Blind Demixing for Low-Latency Communication

Jialin Dong, *Student Member, IEEE*, Kai Yang, *Student Member, IEEE*, and Yuanming Shi, *Member, IEEE*

Abstract— In next-generation wireless networks, low-latency communication is critical to support emerging diversified applications, e.g., tactile Internet and virtual reality. In this paper, a novel blind demixing approach is developed to reduce the channel signaling overhead, thereby supporting low-latency communication. Specifically, we develop a low-rank approach to recover the original information only based on the single observed vector without any channel estimation. To address the unique challenges of multiple non-convex rank-one constraints, the quotient manifold geometry of the product of complex symmetric rank-one matrices is exploited. This is achieved by equivalently reformulating the original problem that uses complex asymmetric matrices to the one that uses Hermitian positive semidefinite matrices. We further generalize the geometric concepts of the complex product manifold via element-wise extension of the geometric concepts of the individual manifolds. The scalable Riemannian optimization algorithms, i.e., the Riemannian gradient descent algorithm and the Riemannian trust-region algorithm, are then developed to solve the blind demixing problem efficiently with low iteration complexity and low iteration cost. The statistical analysis shows that the Riemannian gradient descent with spectral initialization is guaranteed to linearly converge to the ground truth signals provided sufficient measurements. In addition, the Riemannian trust-region algorithm is provable to converge to an approximate local minimum from the arbitrary initialization point. Numerical experiments have been carried out in settings with different types of encoding matrices to demonstrate the algorithmic advantages, performance gains, and sample efficiency of the Riemannian optimization algorithms.

Index Terms— Blind demixing, low-latency communication, low-rank optimization, product manifold, Riemannian optimization.

I. INTRODUCTION

RECENTLY, various emerging 5G applications such as Internet-of-Things (IoT) [1], Tactile Internet [2] and Virtual Reality [3] are unleashing a sense of urgency in providing *low-latency* communications [4], for which innovative new technologies need to be developed. To achieve this goal, various solutions have been investigated, which can be typically categorized into three main types, i.e., radio access network (RAN), core network, as well as mobile edge caching and computing [5]. In particular, by pushing the computation and storage resources to the network edge, followed by network densification, dense Fog-RAN provides a principled way to reduce the latency [6]. In addition, reducing the packet blocklength, e.g., short packets communication [7], is a promising technique in RAN to support low-latency communication, for which the theoretical analysis on the tradeoffs among the channel coding rate, blocklength and error probability was provided in [8].

However, channel signaling overhead reduction becomes critical to design a low latency communication system. In particular, when packet blocklength is reduced as envisioned in 5G systems, channel signaling overhead dominates the major portion of the packet [5]. Furthermore, massive channel acquisition overhead becomes the bottleneck for interference coordination in dense wireless networks [6]. To address this issue, numerous research efforts have been made on channel signaling overhead reduction. The compressed sensing based approach was developed in [9], yielding good performance with low energy, latency and bandwidth cost. The recent proposal of topological interference alignment [10] serves as a promising way to manage the interference based only on the network connectivity information at the transmitters. Furthermore, by equipping a large number of antennas at the base stations, massive MIMO [11] can manage the interference without channel estimation at the transmitters. However, all the methods [10], [11] still assume that the channel state information (CSI) is available for signal detection at the receivers.

More recently, a new proposal has emerged, namely, the mixture of blind deconvolution and demixing [12], i.e., *blind demixing* for brief, regarded as a promising solution to support the efficient low-latency communication without channel estimation at both transmitters and receivers. It also meets the demands for sporadic and short messages in next generation wireless networks [13]. In particular, blind deconvolution is a problem of estimating two unknown vectors from their convolution, which can be exploited in the context of channel coding for multipath channel protection [14]. However, the results of the blind deconvolution problem [14] cannot be directly extended to the blind demixing problem since only a single observed vector is available. Demixing refers to the problem of identifying multiple structured signals by given the mixture of measurements of these signals, which can be exploited in a secure communications protocol [15]. The measurement matrices in the demixing problem are normally assumed to be full-rank matrices to assist theoretical analysis [16]. However, the measurement matrices in the blind demixing problem are rank-one matrices [17], which hamper

the extension of results developed in [16] to the blind demixing problem.

