GRADIENT DESCENT ASCENT FOR MIN-MAX PROBLEMS ON RIEMANNIAN MANIFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

In the paper, we study a class of useful non-convex minimax optimization problems on the Riemanian manifold and propose a class of Riemanian gradient descent ascent algorithms to solve these minimax problems. Specifically, we propose a new Riemannian gradient descent ascent (RGDA) algorithm for the deterministic minimax optimization. Moreover, we prove that the RGDA has a sample complexity of $O(\kappa^2\epsilon^{-2})$ for finding an ϵ-stationary point of the nonconvex strongly-concave minimax problems, where κ denotes the condition number. At the same time, we introduce a Riemannian stochastic gradient descent ascent (RSGDA) algorithm for the stochastic minimax optimization. In the theoretical analysis, we prove that the RSGDA can achieve a sample complexity of $O(\kappa^4\epsilon^{-4})$. To further reduce the sample complexity, we propose a novel momentum variance-reduced Riemannian stochastic gradient descent ascent (MVR-RSGDA) algorithm based on a new momentum variance-reduced technique of STORM. We prove that the MVR-RSGDA algorithm achieves a lower sample complexity of $\tilde{O}(\kappa^3\epsilon^{-3})$ without large batches, which reaches the best known sample complexity for its Euclidean counterparts. In particular, the MVR-RSGDA does not rely on large batch size and only requires one sample at each iteration. This is the first study of the minimax optimization over the Riemannian manifold. Extensive experimental results on the robust deep neural networks training over Stiefel manifold demonstrate the efficiency of our proposed algorithms.

1 INTRODUCTION

In this paper, we study a class of useful non-convex minimax (*a.k.a.* min-max) problems on the Riemannian manifold $\mathcal{M}$ with the definition as:

$$\min_{x\in\mathcal{M}} \max_{y\in\mathcal{Y}} f(x,y), \tag{1}$$

where the function $f(x,y)$ is μ-strongly concave in y but possibly nonconvex in x. Here $\mathcal{Y} \subseteq \mathbb{R}^d$ is a convex and closed set. $f(\cdot,y) : \mathcal{M} \to \mathbb{R}$ for all $y \in \mathcal{Y}$ is a smooth but possibly nonconvex real-valued function on manifold $\mathcal{M}$, and $f(x,\cdot) : \mathcal{Y} \to \mathbb{R}$ for all $x \in \mathcal{M}$ a smooth and (strongly)-concave real-valued function. In this paper, we mainly focus on the stochastic minimax optimization problem $f(x,y) := \mathbb{E}_{\xi\sim\mathcal{D}}[f(x,y;\xi)]$, where ξ is a random variable that follows an unknown distribution $\mathcal{D}$. In fact, the problem (1) is associated to many existing machine learning applications:

1). Robust Training DNNs over Riemannian manifold. Deep Neural Networks (DNNs) recently have been demonstrating exceptional performance on many machine learning applications. However, they are vulnerable to the adversarial example attacks, which show that a small perturbation in the data input can significantly change the output of DNNs. Thus, the security properties of DNNs have been widely studied. One of secured DNN research topics is to enhance the robustness of DNNs under the adversarial example attacks. To be more specific, given training data $\mathcal{D} := \{\xi_i = (a_i, b_i)\}_{i=1}^n$, where $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ represent the features and label of sample ξ_i respectively. Each data sample a_i can be corrupted by a universal small perturbation vector y to generate an adversarial attack sample $a_i + y$. To make DNNs robust against adversarial attacks, one

popular approach is to solve the following robust training problem:

$$\min_{x} \max_{y \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(a_i + y; x), b_i) \,, \tag{2}$$

where x is the weight of the neural network; $h(\cdot; x)$ is the the deep neural network parameterized by x; and $\ell(\cdot)$ is the loss function. Here the constraint $\mathcal{Y} = \{y : \|y\| \leq \varepsilon\}$ indicates that the poisoned samples should not be too different from the original ones.

Recently, the orthonormality on weights of DNNs has gained much interest and has been found to be useful across different tasks such as person re-identification (Sun et al., 2017) and image classification (Xie et al., 2017). In fact, the orthonormality constraints improve the performances of DNNs (Li et al., 2020; Bansal et al., 2018), and reduce overfitting to improve generalization (Cogswell et al., 2015). At the same time, the orthonormality can stabilize the distribution of activations over layers within DNNs (Huang et al., 2018). Thus, we consider the following robust training problem over the Stiefel manifold $\mathcal{M}$:

$$\min_{x \in \mathcal{M}} \max_{y \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(a_i + y; x), b_i). \tag{3}$$

When data are continuously coming, we can rewrite the problem (3) as follows:

$$\min_{x \in \mathcal{M}} \max_{y \in \mathcal{Y}} \mathbb{E}_{\xi}[f(x, y; \xi)], \tag{4}$$

where $f(x, y; \xi) = \ell(h(a + y; x), b)$ with $\xi = (a, b)$.

2). Distributionally Robust Optimization over Riemannian manifold. Distributionally robust optimization (DRO) (Chen et al., 2017; Rahimian & Mehrotra, 2019) is an effective method to deal with the noisy data, adversarial data, and imbalanced data. At the same time, the DRO in the Riemannian manifold setting is also widely applied in machine learning problems such as robust principal component analysis (PCA). To be more specific, given a set of data samples $\{\xi_i\}_{i=1}^n$, the DRO over Riemannian manifold $\mathcal{M}$ can be written as the following minimax problem:

$$\min_{x \in \mathcal{M}} \max_{\mathbf{p} \in \mathcal{S}} \left\{ \sum_{i=1}^{n} p_i \ell(x; \xi_i) - \|\mathbf{p} - \frac{\mathbf{1}}{n}\|^2 \right\}, \tag{5}$$

where $\mathbf{p} = (p_1, \cdots, p_n)$, $\mathcal{S} = \{\mathbf{p} \in \mathbb{R}^n : \sum_{i=1}^n p_i = 1, p_i \geq 0\}$. Here $\ell(x; \xi_i)$ denotes the loss function over Riemannian manifold $\mathcal{M}$, which applies to many machine learning problems such as PCA (Han & Gao, 2020), dictionary learning (Sun et al., 2016), DNNs (Huang et al., 2018), among others. For example, the task of PCA can be cast on a Grassmann manifold.

