second experiment, we compute Wasserstein barycenters of MVNs for different input sizes (Fig. 4). Both experiments indicates that especially the purely stochastic SRFW improves on RFW with comparable accuracy and stability. We did not compare against projection-based methods in the case of Wasserstein barycenters, since to our knowledge there are no implementations with the appropriate projections available.

5. Discussion

We introduced three stochastic Riemannian Fw methods, which go well beyond the deterministic RFW algorithm proposed in Weber & Sra (2017). In particular, we (i) allow for an application to nonconvex,

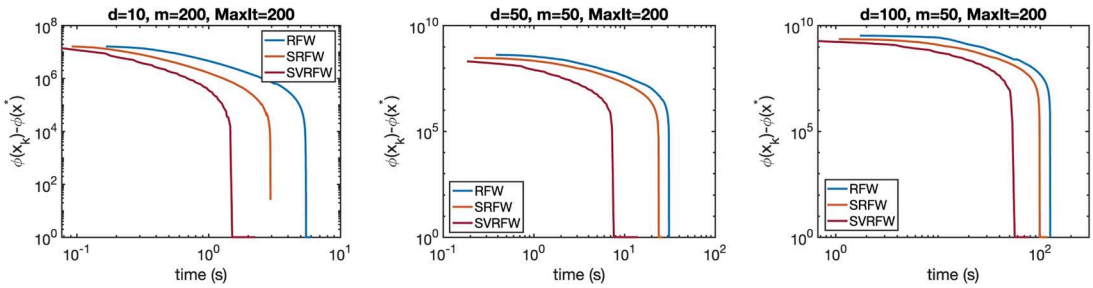

FIG. 4. **Wasserstein barycenters for MVNs.** Performance of RFW and stochastic variants for well-conditioned inputs of different sizes (d, size of matrices; m, number of matrices; K, number of iterations); initialized at $X_0 = A$. Again, A denotes the arithmetic mean of $\mathscr{C}$ and α the smallest eigenvalue over $\mathscr{C}$.

stochastic problems and (ii) improve the oracle complexities by replacing the computation of full gradients with stochastic gradient estimates. For the latter task, we analyze both fully stochastic and semi-stochastic variance-reduced estimators. Moreover, we implement the recently proposed SPIDER technique that significantly improves the classical Robbins–Monroe and variance-reduced gradient estimates by circumventing the need to recompute full gradients periodically.

We discuss applications of our methods to the computation of the Riemannian centroid and Wasserstein barycenters, both fundamental subroutines of potential value in several applications, including in machine learning. In validation experiments, we observe performance gains compared to the deterministic RFW as well as state-of-the-art deterministic and stochastic Riemannian methods.

This paper focused on developing a non-asymptotic convergence analysis and on establishing theoretical guarantees for our methods. Future work includes implementation of our algorithms for other manifolds and other classical Riemannian optimization tasks (see, e.g., Absil & Hosseini, 2017). This includes tasks with constraints on determinants or condition numbers. An important example for the latter is the task of learning a DPP kernel (see, e.g., Mariet & Sra, 2015), which can be formulated as a stochastic, geodesically convex problem. We hope to explore practical applications of our approach to large-scale constrained problems in machine learning and statistics.

Furthermore, instead of using exponential maps, one can reformulate our proposed methods using retractions. For projected-gradient methods, the practicality of retraction-based approaches has been established (Absil *et al.*, 2008), rendering this a promising extension for future research.

Acknowledgements

The authors thank Charles Fefferman and an anonymous reviewer for helpful comments on the manuscript.

Funding

SS acknowledges support from an NSF BIGDATA grant (1741341) and the NSF CAREER grant (1846088).

REFERENCES

ABSIL, P.-A. & HOSSEINI, S. (2017) *A Collection of Nonsmooth Riemannian Optimization Problems*. International Series of Numerical Mathematics.

Absil, P.-A., Mahony, R. & Sepulchre, R. (2008) *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press.

Agarwal, A. & Bottou, L. (2015) A lower bound for the optimization of finite sums. ICML'15, pp. 78–86. JMLR.org.

Ando, T. (1979) Concavity of certain maps on positive definite matrices and applications to Hadamard products. *Linear Algebra Appl.*, **26**, 203–241.

Bhatia, R. (2007) *Positive Definite Matrices*. Princeton University Press.

Bhatia, R., Jain, T. & Lim, Y. (2018a) On the Bures–Wasserstein distance between positive definite matrices. *Exposition. Math.*

Bhatia, R., Jain, T. & Lim, Y. (2018b) Strong convexity of sandwiched entropies and related optimization problems. *Rev. Math. Phys.*, **30**, 1850014.

Billera, L. J., Holmes, S. P. & Vogtmann, K. (2001) Geometry of the space of phylogenetic trees. *Adv. Appl. Math.*, **27**, 733–767.

Bonnabel, S. (2013) Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Automat. Contr.*, **58**, 2217–2229.

Boumal, N., Mishra, B., Absil, P.-A. & Sepulchre, R. (2014) Manopt, a Matlab toolbox for optimization on manifolds. *J. Mach. Learn. Res.*, **15**, 1455–1459.

Edelman, A., Arias, T. A. & Smith, S. T. (1998) The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, **20**, 303–353.