In this paper, we consider the blind demixing problem in a specific scenario, i.e., an orthogonal frequency division multiplexing (OFDM) system and propose a low-rank approach to recover the original signals in this problem. However, the resulting rank-constrained optimization problem is known to be non-convex and highly intractable. A growing body of literature has proposed marvelous algorithms to deal with low-rank problems. In particular, convex relaxation approach is an effective way to solve this problem with theoretical guarantees [12]. However, it is not scalable to the medium- and large-scale problems due to the high iteration cost of the convex programming technique. To enable scalability, non-convex algorithms (e.g, regularized gradient descent algorithm [17] and iterative hard thresholding method [18]), endowed with lower iteration cost, have been developed. However, the overall computational complexity of these algorithms is still high due to the slow convergence rate, i.e., high iteration complexity.

To address the limitations of the existing algorithms for the blind demixing problem, we propose the Riemannian optimization algorithm over a product complex manifold in order to simultaneously reduce the iteration cost and iteration complexity. Specifically, the quotient manifold geometry of the product of complex symmetric rank-one matrices is exploited. This is achieved by equivalently reformulating the original complex asymmetric matrices as Hermitian positive semidefinite (PSD) matrices. To reduce the iteration complexity, the Riemannian gradient descent algorithm and the Riemannian trust-region algorithm are developed to support linear convergence rate and superlinear convergence rate, respectively. By exploiting the benign geometric structure of the blind demixing problem, i.e., symmetric rank-one matrices, the iteration cost can be significantly reduced (the same as the regularized gradient descent algorithm [17]) and is scalable to large-size problem.

In this paper, we prove that, for blind demixing, the Riemannian gradient descent algorithm with spectral initialization can linearly converge to the ground truth signals with high probability provided sufficient measurements. Numerical experiments will demonstrate the algorithmic advantages, performance gains and sample efficiency of the Riemannian optimization algorithms.

A. Related Works

1) Convex Optimization Approach: To address the algorithmic challenge of the rank-constraint optimization problem, the work [12] investigated the nuclear norm minimization method for the blind demixing problem. Although the nuclear norm based approach can solve this problem in polynomial time, the high iteration complexity yielded by the resulting semidefinite program (SDP) limits the scalability of the convex relaxation approach. This motivates the development of non-convex algorithms in order to reduce the iteration cost and simultaneously maintain competitive theoretical recovery guarantees compared with convex methods.

2) Non-Convex Optimization Paradigms: The recent work [17] proposed a non-convex regularized gradient-descent based method with a elegant initialization to solve the blind demixing problem at a linear convergence rate. Another work [18] implemented thresholding-based methods to solve the demixing problem for general rank-r matrices. The algorithm in [18] linearly converges to the global minimum with a similar initial strategy in [17]. Even though the iteration cost of the non-convex algorithm is lower than the convex approach, the overall computational complexity is still high due to the slow convergence rate, i.e., high iteration complexity. Moreover, both non-convex algorithms require careful initialization to achieve desirable performance.

To address the above limitations of the existing algorithms, we develop the Riemannian optimization algorithms to simultaneously reduce the iteration cost and iteration complexity. However, most of current developed Riemannian optimization algorithms for low-rank optimization problem [10], [20] are developed in real space with respect to a single optimization variable. Recently, a Riemannian steepest descent method is developed to solve the blind deconvolution problem [21], where the regularization is needed to provide statistical guarantees. In the blind demixing problem, the following coupled challenges arise due to *multiple complex asymmetric* variables:

- Constructing product Riemannian manifold for the multiple complex asymmetric rank-one matrices.
- Developing the Riemannian optimization algorithm on the complex product manifold.

Therefore, it is crucial to address these unique challenges to solve the blind demixing problem via the Riemannian optimization technique.

B. Contributions

The major contributions of the paper are summarized as follows:

1) We present a novel blind demixing approach to support low-latency communication in an orthogonal frequency division multiplexing (OFDM) system, thereby recovering the information signals without channel estimation. A low-rank approach is further developed to solve the blind demixing problem.

2) To efficiently exploit the quotient manifold geometry of the product of complex symmetric rank-one matrices, we equivalently reformulate the original complex asymmetric matrices to Hermitian positive semidefinite matrices.

3) To simultaneously reduce the iteration cost and iteration complexity as well as enhance the estimation performance, we develop the scalable Riemannian gradient descent algorithm and Riemannian trust-region algorithm by exploiting the benign geometric structure of symmetric rank-one matrices. This is achieved by factorizing the symmetric rank-one matrices.

4) We prove that, for blind demixing, the Riemannian gradient descent linearly converges to the ground truth signals with high probability provided sufficient measurements.

C. Organization and Notations

The remainder of this paper is organized as follows. The system model and problem formulations are presented in Section II. In Section III, we introduce the versatile framework of Riemannian optimization on the product manifold. The process of computing optimization related ingredients and the Riemannian optimization algorithms are explicated in Section IV. The theoretical guarantees of the Riemannian gradient descent algorithm is then presented in Section V. Numerical results will be illustrated in Section VI. We further conclude this paper in Section VII.