So far, the existing minimax optimization methods such as gradient descent ascent method only focus on the minimax problems in the Euclidean space. Clearly, these minimax optimization methods are not effectively competent to solve the above minimax problem (1) on the Riemannian manifold. In this paper, thus, we propose a class of efficient Riemannian gradient descent ascent algorithms via using general retraction and vector transport. When the problem (1) is deterministic, we propose a new deterministic Riemannian gradient descent ascent algorithm. When the problem (1) is stochastic, we propose two efficient stochastic Riemannian gradient descent ascent algorithms. Our main **contributions** can be summarized as follows:

1) We propose a new Riemannian gradient descent ascent (RGDA) algorithm for the deterministic minimax optimization problem (1). We prove that the RGDA has a sample complexity of $O(\kappa^2\epsilon^{-2})$ for finding an ϵ-stationary point.
2) We also propose a new Riemannian stochastic gradient descent ascent (RSGDA) algorithm for the stochastic minimax optimization. In the theoretical analysis, we prove that the SRGDA has a sample complexity of $O(\kappa^4\epsilon^{-4})$.
3) To further reduce the sample complexity, we introduce a novel momentum variance-reduced Riemannian stochastic gradient descent ascent (MVR-RSGDA) algorithm based on a new momentum variance-reduced technique of STORM. We prove that the MVR-RSGDA achieves a lower sample complexity of $\tilde{O}(\kappa^3\epsilon^{-3})$ (please see Table 1), which reaches the best known sample complexity for its Euclidean counterparts.
4) Extensive experimental results on the robust DNN training over Stiefel manifold demonstrate the efficiency of our proposed algorithms.

Table 1: Convergence properties comparison of our algorithms for obtaining an ϵ-stationary point of the min-max optimization problem (1). κ denotes the condition number of function $f(x,\cdot)$.

Problem	Algorithm	Learning Rate	Batch Size	Complexity
Deterministic	RGDA	Constant	–	$O(\kappa^2\epsilon^{-2})$
Stochastic	RSGDA	Constant	$O(\kappa^2\epsilon^{-2})$	$O(\kappa^4\epsilon^{-4})$
	MVR-RSGDA	Decrease	$O(1)$	$\tilde{O}(\kappa^3\epsilon^{-3})$

2 Related Works

In this section, we briefly review the minimax optimization and Riemannian manifold optimization, respectively.

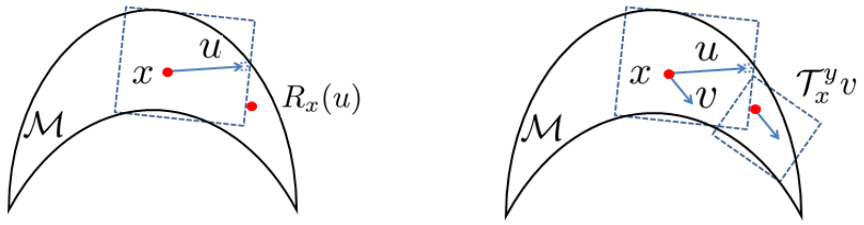
2.1 Minimax Optimization

Minimax optimization recently has been widely applied in many machine learning problems such as adversarial training (Goodfellow et al., 2014; Liu et al., 2019), reinforcement learning (Zhang et al., 2019; 2020), and distribution learning (Razaviyayn et al., 2020). At the same time, many efficient min-max methods (Rafique et al., 2018; Lin et al., 2019; Nouiehed et al., 2019; Thekumparampil et al., 2019; Lin et al., 2020; Yang et al., 2020; Ostrovskii et al., 2020; Yan et al., 2020; Xu et al., 2020a; Luo et al., 2020; Xu et al., 2020b; Boţ & Böhm, 2020; Huang et al., 2020) have been proposed for solving these minimax optimization problems. For example, Thekumparampil et al. (2019) has proposed a class of efficient dual implicit accelerated gradient algorithms to solve smooth min-max optimization. Lin et al. (2019) has proposed a class of efficient gradient decent ascent methods for non-convex minimax optimization. Subsequently, accelerated first-order algorithms Lin et al. (2020) have been proposed for minimax optimization. Xu et al. (2020b) proposed a unified single-loop alternating gradient projection algorithm for (non)convex-(non)concave minimax problems. Ostrovskii et al. (2020) proposed an efficient algorithm for finding first-order Nash equilibria in nonconvex concave minimax problems. Xu et al. (2020a); Luo et al. (2020) proposed a class of fast stochastic variance-reduced GDA algorithms to solve the stochastic minimax problems. More recently, Huang et al. (2020) has presented a class of new momentum-based first-order and zeroth-order descent ascent method for the nonconvex strongly concave minimax problems.

2.2 Riemannian Manifold Optimization

Riemannian manifold optimization are widely applied in many machine learning problems including dictionary learning (Sun et al., 2016), matrix factorization (Vandereycken, 2013), and deep neural networks (Huang et al., 2018). Thus, many Riemannian optimization methods recently have been proposed. For example, Zhang & Sra (2016); Liu et al. (2017) have proposed some efficient first-order gradient methods for geodesically convex functions. Subsequently, (Zhang et al., 2016) has presented fast stochastic variance-reduced methods to Riemannian manifold optimization. More recently, Sato et al. (2019) has proposed fast first-order gradient algorithms for Riemannian manifold optimization by using general retraction and vector transport. Subsequently, based on these retraction and vector transport, some fast Riemannian gradient-based methods (Zhang et al., 2018; Kasai et al., 2018; Zhou et al., 2019; Han & Gao, 2020) have been proposed for non-convex optimization. In addition, Riemannian Adam-type algorithms Kasai et al. (2019) have been proposed for matrix manifold optimization.