Fang, C., Li, C. J., Lin, Z. & Zhang, T. (2018) SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *NeurIPS*.

Frank, M. & Wolfe, P. (1956) An algorithm for quadratic programming. *Nav. Res. Logist. Q.*, **3**.

Huang, W., Absil, P.-A. & Gallivan, K. A. (2018) A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems. *SIAM J. Optim.*, **28**, 470–495.

Jaggi, M. (2013) Revisiting Frank–Wolfe: projection-free sparse convex optimization. *International Conference on Machine Learning (ICML)*, pp. 427–435.

Jost, J. (2011) *Riemannian Geometry and Geometric Analysis*. Springer.

Journée, M., Bach, F., Absil, P.-A. & Sepulchre, R. (2010) Low-rank optimization on the cone of positive semidefinite matrices. *SIAM J Optim.*, **20**, 2327–2351.

Kasai, H., Jawanpuria, P. & Mishra, B. (2019) *Adaptive stochastic gradient algorithms on Riemannian manifolds*.

Kasai, H., Mishra, B. & Sato, H. (2018a) *Rsopt (Riemannian stochastic optimization algorithms)*.

Kasai, H., Sato, H. & Mishra, B. (2018b) Riemannian stochastic recursive gradient algorithm. In *Proceedings of the 35th International Conference on Machine Learning*. Volume **80** of Proceedings of Machine Learning Research. PMLR, pp. 2516–2524.

Lacoste-Julien, S. (2016) Convergence rate of Frank–Wolfe for non-convex objectives. arXiv preprint, arXiv:1607.00345.

Lacoste-Julien, S. & Jaggi, M. (2015) On the global linear convergence of Frank–Wolfe optimization variants. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1, NIPS'15. Cambridge, MA, USA: MIT Press, pp. 496–504.

Liu, C. & Boumal, N. (2019) Simple algorithms for optimization on Riemannian manifolds with constraints. *Appl. Math. Optim.*

Mariet, Z. & Sra, S. (2015) Fixed-point algorithms for learning determinantal point processes. *Proceedings of the 32nd International Conference on Machine Learning*. Volume 37 of Proceedings of Machine Learning Research (F. Bach & D. Blei eds). Lille, France: PMLR, pp. 2389–2397.

Nemirovskiĭ, A. & Yudin, D. (1983) *Problem Complexity and Method Efficiency in Optimization*. Wiley.

Nguyen, L. M., Liu, J., Scheinberg, K. & Takáč, M. (2017). Sarah: a novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, ICML'17, pp. 2613–2621. JMLR.org.

Nickel, M. & Kiela, D. (2017) Poincaré embeddings for learning hierarchical representations. *Advances in Neural Information Processing Systems*, pp. 6338–6347.

Reddi, S. J., Sra, S., Póczos, B. & Smola, A. (2016) Stochastic Frank–Wolfe methods for nonconvex optimization. *The 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1244–1251.

Sala, F., De Sa, C., Gu, A. & Re, C. (2018) Representation tradeoffs for hyperbolic embeddings. *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 4460–4469.

Sato, H., Kasai, H. & Mishra, B. (2017) Riemannian stochastic variance reduced gradient. arXiv preprint, arXiv:1702.05594.

Stegle, O., Lippert, C., Mooij, J. M., Lawrence, N. D. & Borgwardt, K. (2011). Efficient inference in matrix-variate gaussian models with \iid observation noise (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & Q. Weinberger, K. eds). *Advances in Neural Information Processing Systems* 24, pp. 630–638.

Tripuraneni, N., Flammarion, N., Bach, F. & Jordan, M. I. (2018) Averaging stochastic gradient descent on Riemannian manifolds. *Proc. Mach. Learn. Res.*, **75**, 1–38.

Udriste, C. (1994) *Convex Functions and Optimization Methods on Riemannian Manifolds*. Springer Science & Business Media.

Vandereycken, B. (2013) Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, **23**, 1214–1236.

Weber, M. (2020) Neighborhood growth determines geometric priors for relational representation learning. *The 23nd International Conference on Artificial Intelligence and Statistics*.

Weber, M. & Sra, S. (2017) Frank–Wolfe methods for geodesically convex optimization with application to the matrix geometric mean. arXiv:1710.10770.

Yuan, X., Huang, W., Absil, P.-A., & Gallivan, K. A. (2016) *Procedia Computer Science*.

Zhang, H., Reddi, J., & Sra, S. (2016) Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon & R. Garnett eds). *Advances in Neural Information Processing Systems* 29, pp. 4592–4600.

Zhang, H. & Sra, S. (2016) First-order methods for geodesically convex optimization. *The 29th Annual Conference on Learning Theory*. Volume 49 of Proceedings of Machine Learning Research. PMLR, pp. 1617–1638.

Zhang, T. (2017) A majorization-minimization algorithm for computing the Karcher mean of positive definite matrices. *SIAM J. Matrix Anal. Appl.*, **38**, 387–400.

Zhang, J., Zhang, H., & Sra, S. (2018a) R-SPIDER: a fast Riemannian stochastic optimization algorithm with curvature independent rate. *CoRR*, abs/1811.04194.

Zhou, P., Yuan, X.-T., & Feng, J. (2018b) Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. arXiv preprint, arXiv:1811.08109.