Throughout the paper following notions are used. Vectors and matrices are denoted by bold lowercase and bold uppercase letters respectively. Specifically, we let $\{X_k\}_{k=1}^s$ be the target matrices and $\{X_k^{[t]}\}_{k=1}^s$ be the t-th iterate of the algorithms where s is the number of users/devices. For a vector z, $\|z\|$ denotes its Euclidean norm. For a matrix M, $\|M\|_*$ and $\|M\|_F$ denote its nuclear norm and Frobenius norm respectively. For both matrices and vectors, M^{H} and z^{H} denote their complex conjugate transpose. $\bar{z}$ is the complex conjugate of the vector z. a^* is the complex conjugate of the complex constant a. The inner product of two complex matrices M_1 and M_2 is defined as $\langle M_1, M_2 \rangle = \mathrm{Tr}(M_1^{\mathsf{H}} M_2)$. Let I_L denote the identity matrix with size of $L \times L$.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present a blind demixing approach to support the low-latency communication by reducing the channel signaling overhead in orthogonal frequency division multiplexing (OFDM) system. We develop a low-rank optimization model to recover the original information signal for the blind demixing problem, which, however, turns out to be highly intractable. The Riemannian optimization approach is then motivated to address the computational issue.

A. Orthogonal Frequency Division Multiplexing

We first briefly introduce the orthogonal frequency division multiplexing (OFDM). The basic idea of OFDM is that by exploiting the eigenfunction property of sinusoids in linear-time-variant (LTI) system, transformation into the frequency domain is a particularly benign way to communicate over frequency-selective channels [22].

Let $p[n]$ and $\theta[n]$ denote the transmitted signal and the received signal in the n-th time slot, respectively. q_ℓ denotes the ℓ-th tap channel impulse response which does not change with n. Thus the channel is linear time-invariant. Here, the discrete-time model is given as

$$\theta[n] = \sum_{\ell=0}^{L'-1} q_\ell p[n-\ell], \tag{1}$$

where L' is a finite number of non-zero taps. However, the sinusoids are not eigenfunctions when we transmit data symbols $p = [p[0], p[1], \cdots, p[N_p-1]]^{\top} \in \mathbb{C}^{N_p}$ over only a finite duration [22]. To restore the eigenfunction property, we add a *cyclic prefix* to p. Specifically, we add a prefix of
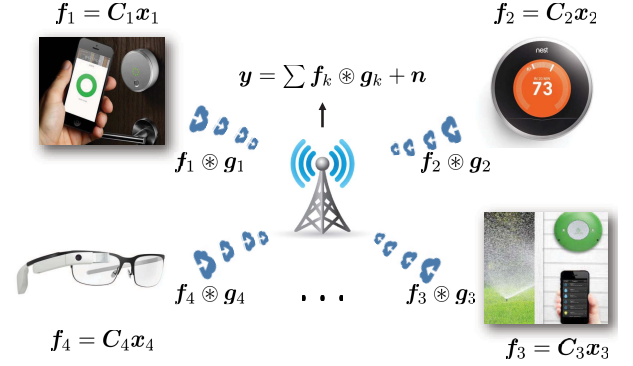

Fig. 1. Blind demixing for multi-user low-latency communication systems without channel estimation.

length $L'-1$ consisting of data symbols rotated cyclically:

$$d = [p[N_p - L' + 1], \cdots, p[N_p - 1], p[0], p[1], \cdots,$$
$$p[N_p - 1]]^{\top} \in \mathbb{C}^{N_p + L' - 1}. \tag{2}$$

We only consider the output over the time interval $n \in [L', N_p + L' - 1]$, represented as

$$\theta[n] = \sum_{\ell=0}^{L'-1} q_\ell d[(n - L' - \ell) \text{ modulo } N_p]. \tag{3}$$

Denoting the output of length N_p as

$$\theta = [\theta[L'], \cdots, \theta[N_p + L' - 1]]^{\top}, \tag{4}$$

and the channel impulse as $q = [q_0, q_1, \cdots, q_{L'-1}, 0, \cdots, 0]^{\top} \in \mathbb{C}^{N_p}$, thus (3) can be rewritten as $\theta = q \circledast p$, where the notion $\circledast$ denotes the *cyclic convolution*. This method is called *cyclic extension*.