Notations: $\|\cdot\|$ denotes the ℓ_2 norm for vectors and spectral norm for matrices. $\langle x, y\rangle$ denotes the inner product of two vectors x and y. For function $f(x,y)$, $f(x,\cdot)$ denotes function *w.r.t.* the second variable with fixing x, and $f(\cdot,y)$ denotes function *w.r.t.* the first variable with fixing y. Let $\nabla_x f(x,y)$ and $\nabla_y f(x,y)$ denote the partial gradients *w.r.t.* variables x and y of function $f(x,y)$, respectively. Given a convex closed set $\mathcal{Y}$, we define a projection operation on the set $\mathcal{Y}$ as $\mathcal{P}_{\mathcal{Y}}(y_0) = \arg\max_{y\in\mathcal{Y}} \frac{1}{2}\|y-y_0\|^2$. We denote $a = O(b)$ if $a \leq Cb$ for some constant $C > 0$, and the notation $\tilde{O}(\cdot)$ hides logarithmic terms. I_d denotes the identity matrix with d dimension. The operation $\bigoplus$ denotes the Whitney sum.



(a) Retraction R_x (b) Vector Transport $\mathcal{T}_x^y$

Figure 1: Illustration of manifold operations.(a) A vector u in $T_x\mathcal{M}$ is mapped to $R_x(u)$ in $\mathcal{M}$; (b) A vector v in $T_x\mathcal{M}$ is transported to $T_y\mathcal{M}$ by $\mathcal{T}_x^y v$ (or $\mathcal{T}_u v$), where $y = R_x(u)$ and $u \in T_x\mathcal{M}$.

3 PRELIMINARIES

In this section, we first review some basic information on the Riemannian manifold $\mathcal{M}$. In general, the manifold $\mathcal{M}$ is endowed with a smooth inner product $\langle \cdot, \cdot \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ on tangent space $T_x\mathcal{M}$ for every $x \in \mathcal{M}$. The induced norm $\| \cdot \|_x$ of a tangent vector in $T_x\mathcal{M}$ is associated with the Riemannian metric. We first define a retraction $R_x : T_x\mathcal{M} \to \mathcal{M}$ maps tangent space $T_x\mathcal{M}$ onto $\mathcal{M}$ with a local rigidity condition that preserves the gradients at $x \in \mathcal{M}$ (Please see Fig.1 (a)). The retraction R_x satisfies all of the following: 1) $R_x(0) = x$, where $0 \in T_x\mathcal{M}$; 2) $\langle \nabla R_x(0), u \rangle_x = u$ for $u \in T_x\mathcal{M}$. In fact, exponential mapping Exp_x is a special case of retraction R_x that locally approximates the exponential mapping Exp_x up to the first order on the manifold.

Next, we define a vector transport $\mathcal{T} : T\mathcal{M} \bigoplus T\mathcal{M} \to T\mathcal{M}$ (Please see Fig.1 (b)) that satisfies all of the following 1) $\mathcal{T}$ has an associated retraction R, i.e., for $x \in \mathcal{M}$ and $w, u \in T_x\mathcal{M}$, $\mathcal{T}_u w$ is a tangent vector at $R_x(w)$; 2) $\mathcal{T}_0 v = v$; 3) $\mathcal{T}_u(av + bw) = a\mathcal{T}_u v + b\mathcal{T}_u w$ for all $a, b \in \mathbb{R}$ an $u, v, w \in T\mathcal{M}$. Vector transport $\mathcal{T}_x^y v$ or equivalently $\mathcal{T}_u v$ with $y = R_x(u)$ transports $v \in T_x\mathcal{M}$ along the retraction curve defined by direction u. Here we focus on the isometric vector transport $\mathcal{T}_x^y$, which satisfies $\langle u, v \rangle_x = \langle \mathcal{T}_x^y u, \mathcal{T}_x^y v \rangle_y$ for all $u, v \in T_x\mathcal{M}$.

Based on the above definitions, we give some standard assumptions about the problem (1).

Assumption 1. *Each component function $f(x, y)$ is continuously differentiable in both $x \in \mathcal{M}$ and $y \in \mathcal{Y}$, and there exist constants L_{11}, L_{12}, L_{21} and L_{22}, such that for every $x, x_1, x_2 \in \mathcal{M}$ and $y, y_1, y_2 \in \mathcal{Y}$, we have*

$$\|\nabla_x f(x_1, y; \xi) - \mathcal{T}_{x_2}^{x_1} \nabla_x f(x_2, y; \xi)\| \le L_{11}\|u\|, \ \|\nabla_x f(x, y_1; \xi) - \nabla_x f(x, y_2; \xi)\| \le L_{12}\|y_1 - y_2\|$$

$$\|\nabla_y f(x_1, y; \xi) - \mathcal{T}_{x_2}^{x_1} \nabla_y f(x_2, y; \xi)\| \le L_{21}\|u\|, \ \|\nabla_y f(x, y_1; \xi) - \nabla_y f(x, y_2; \xi)\| \le L_{22}\|y_1 - y_2\|$$

where $u \in T_{x_1}\mathcal{M}$ and $x_2 = R_{x_1}(u)$.