B. System Model

In orthogonal frequency division multiplexing, we consider a network with one base station and s mobile users, as shown in Fig. 1. Specifically, let $x_k \in \mathbb{C}^N$ be the original signals of length N from the k-th user, which is usually assumed to be drawn from Gaussian distribution for the convenience of theoretical analysis [12], [17]. To make the model practical, in contrast, we consider x_k as an OFDM signal [23, Sec. 2.1], which consists of N orthogonal subcarriers modulated by N parallel data streams, represented as

$$x = F^{\mathsf{H}} s, \tag{5}$$

where $F \in \mathbb{C}^{N \times N}$ is the normalized discrete Fourier transform (DFT) matrix and $s \in \mathbb{C}^N$ is taken from QAM symbol constellation. Over L time slots, the transmit signal at the k-th user is given by $f_k = C_k x_k$, where $C_k \in \mathbb{C}^{L \times N}$ with $L > N$ is the encoding matrix and available to the base station. The signals f_k's are passing through individual time-invariant channels with impulse responses h_k's where $h_k \in \mathbb{C}^K$ has a maximum delay of at most K samples. In the OFDM system, this model can be represented in terms of circular convolutions of transmit signals $f_k \in \mathbb{C}^L$ with zero-padded channel vector $g_k \in \mathbb{C}^L$, i.e.,

$$g_k = [h_k^{\top}, 0, \cdots, 0]^{\top}. \tag{6}$$

Hence, the received signal is given as

$$z = \sum_{k=1}^{s} \boldsymbol{f}_k \circledast \boldsymbol{g}_k + \boldsymbol{n}, \qquad (7)$$

where $\boldsymbol{n}$ denotes the additive white complex Gaussian noise. Our goal is to recover the original information signals $\{\boldsymbol{x}_k\}_{k=1}^{s}$ from the single observation $\boldsymbol{z}$ without knowing channel impulse response $\{\boldsymbol{g}_k\}_{k=1}^{s}$. We call this problem as the *blind demixing* problem.

However, the above information recovery problem is highly intractable without any further structural assumptions. Fortunately, in wireless communication, we can design the encoding matrices $\{\boldsymbol{C}_k\}_{k=1}^{s}$ such that it satisfies "local" mutual incoherence conditions [12]. Specifically, from the practical points of view, we design the encoding matrix $\boldsymbol{C}_k$ as a Hadamard-type matrix [17], represented as

$$\boldsymbol{C}_k = \boldsymbol{F} \boldsymbol{D}_k \boldsymbol{H}, \qquad (8)$$

where $\boldsymbol{F} \in \mathbb{C}^{L \times L}$ is the normalized discrete Fourier transform (DFT) matrix, $\boldsymbol{D}_k$'s are independent diagonal binary ± 1 matrices and $\boldsymbol{H} \in \mathbb{C}^{L \times N}$ fixed partial deterministic Hadamard matrix. Furthermore, due to the physical properties of channel propagation [24], the impulse response $\boldsymbol{g}_k$ is compactly supported [18]. Here, the size of the compact set of $\boldsymbol{g}_k$, i.e., K where $K < L$, is termed as its maximum delay spread from an engineering perspective [18]. In this paper, we assume that the impulse response $\boldsymbol{g}_k$ is not available to both receivers and transmitters during the transmissions in order to reduce the channel signaling overhead [17].

C. Demixing of Rank-One Matrices

Let $\boldsymbol{B} \in \mathbb{C}^{L \times K}$ consist of the first K columns of $\boldsymbol{F}$. Due to the benign property of cyclic extension, it is convenient to represent the formulation (7) in the Fourier domain [12], [22]

$$\boldsymbol{y} = \boldsymbol{F} \boldsymbol{z} = \sum_{k} (\boldsymbol{F} \boldsymbol{C}_k \boldsymbol{x}_k) \odot \boldsymbol{B} \boldsymbol{h}_k + \boldsymbol{F} \boldsymbol{n}, \qquad (9)$$

where $\odot$ denotes the componentwise product. The first term of (9) can be further rewritten as [18] $[(\boldsymbol{F} \boldsymbol{C}_k \boldsymbol{x}_k) \odot \boldsymbol{B} \boldsymbol{h}_k]_i = (\boldsymbol{c}_{ki}^{\mathsf{H}} \boldsymbol{x}_k)(\boldsymbol{b}_i^{\mathsf{H}} \boldsymbol{h}_k) = \langle \boldsymbol{c}_{ki} \bar{\boldsymbol{b}}_i^{\mathsf{H}}, \boldsymbol{X}_k \rangle$, where $\boldsymbol{c}_{ki}^{\mathsf{H}}$ denotes the i-th row of $\boldsymbol{F} \boldsymbol{C}_k$, $\boldsymbol{b}_i^{\mathsf{H}}$ represents the i-th row of $\boldsymbol{B}$ and $\boldsymbol{X}_k = \boldsymbol{x}_k \bar{\boldsymbol{h}}_k^{\mathsf{H}} \in \mathbb{C}^{N \times K}$ is a rank-one matrix. Hence, the received signal at the base station can be represented in the Fourier domain as

$$\boldsymbol{y} = \sum_{k=1}^{s} \mathcal{A}_k(\boldsymbol{X}_k) + \boldsymbol{e}, \qquad (10)$$

where the vector $\boldsymbol{e} = \boldsymbol{F} \boldsymbol{n}$ and the linear operator $\mathcal{A}_k : \mathbb{C}^{N \times K} \to \mathbb{C}^{L}$ is given as [17]

$$\mathcal{A}_k(\boldsymbol{X}_k) := \{ \langle \boldsymbol{c}_{ki} \bar{\boldsymbol{b}}_i^{\mathsf{H}}, \boldsymbol{X}_k \rangle \}_{i=1}^{L} = \{ \langle \boldsymbol{A}_{ki}, \boldsymbol{X}_k \rangle \}_{i=1}^{L}, \quad (11)$$

with $\boldsymbol{A}_{ki} = \boldsymbol{c}_{ki} \bar{\boldsymbol{b}}_i^{\mathsf{H}}$. We thus formulate the blind demixing problem as the following low-rank optimization problem:

$$\mathscr{P} : \underset{\boldsymbol{W}_k, k=1, \cdots, s}{\text{minimize}} \left\| \sum_{k=1}^{s} \mathcal{A}_k(\boldsymbol{W}_k) - \boldsymbol{y} \right\|^2$$
$$\text{subject to } \text{rank}(\boldsymbol{W}_k) = 1, \ k = 1, \cdots, s, \quad (12)$$

where $\boldsymbol{W}_k \in \mathbb{C}^{N \times K}, k = 1, \cdots, s$ and $\{\mathcal{A}_k\}_{k=1}^{s}$ are known. However, problem $\mathscr{P}$ turns out to be highly intractable due

to non-convexity of rank-one constraints. Despite the non-convexity of problem (12), we solve this problem via exploiting the quotient manifold geometry of the product of complex symmetric rank-one matrices. This is achieved by equivalently reformulating the original complex asymmetric matrices as the Hermitian positive semidefinite (PSD) matrices. The first-order and second-order algorithms are further developed on the quotient manifold, which enjoy algorithmic advantages, performance gains and sample efficiency.

D. Problem Analysis

To address the algorithmic challenge of problem $\mathscr{P}$, enormous progress has been recently made to develop convex methods [12], [16] and non-convex methods [17], [18]. In this subsection, we will first review the existing algorithms for the blind demixing problem. Then we identify the limitations of state-of-the-art algorithms and develop Riemannian trust-region algorithm to address these limitations. Unique challenges of developing the Riemannian optimization algorithm will be further revealed.

1) Convex Relaxation Approach: A line of literature [12] adopted the nuclear norm minimization method to reformulate the problem $\mathscr{P}$ as

$$\underset{\boldsymbol{W}_k, k=1, \cdots, s}{\text{minimize}} \ \sum_{k=1}^{s} \|\boldsymbol{W}_k\|_*$$
$$\text{subject to } \left\| \sum_{k=1}^{s} \mathcal{A}_k(\boldsymbol{W}_k) - \boldsymbol{y} \right\| \le \varepsilon, \quad (13)$$

where $\boldsymbol{W}_k \in \mathbb{C}^{N \times K}$ and the parameter ε is an upper bound of $\|\boldsymbol{e}\|$ in (10) and assumed to be known. While the blind demixing problem can be solved by convex technique provably and robustly under certain situations, the convex relaxation approach is computationally infeasible to the medium-scale or large-scale problems due to the limitations of high iteration cost. This motivates the development of efficient non-convex approaches with lower iteration cost.