Assumption 1 implies the partial gradient $\nabla_x f(x, y; \xi)$ is retraction L_{11}-Lipschitz as in (Han & Gao, 2020) and $\nabla_y f(x, y; \xi)$ is L_{22}-Lipschitz continous. For the deterministic problem, let $f(x, y)$ instead of $f(x, y; \xi)$ in Assumption 1. Since $f(x, y)$ is strongly concave in $y \in \mathcal{Y}$, there exists a unique solution to the problem $\max_{y \in \mathcal{Y}} f(x, y)$ for any x. We define the function $\Phi(x) = \max_{y \in \mathcal{Y}} f(x, y)$ and $y^*(x) = \arg\max_{y \in \mathcal{Y}} f(x, y)$.

Assumption 2. *The function $\Phi(x)$ is retraction L-smooth. There exists a constant $L > 0$, for all $x \in \mathcal{M}, z = R_x(u)$ with $u \in T_x\mathcal{M}$, such that*

$$\Phi(z) \le \Phi(x) + \langle \nabla\Phi(x), u \rangle + \frac{L}{2}\|u\|^2. \tag{6}$$

Assumption 3. *The objective function $f(x, y)$ is μ-strongly concave w.r.t y, i.e., for any $x \in \mathcal{M}$*

$$f(x, y_1) \le f(x, y_2) + \langle \nabla_y f(x, y_2), y_1 - y_2 \rangle - \frac{\mu}{2}\|y_1 - y_2\|^2, \ \forall y_1, y_2 \in \mathcal{Y}. \tag{7}$$

Assumption 4. *The function $\Phi(x)$ is bounded from below in $\mathcal{M}$,* i.e., $\Phi^* = \inf_{x \in \mathcal{M}} \Phi(x)$.

Assumption 5. *The variance of stochastic gradient is bounded,* i.e., *there exists a constant $\sigma_1 > 0$ such that for all x, it follows $\mathbb{E}_\xi\|\nabla_x f(x, y; \xi) - \nabla_x f(x, y)\|^2 \le \sigma_1^2$; There exists a constant $\sigma_2 > 0$ such that for all y, it follows $\mathbb{E}_\xi\|\nabla_y f(x, y; \xi) - \nabla_y f(x, y)\|^2 \le \sigma_2^2$. We also define $\sigma = \max\{\sigma_1, \sigma_2\}$.*

Algorithm 1 RGDA and RSGDA Algorithms for Min-Max Optimization

1: **Input:** T, parameters $\{\gamma, \lambda, \eta_t\}_{t=1}^T$, mini-batch size B, and initial input $x_1 \in \mathcal{M}$, $y_1 \in \mathcal{Y}$;
2: **for** $t = 1, 2, \ldots, T$ **do**
3: **(RGDA)** Compute deterministic gradients

$$v_t = \nabla_x f(x_t, y_t),\ w_t = \nabla_y f(x_t, y_t); \tag{8}$$

4: **(RSGDA)** Draw B i.i.d. samples $\{\xi_t^i\}_{i=1}^B$, then compute stochastic gradients

$$v_t = \frac{1}{B}\sum_{i=1}^B \nabla_x f(x_t, y_t; \xi_t^i),\ w_t = \frac{1}{B}\sum_{i=1}^B \nabla_y f(x_t, y_t; \xi_t^i); \tag{9}$$

5: Update: $x_{t+1} = R_{x_t}(-\gamma\eta_t v_t)$;
6: Update: $\tilde{y}_{t+1} = \mathcal{P}_{\mathcal{Y}}(y_t + \lambda w_t)$ and $y_{t+1} = y_t + \eta_t(\tilde{y}_{t+1} - y_t)$;
7: **end for**
8: **Output:** x_ζ and y_ζ chosen uniformly random from $\{x_t, y_t\}_{t=1}^T$.

4 Riemanian Gradient Descent Ascent

In the section, we propose a class of Riemannian gradient descent ascent algorithm to solve the deterministic and stochastic minimax optimization problem (1), respectively.

4.1 RGDA and RSGDA Algorithms

In this subsection, we propose an efficient Riemannian gradient descent ascent (RGDA) algorithm to solve the deterministic min-max problem (1). At the same time, we propose a standard Riemannian stochastic gradient descent ascent (RSGDA) algorithm to solve the stochastic min-max problem (1). Algorithm 1 summarizes the algorithmic framework of the RGDA and RSGDA algorithms.

At the step 5 of Algorithm 1, we apply the retraction operator to ensure the variable x_t for all $t \geq 1$ in the manifold $\mathcal{M}$. At the step 6 of Algorithm 1, we use $0 < \eta_t \leq 1$ to ensure the variable y_t for all $t \geq 1$ in the convex constraint $\mathcal{Y}$.

Here we define a reasonable metric to measure the convergence:

$$\mathcal{H}_t = \|\nabla\Phi(x_t)\| + \tilde{L}\|y_t - y^*(x_t)\|, \tag{10}$$

where $\tilde{L} = \max(1, L_{11}, L_{12}, L_{21}, L_{22})$, and the first term of $\mathcal{H}_t$ measures convergence of the iteration solutions $\{x_t\}_{t=1}^T$, and the last term measures convergence of the iteration solutions $\{y_t\}_{t=1}^T$. Since the function $f(x, y)$ is strongly concave in $y \in \mathcal{Y}$, there exists a unique solution $y^*(x)$ to the problem $\max_{y\in\mathcal{Y}} f(x, y)$ for any $x \in \mathcal{M}$. Thus, we apply the standard metric $\|y_t - y^*(x_t)\|$ to measure convergence of the parameter y. Given $y = y^*(x_t)$, we use the standard metric $\|\nabla\Phi(x_t)\| = \|\nabla_x f(x_t, y^*(x_t))\|$ to measure convergence of the parameter x. Note that we use the coefficient $\tilde{L}$ to balance the scale of metrics of the variable x and the variable y.