2) Non-Convex Optimization Paradigms: A line of recent work [17], [18] has developed non-convex algorithms which reduces the iteration cost. In particular, work [18] solved problem $\mathscr{P}$ via the hard thresholding technique. Specifically, the t-th iterate with respect to the k-th variable is given by $\boldsymbol{W}_k^{[t+1]} = \mathcal{F}_r(\boldsymbol{W}_k^{[t]} + \alpha_k^{[t]} \mathcal{P}_{T_{k,t}}(\boldsymbol{G}_k^{[t]}))$, where the hard thresholding operator $\mathcal{F}_r$ returns the best rank-r approximation of a matrix, $\mathcal{P}_{T_{k,t}}(\boldsymbol{G}_k^{[t]})$ represents the projection of the search direction to the tangent space $T_{k,t}$, and the stepsize is denoted as $\alpha_k^{[t]} = \|\mathcal{P}_{T_{k,t}}(\boldsymbol{G}_k^{[t]})\|_F^2 / \|\mathcal{A}_k \mathcal{P}_{T_{k,t}}(\boldsymbol{G}_k^{[t]})\|_2^2$. Therein, $\boldsymbol{G}_k^{[t]}$ is defined in [18], given by $\boldsymbol{G}_k^{[t]} = \mathcal{A}_k^*(\boldsymbol{r}^{[t]})$, where $\boldsymbol{r}^{[t]} = \boldsymbol{y} - \sum_{k=1}^{s} \mathcal{A}_k(\boldsymbol{W}_k^{[t]}), \mathcal{A}_k^*(\boldsymbol{z}) = \sum_{l=1}^{L} z_l \boldsymbol{b}_i \boldsymbol{c}_{ki}^{\mathsf{H}}$.

Moreover, matrix factorization also serves as a powerful method to address the low-rank optimization problem. Specifically, Ling and Strohmer [17] developed an algorithm solving the blind demixing problem based on matrix factorization and regularized gradient descent method. Specifically, problem $\mathscr{P}$ can be rewritten as

$$\underset{\boldsymbol{u}_k, \boldsymbol{v}_k, k=1, \cdots, s}{\text{minimize}} \ F(\boldsymbol{u}, \boldsymbol{v}) := g(\boldsymbol{u}, \boldsymbol{v}) + \lambda R(\boldsymbol{u}, \boldsymbol{v}), \quad (14)$$

where $g(\boldsymbol{u}, \boldsymbol{v}) := \| \sum_{k=1}^{s} \mathcal{A}_k(\boldsymbol{u}_k \boldsymbol{v}_k^{\mathsf{H}}) - \boldsymbol{y} \|^2$ with $\boldsymbol{u}_k \in \mathbb{C}^N, \boldsymbol{v}_k \in \mathbb{C}^K$ and the regularizer $R(\boldsymbol{u}, \boldsymbol{v})$ is proposed to

force the iterates to lie in the *basin of attraction* [17]. The algorithm starts from a good initial point and updates the iterates simultaneously:

$$\boldsymbol{u}_k^{[t+1]} = \boldsymbol{u}_k^{[t]} - \eta \nabla F_{\boldsymbol{u}_k}(\boldsymbol{u}_k^{[t]}, \boldsymbol{v}_k^{[t]}), \tag{15}$$

$$\boldsymbol{v}_k^{[t+1]} = \boldsymbol{v}_k^{[t]} - \eta \nabla F_{\boldsymbol{v}_k}(\boldsymbol{u}_k^{[t]}, \boldsymbol{v}_k^{[t]}), \tag{16}$$

where $\nabla F_{\boldsymbol{u}_k}$ denotes the derivative of the objective function (14) with respect to $\boldsymbol{u}_k$. Although the above non-convex algorithms have low iteration cost, the overall computational complexity is still high due to the slow convergence rate, i.e., high iteration complexity. This motivates to design efficient algorithms to simultaneously reduce the iteration cost and iteration complexity.

E. Riemannian Optimization Approach

In this paper, we develop Riemannian optimization algorithms to solve problem $\mathscr{P}$, thereby addressing the limitations of the existing algorithms (e.g., regularized gradient descent algorithm [17], nuclear norm minimization method [14] and fast iterative hard thresholding algorithm [18]) by

- Exploiting the Riemannian quotient geometry of the product of *complex asymmetric rank-one* matrices to reduce the iteration cost.
- Developing scalable Riemannian gradient descent algorithm and Riemannian trust-region algorithm to reduce the iteration complexity.

The Riemannian optimization technique has been applied in a wide range of areas to solve rank-constrained problem and achieves excellent performance, e.g., the low-rank matrix completion problem [20], [25], [26], topological interference management problem [10] and blind deconvolution [21]. However, all of the current Riemannian optimization problems for low-rank optimization problem are developed on the real Riemannian manifold (e.g., real Grassmann manifold [20] and quotient manifold of fixed-rank matrices [10], [25], [26]) with single non-symmetric variable or the complex Riemannian manifold with single symmetric variable [27], [28] or the complex Riemannian manifold with single non-symmetric variable [21]. Thus unique challenges arise due to *multiple complex asymmetric variables* in problem $\mathscr{P}$. In this paper, we propose to construct product Riemannian manifold for the multiple complex asymmetric rank-one matrices, followed by developing the scalable Riemannian optimization algorithms.

III. RIEMANNIAN OPTIMIZATION OVER PRODUCT MANIFOLDS

In this section, to exploit the Riemannian quotient geometry of the product of complex symmetric rank-one matrices [27], we reformulate the original optimization problem on complex asymmetric matrices to the one on Hermitian positive semidefinite (PSD) matrices. The Riemannian optimization algorithms are further developed via exploiting the quotient manifold geometry of the complex product manifold.

A. Product Riemannian Manifold

Let the manifold $\mathcal{M}$ denote the Riemannian manifold endowed with the Riemannian metric $g_{\boldsymbol{M}_k}$ where $k \in [s]$ with

$[n] = \{1, 2, \cdots, n\}$. The set $\mathcal{M}^s = \underbrace{\mathcal{M} \times \mathcal{M} \times \cdots \times \mathcal{M}}_{s}$ is defined as the set of matrices $(\boldsymbol{M}_1, \cdots, \boldsymbol{M}_s)$ where $\boldsymbol{M}_k \in \mathcal{M}, k = 1, 2 \cdots, s$, and is called product manifold. Its manifold topology is identified to the product topology [29].

In order to develop the optimization algorithms on the manifold, a notion of length that applies to tangent vectors is needed [29]. By taking the manifold $\mathcal{M}$ as an example, this goal is achieved by endowing each tangent space $T_{\boldsymbol{M}}\mathcal{M}$ with a smoothly varying inner product $g_{\boldsymbol{M}}(\boldsymbol{\zeta}_{\boldsymbol{M}}, \boldsymbol{\eta}_{\boldsymbol{M}})$ where $\boldsymbol{\zeta}_{\boldsymbol{M}}, \boldsymbol{\eta}_{\boldsymbol{M}} \in T_{\boldsymbol{M}}\mathcal{M}$. Endowed with a inner product $g_{\boldsymbol{M}}$, the manifold $\mathcal{M}$ is called the *Riemannian manifold*. The smoothly varying inner product is called the *Riemannian metric*. The above discussions are also applied to the product manifold. Based on the above discussions, we characterize the notion of length on the product manifold via endowing tangent space $T_{\boldsymbol{V}}\mathcal{M}^s$ with the smoothly varying inner product

$$g_{\boldsymbol{V}}(\boldsymbol{\zeta}_{\boldsymbol{V}}, \boldsymbol{\eta}_{\boldsymbol{V}}) := \sum_{k=1}^{s} g_{\boldsymbol{M}_k}(\boldsymbol{\zeta}_{\boldsymbol{M}_k}, \boldsymbol{\eta}_{\boldsymbol{M}_k}). \tag{17}$$

Thus, with $\mathcal{M}$ as the Riemannian manifold, the product manifold $\mathcal{M}^s$ is also called Riemannian manifold, endowed with the Riemannian metric $g_{\boldsymbol{V}}$.

B. Handling Complex Asymmetric Matrices

To handle complex asymmetric matrices, we propose a linear map which is exploited to convert the optimization variables to the Hermitian positive semidefinite matrix. Let $\mathbb{S}_+^{(N+K)}$ denote the set of Hermitian positive semidefinite matrices. Define the linear map $\mathcal{J}_k : \mathbb{S}_+^{(N+K)} \rightarrow \mathbb{C}^L$ and a Hermitian positive semidefinite (PSD) matrix $\boldsymbol{Y}_k$ such that $[\mathcal{J}_k(\boldsymbol{Y}_k)]_i = \langle \boldsymbol{J}_{ki}, \boldsymbol{Y}_k \rangle$ with $\boldsymbol{Y}_k \in \mathbb{S}_+^{(N+K)}$ and $\boldsymbol{J}_{ki}$ as

$$\boldsymbol{J}_{ki} = \begin{bmatrix} \boldsymbol{0}_{N \times N} & \boldsymbol{A}_{ki} \\ \boldsymbol{0}_{K \times N} & \boldsymbol{0}_{K \times K} \end{bmatrix} \in \mathbb{C}^{(N+K) \times (N+K)}, \tag{18}$$

where $\boldsymbol{A}_{ki}$ is given in (11). Note that

$$[\mathcal{J}_k(\boldsymbol{M}_k)]_i = \langle \boldsymbol{J}_{ki}, \boldsymbol{M}_k \rangle = \langle \boldsymbol{A}_{ki}, \boldsymbol{x}_k \overline{\boldsymbol{h}}_k^{\mathsf{H}} \rangle \tag{19}$$