4.2 MVR-RSGDA algorithm

In this subsection, we propose a novel momentum variance-reduced stochastic Riemannian gradient descent ascent (MVR-RSGDA) algorithm to solve the stochastic min-max problem (1), which builds on the momentum-based variance reduction technique of STORM (Cutkosky & Orabona, 2019). Algorithm 2 describes the algorithmic framework of MVR-RSGDA method.

In Algorithm 2, we use the momentum-based variance-reduced technique of STORM to update stochastic Riemannian gradient v_t:

$$\begin{aligned} v_{t+1} &= \alpha_{t+1}\underbrace{\nabla_x f(x_{t+1}, y_{t+1}; \xi_{t+1})}_{\text{SGD}} + (1-\alpha_{t+1})\underbrace{\big(\nabla_x f(x_{t+1}, y_{t+1}; \xi_{t+1}) - \mathcal{T}_{x_t}^{x_{t+1}}\big(\nabla_x f(x_t, y_t; \xi_t) - v_t\big)\big)}_{\text{SPIDER}} \\ &= \nabla_x f(x_{t+1}, y_{t+1}; \xi_{t+1}) + (1-\alpha_{t+1})\mathcal{T}_{x_t}^{x_{t+1}}\big(v_t - \nabla_x f(x_t, y_t; \xi_t)\big), \end{aligned} \tag{13}$$

Algorithm 2 MVR-RSGDA Algorithm for Min-Max Optimization

Input: T, parameters $\{\gamma, \lambda, b, m, c_1, c_2\}$ and initial input $x_1 \in \mathcal{M}$ and $y_1 \in \mathcal{Y}$;
Draw a sample ξ_1, then compute $v_1 = \nabla_x f(x_1, y_1; \xi_1)$ and $w_1 = \nabla_y f(x_1, y_1; \xi_1)$;
for $t = 1, 2, \ldots, T$ **do**
  Compute $\eta_t = \frac{b}{(m+t)^{1/3}}$;
  Update: $x_{t+1} = R_{x_t}(-\gamma\eta_t v_t)$;
  Update: $\tilde{y}_{t+1} = \mathcal{P}_{\mathcal{Y}}(y_t + \lambda w_t)$ and $y_{t+1} = y_t + \eta_t(\tilde{y}_{t+1} - y_t)$;
  Compute $\alpha_{t+1} = c_1\eta_t^2$ and $\beta_{t+1} = c_2\eta_t^2$;
  Draw a sample ξ_{t+1}, then compute

$$v_{t+1} = \nabla_x f(x_{t+1}, y_{t+1}; \xi_{t+1}) + (1 - \alpha_{t+1})\mathcal{T}_{x_t}^{x_{t+1}}\big[v_t - \nabla_x f(x_t, y_t; \xi_{t+1})\big], \tag{11}$$

$$w_{t+1} = \nabla_y f(x_{t+1}, y_{t+1}; \xi_{t+1}) + (1 - \beta_{t+1})\big[w_t - \nabla_y f(x_t, y_t; \xi_{t+1})\big]; \tag{12}$$

end for
Output: x_ζ and y_ζ chosen uniformly random from $\{x_t, y_t\}_{t=1}^T$.

where $\alpha_{t+1} \in (0, 1]$. When $\alpha_{t+1} = 1$, v_t will degenerate a vanilla stochastic Riemannian gradient; When $\alpha_{t+1} = 0$, v_t will degenerate a stochastic Riemannian gradient based on variance-reduced technique of SPIDER (Nguyen et al., 2017; Fang et al., 2018). Similarly, we use this momentum-based variance-reduced technique to estimate the stochastic gradient w_t.

5 Convergence Analysis

In this section, we study the convergence properties of our SGDA, RSGDA and MVR-RSGDA algorithms under some mild conditions. For notational simplicity, let $\tilde{L} = \max(1, L_{11}, L_{12}, L_{21}, L_{22})$ and $\kappa = L_{21}/\mu$ denote the number condition of function $f(x, y)$. We first give a useful lemma.

Lemma 1. *Under the above assumptions, the gradient of function $\Phi(x) = \max_{y\in\mathcal{Y}} f(x, y)$ is retraction G-Lipschitz, and the mapping or function $y^*(x) = \arg\max_{y\in\mathcal{Y}} f(x, y)$ is retraction κ-Lipschitz. Given any $x_1, x_2 = R_{x_1}(u) \in \mathcal{M}$ and $u \in T_{x_1}\mathcal{M}$, we have*

$$\|\nabla\Phi(x_1) - \mathcal{T}_{x_2}^{x_1}\nabla\Phi(x_2)\| \leq G\|u\|,$$
$$\|y^*(x_1) - y^*(x_2)\| \leq \kappa\|u\|,$$

where $G = \kappa L_{12} + L_{11}$ and $\kappa = L_{21}/\mu$.

5.1 Convergence Analysis of both the RGDA and RSGDA Algorithms

In the subsection, we study the convergence properties of deterministic RGDA and stochastic RSGDA algorithms. The related proofs of RGDA and RSGDA are provided in Appendix A1.

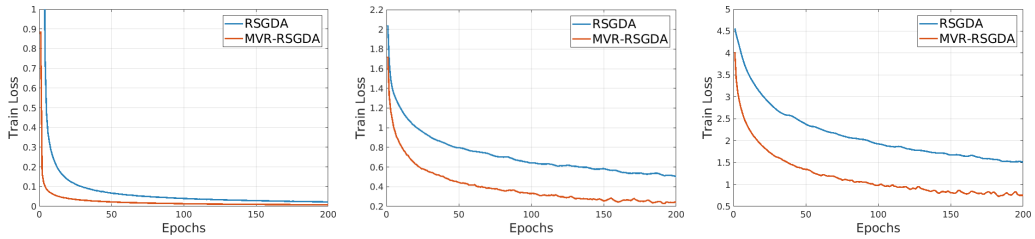
Theorem 1. *Suppose the sequence $\{x_t, y_t\}_{t=1}^T$ be generated from Algorithm 1 by using deterministic gradients. Given $\eta = \eta_t$ for all $t \geq 1$, $0 < \eta \leq \min(1, \frac{1}{2\gamma L})$, $0 < \lambda \leq \frac{1}{6\tilde{L}}$ and $0 < \gamma \leq \frac{\mu\lambda}{10\tilde{L}\kappa}$, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla\Phi(x_t)\| + \tilde{L}\|y_t - y^*(x_t)\|\big] \leq \frac{2\sqrt{\Phi(x_1) - \Phi^*}}{\sqrt{\gamma\eta T}}. \tag{14}$$

Remark 1. *Since $0 < \eta \leq \min(1, \frac{1}{2\gamma L})$ and $0 < \gamma \leq \frac{\mu\lambda}{10\tilde{L}\kappa}$, we have $0 < \eta\gamma \leq \min(\frac{\mu\lambda}{10\tilde{L}\kappa}, \frac{1}{2L})$. Let $\eta\gamma = \min(\frac{\mu\lambda}{10\tilde{L}\kappa}, \frac{1}{2L})$, we have $\eta\gamma = O(\frac{1}{\kappa^2})$. The RGDA algorithm has convergence rate of $O\big(\frac{\kappa}{T^{1/2}}\big)$. By $\frac{\kappa}{T^{1/2}} \leq \epsilon$, i.e., $\mathbb{E}[\mathcal{H}_\zeta] \leq \epsilon$, we choose $T \geq \kappa^2\epsilon^{-2}$. In the deterministic RGDA Algorithm, we need one sample to estimate the gradients v_t and w_t at each iteration, and need T iterations. Thus, the RGDA reaches a sample complexity of $T = O(\kappa^2\epsilon^{-2})$ for finding an ϵ-stationary point.*

Theorem 2. *Suppose the sequence $\{x_t, y_t\}_{t=1}^T$ be generated from Algorithm 1 by using stochastic gradients. Given $\eta = \eta_t$ for all $t \geq 1$, $0 < \eta \leq \min(1, \frac{1}{2\gamma L})$, $0 < \lambda \leq \frac{1}{6\tilde{L}}$ and $0 < \gamma \leq \frac{\mu\lambda}{10\tilde{L}\kappa}$, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla\Phi(x_t)\| + \tilde{L}\|y_t - y^*(x_t)\|\big] \leq \frac{2\sqrt{\Phi(x_1) - \Phi^*}}{\sqrt{\gamma\eta T}} + \frac{\sqrt{2}\sigma}{\sqrt{B}} + \frac{5\sqrt{2}\tilde{L}\sigma}{\sqrt{B}\mu}. \tag{15}$$



(a) MNIST (b) CIFAR-10 (c) CIFAR-100

Figure 2: Training loss of robust training DNNs with orthogonality regularization on weights.

Remark 2. *Since* $0 < \eta \leq \min(1, \frac{1}{2\gamma L})$ *and* $0 < \gamma \leq \frac{\mu\lambda}{10\tilde{L}\kappa}$, *we have* $0 < \eta\gamma \leq \min(\frac{\mu\lambda}{10\tilde{L}\kappa}, \frac{1}{2L})$. *Let* $\eta\gamma = \min(\frac{\mu\lambda}{10\tilde{L}\kappa}, \frac{1}{2L})$, *we have* $\eta\gamma = O(\frac{1}{\kappa^2})$. *Let* $B = T$, *the RSGDA algorithm has convergence rate of* $O\big(\frac{\kappa}{T^{1/2}}\big)$. *By* $\frac{\kappa}{T^{1/2}} \leq \epsilon$, *i.e.,* $\mathbb{E}[\mathcal{H}_\zeta] \leq \epsilon$, *we choose* $T \geq \kappa^2\epsilon^{-2}$. *In the stochastic RSGDA Algorithm, we need* B *samples to estimate the gradients* v_t *and* w_t *at each iteration, and need* T *iterations. Thus, the RSGDA reaches a sample complexity of* $B * T = O(\kappa^4\epsilon^{-4})$ *for finding an* ϵ*-stationary point.*

5.2 Convergence Analysis of the MVR-RSGDA Algorithm

In the subsection, we provide the convergence properties of the MVR-RSGDA algorithm. The related proofs of MVR-RSGDA are provided in Appendix A2.

Theorem 3. *Suppose the sequence* $\{x_t, y_t\}_{t=1}^T$ *be generated from Algorithm 2. Given* $y_1 = y^*(x_1)$, $c_1 \geq \frac{2}{3b^3} + 2\lambda\mu$, $c_2 \geq \frac{2}{3b^3} + \frac{50\lambda\tilde{L}^2}{\mu}$, $b > 0$, $m \geq \max\big(2, (\tilde{c}b)^3\big)$, $0 < \gamma \leq \frac{\mu\lambda}{2\kappa\tilde{L}\sqrt{25+4\mu\lambda}}$ *and* $0 < \lambda \leq \frac{1}{6\tilde{L}}$, *we have*

$$\frac{1}{T}\sum_{t=1}^{T}\big(\|\nabla\Phi(x_t)\| + \tilde{L}\|y_t - y^*(x_t)\|\big) \leq \frac{\sqrt{2M'}m^{1/6}}{T^{1/2}} + \frac{\sqrt{2M'}}{T^{1/3}}, \tag{16}$$

where $\tilde{c} = \max(1, c_1, c_2, 2\gamma L)$ *and* $M' = \frac{2(\Phi(x_1)-\Phi^*)}{\gamma b} + \frac{2\sigma^2}{\lambda\mu\eta_0 b} + \frac{2(c_1^2+c_2^2)\sigma^2 b^2}{\lambda\mu}\ln(m+T)$.

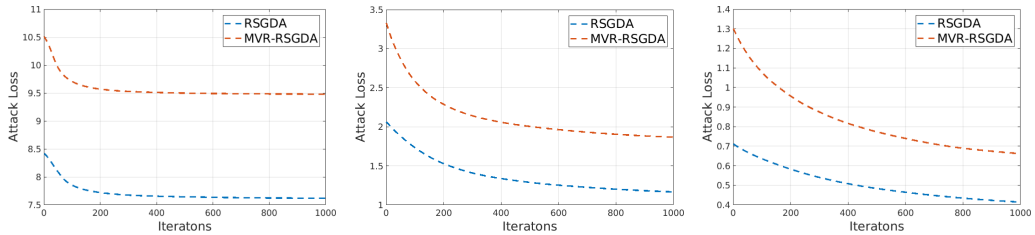
Remark 3. *Let* $c_1 = \frac{2}{3b^3} + 2\lambda\mu$, $c_2 = \frac{2}{3b^3} + \frac{50\lambda\tilde{L}^2}{\mu}$, $\lambda = \frac{1}{6\tilde{L}}$, $\gamma = \frac{\mu\lambda}{2\kappa\tilde{L}\sqrt{25+4\mu\lambda}}$, *it is easy verified that* $\gamma = O(\frac{1}{\kappa})$, $c_1 = O(1)$ *and* $c_2 = O(\kappa)$. *So we have* $M' = O\big(\kappa^2\ln(T)\big)$. *Thus, the MVR-RSGDA algorithm has convergence rate of* $\tilde{O}\big(\frac{\kappa}{T^{1/3}}\big)$. *By* $\frac{\kappa}{T^{1/3}} \leq \epsilon$, *i.e.,* $\mathbb{E}[\mathcal{H}_\zeta] \leq \epsilon$, *we choose* $T \geq \kappa^3\epsilon^{-3}$. *In Algorithm 2, we only require one sample to estimate the stochastic gradients* v_t *and* w_t *at each iteration, and need* T *iterations. Thus, the MVR-RSGDA has a sample complexity of* $T = \tilde{O}\big(\kappa^3\epsilon^{-3}\big)$ *for finding an* ϵ*-stationary point.*

6 Experiments

In this section, we apply deep neural networks (DNNs) robust training over the Stiefel manifold $\mathrm{St}(r, d) = \{W \in \mathbb{R}^{d\times r} : W^TW = I_r\}$ to evaluate the performance of our algorithms. In the experiment, we use MNIST, CIFAR-10, and CIFAR-100 datasets to train the model (More results on SVHN, STL10, and FashionMNIST datasets are provided in the Appendix B). Considering the sample size is large in these datasets, we only compare the proposed stochastic algorithms (RSGDA and MVR-RSGDA) in the experiments.

6.1 Experimental Setting

Given a deep neural network $h(\cdot; x)$ parameterized by x as shown in the above problem (2), the weights of l-th layer is $x_i \in \mathrm{St}(n_{\text{in}}^l, n_{\text{out}}^l)$, where $\mathrm{St}(n_{\text{in}}^l, n_{\text{out}}^l)$ is the Stiefel manifold of l-th layer. For the weights in dense layers, $n_{\text{in}}^l, n_{\text{out}}^l$ are the number of inputs and outputs neurons. For the weights in convolution layers, n_{in}^l is the number of input channels, n_{out}^l is the product of the number of output channels and kernel sizes. Note that the trainable parameters from other components (e.g. batchnorm) are not in Stiefel manifold.



(a) MNIST (b) CIFAR-10 (c) CIFAR-100

Figure 3: Attack loss when using uniform attack on DNNs trained by RSGDA and MVR-RSGDA.

Table 2: Test accuracy against nature images and uniform attack for MNIST, CIFAR-10 and CIFAR-100 datasets.

Method	Eval. Against	MNIST	CIFAR-10	CIFAR-100
RSGDA	Nat. Images	98.92%	62.87%	29.92%
	Uniform Attack	98.58%	61.45%	31.14%
MVR-RSGDA	Nat. Images	**99.15%**	**67.45%**	**32.92%**
	Uniform Attack	**99.03%**	**68.56%**	**34.69%**

For both RSGDA and MVR-RSGDA algorithms, we set $\{\gamma, \lambda\}$ to $\{1.0, 0.1\}$. We further set $\{b, m, c_1, c_2\}$ to 0.5, 8, 512, 512 for MVR-RSGDA. η in RSGDA is set to 0.01. For both algorithms, the mini-batch size is set to 512. We set ϵ for y as 0.05 and 0.03 for the MNIST dataset and CIFAR-10/100 datasets. The above settings are the same for all datasets. An 8-layer (5 convolution layers and 3 dense layers) deep neural network is used in all experiments. All codes are implemented with McTorch (Meghwanshi et al., 2018) which is based on PyTorch (Paszke et al., 2019).

6.2 Experimental Results

The training loss of the robust training problem in the above Eq. (2) is shown in Fig. 2. From the figure, we can see that MVR-RSGDA enjoys a faster convergence speed compared to the baseline RSGDA. It's also clear that when the dataset becomes complicate (from MNIST to CIFAR-10/100), the advantage of MVR-RSGDA becomes larger.

When it comes to robust training, the training loss is not enough to identify which algorithm is better. We also use a variant of uniform perturbation to attack the model trained by our algorithms. We follow the design of uniform attack in previous works (Moosavi-Dezfooli et al., 2017; Chaubey et al., 2020), and the detail uniform attack objective is shown below:

$$\min_{y \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^{n} \max\left(h_{b_i}(y + a_i) - \max_{j \neq b_i} h_j(y + a_i), 0\right), \quad \text{s.t. } \mathcal{Y} = \{\|y\|_\infty \leq \varepsilon\} \tag{17}$$

where h_j is the j-th logit of the output from the deep neural network, and y here is a uniform permutation added for all inputs. In practice, we sample a mini-batch with 512 samples at each iteration. The optimization of the uniform permutation lasts for 1000 iterations for all settings. The attack loss is presented in Fig 3. The attack loss for the model trained by MVR-RSGDA is higher compared to RSGDA, which indicates the model trained by MVR-RSGDA is harder to attack and thus more robust. The test accuracy with natural image and uniform attack is shown in Tab. 2, which also suggests the advantage of MVR-RSGDA. More results are provided in Appendix B

7 Conclusion

In the paper, we studied a class of useful min-max optimization problems on the Riemanian manifold. Moreover, we proposed a class of efficient Riemanian gradient descent ascent algorithms to solve these minimax problems, and studied the convergence properties of the proposed algorithms. For example, we proved that our MVR-RSGDA algorithm achieves a sample complexity of $\tilde{O}(\kappa^3\epsilon^{-3})$ without large batches, which reaches the best known sample complexity for its Euclidean counterparts. As far as we know, our methods are the first work to bridge the popular minimax optimization with manifold optimization.

REFERENCES

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In *Advances in Neural Information Processing Systems*, pp. 4261–4271, 2018.

Radu Ioan Boţ and Axel Böhm. Alternating proximal-gradient steps for (stochastic) nonconvex-concave minimax problems. *arXiv preprint arXiv:2007.13605*, 2020.

Ashutosh Chaubey, Nikhil Agrawal, Kavya Barnwal, Keerat K Guliani, and Pramod Mehta. Universal adversarial perturbations: A survey. *arXiv preprint arXiv:2005.08087*, 2020.

Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems*, pp. 4705–4714, 2017.

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, pp. 15210–15219, 2019.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Andi Han and Junbin Gao. Riemannian stochastic recursive momentum method for non-convex optimization. *arXiv preprint arXiv:2008.04555*, 2020.

Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170*, 2020.

Lei Huang, Xianglong Liu, Bo Lang, Adams Wei Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 3271–3278. AAAI Press, 2018.

Hiroyuki Kasai, Hiroyuki Sato, and Bamdev Mishra. Riemannian stochastic recursive gradient algorithm. In *International Conference on Machine Learning*, pp. 2516–2524, 2018.

Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. *arXiv preprint arXiv:1902.01144*, 2019.

Jun Li, Li Fuxin, and Sinisa Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *arXiv preprint arXiv:2002.01113*, 2020.

Tianyi Lin, Chi Jin, and Michael I Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331*, 2019.

Tianyi Lin, Chi Jin, Michael Jordan, et al. Near-optimal algorithms for minimax optimization. *arXiv preprint arXiv:2002.02417*, 2020.

Mingrui Liu, Youssef Mroueh, Jerret Ross, Wei Zhang, Xiaodong Cui, Payel Das, and Tianbao Yang. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. *arXiv preprint arXiv:1912.11940*, 2019.

Yuanyuan Liu, Fanhua Shang, James Cheng, Hong Cheng, and Licheng Jiao. Accelerated first-order methods for geodesically convex optimization on riemannian manifolds. In *Advances in Neural Information Processing Systems*, pp. 4868–4877, 2017.

Luo Luo, Haishan Ye, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *arXiv preprint arXiv:2001.03724*, 2020.

Mayank Meghwanshi, Pratik Jawanpuria, Anoop Kunchukuttan, Hiroyuki Kasai, and Bamdev Mishra. Mctorch, a manifold optimization library for deep learning. *arXiv preprint arXiv:1810.01811*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pp. 2613–2621, 2017.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, pp. 14934–14942, 2019.

Dmitrii M Ostrovskii, Andrew Lowy, and Meisam Razaviyayn. Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems. *arXiv preprint arXiv:2002.07919*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019.

Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Meisam Razaviyayn, Tianjian Huang, Songtao Lu, Maher Nouiehed, Maziar Sanjabi, and Mingyi Hong. Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances. *IEEE Signal Processing Magazine*, 37(5):55–66, 2020.

Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM Journal on Optimization*, 29(2):1444–1472, 2019.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method. *IEEE Transactions on Information Theory*, 63(2):885–914, 2016.

Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3800–3808, 2017.

Kiran K Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Efficient algorithms for smooth minimax optimization. In *Advances in Neural Information Processing Systems*, pp. 12680–12691, 2019.

Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6176–6185, 2017.

Tengyu Xu, Zhe Wang, Yingbin Liang, and H Vincent Poor. Enhanced first and zeroth order variance reduced algorithms for min-max optimization. *arXiv preprint arXiv:2006.09361*, 2020a.

Zi Xu, Huiling Zhang, Yang Xu, and Guanghui Lan. A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems. *arXiv preprint arXiv:2006.02032*, 2020b.

Yan Yan, Yi Xu, Qihang Lin, Wei Liu, and Tianbao Yang. Sharp analysis of epoch stochastic gradient descent ascent methods for min-max optimization. *arXiv preprint arXiv:2002.05309*, 2020.

Junchi Yang, Negar Kiyavash, and Niao He. Global convergence and variance-reduced optimization for a class of nonconvex-nonconcave minimax problems. *arXiv preprint arXiv:2002.09621*, 2020.

Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pp. 1617–1638, 2016.

Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. In *Advances in Neural Information Processing Systems*, pp. 4592–4600, 2016.

Jingzhao Zhang, Hongyi Zhang, and Suvrit Sra. R-spider: A fast riemannian stochastic optimization algorithm with curvature independent rate. *arXiv preprint arXiv:1811.04194*, 2018.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Kaiqing Zhang, Sham M Kakade, Tamer Başar, and Lin F Yang. Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. *arXiv preprint arXiv:2007.07461*, 2020.

Pan Zhou, Xiaotong Yuan, Shuicheng Yan, and Jiashi Feng. Faster first-order methods for stochastic non-convex optimization on riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 2019.