where $\boldsymbol{M}_k = \boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}}$ with $\boldsymbol{w}_k = [\boldsymbol{x}_k^{\mathsf{H}} \ \overline{\boldsymbol{h}}_k^{\mathsf{H}}]^{\mathsf{H}} \in \mathbb{C}^{N+K}$. Hence, problem $\mathscr{P}$ can be equivalently reformulated as the following optimization problem on the set of Hermitian positive semidefinite matrices

$$\underset{\boldsymbol{M}_k, k=1, \cdots, s}{\text{minimize}} \ \left\| \sum_{k=1}^{s} \mathcal{J}_k(\boldsymbol{M}_k) - \boldsymbol{y} \right\|^2$$
$$\text{subject to} \ \text{rank}(\boldsymbol{M}_k) = 1, \quad k = 1, \cdots, s, \tag{20}$$

where $\boldsymbol{M}_k \in \mathbb{S}_+^{(N+K)}$. Based on equality (19), we know that the top-right $N \times K$ submatrix of the estimated matrix $\hat{\boldsymbol{M}}_k$ in problem (20) is corresponding to the estimated matrix $\hat{\boldsymbol{W}}_k$ in problem (12). Furthermore, we define $\boldsymbol{V} = \{\boldsymbol{M}_k\}_{k=1}^s \in \mathcal{M}^s$, where $\mathcal{M}$ denotes the manifold encoded with the rank-one matrices and $\mathcal{M}^s$ represents the product of s manifolds $\mathcal{M}$. By exploiting the Riemannian manifold geometry, the rank-constrained optimization problem (20) can be transformed into the following unconstrained optimization problem over the search space of the product manifold $\mathcal{M}^s$:

$$\underset{\boldsymbol{V} = \{\boldsymbol{M}_k\}_{k=1}^s}{\text{minimize}} \ f(\boldsymbol{V}) := \left\| \sum_{k=1}^{s} \mathcal{J}_k(\boldsymbol{M}_k) - \boldsymbol{y} \right\|^2. \tag{21}$$

Note that the theoretical advantages of the symmetric transformation have been recently revealed in [30] for the low-rank optimization problems in machine learning and high-dimensional statistics. Furthermore, by factorizing the Hermitian positive semidefinite matrix $M_k = w_k w_k^{\mathsf{H}}$, $k = 1, \cdots, s$, it yields only s vector variables, i.e., $w_k \in \mathbb{C}^{N+K}$, for optimization problem (21), which simplifies the derivation of optimization related ingredients.

C. Quotient Manifold Space

The main idea of Riemannian optimization for rank-constrained problem is based on matrix factorization [26], [27]. In particular, the factorization $M_k = w_k w_k^{\mathsf{H}}$ where $w_k \in \mathbb{C}^{N+K}$ is prevalent in dealing with rank-one Hermitian positive semidefinite matrices [27], [28]. This factorization also takes advantages of lower-dimensional search space [20] over the other general forms of matrix factorization for rank-one matrices [26]. However, the factorization $M_k = w_k w_k^{\mathsf{H}}$ is not unique because the transformation $w_k \mapsto a_k w_k$ leaves the matrix $w_k w_k^{\mathsf{H}}$ unchanged, where $a_k \in \mathrm{SU}(1) := \{a_k \in \mathbb{C} : a_k a_k^* = a_k^* a_k = 1\}$ and $\mathrm{SU}(1)$ is the special unitary group of degree 1. In particular, the non-uniqueness yields a profound affect on the performance of second-order optimization algorithms which require non-degenerate critical points. To address this indeterminacy, we encode the transformation $w_k \mapsto a_k w_k$ where $k = 1, 2, \cdots, s$, in an abstract search space to construct the equivalence class:

$$[M_k] = \{a_k w_k : a_k a_k^* = a_k^* a_k = 1, a_k \in \mathbb{C}\}. \qquad (22)$$

The product of $[M_k]$'s yields the equivalence class

$$[V] = \{[M_k]\}_{k=1}^s. \qquad (23)$$

The set of equivalence classes (23) is denoted as $\mathcal{M}^s/\sim$, called the *quotient space* [27]. Since the quotient manifold $\mathcal{M}^s/\sim$ is an abstract space, in order to execute the optimization algorithm, the matrix representations defined in the computational space are needed to represent corresponding abstract geometric objects in the abstract space [29]. We denote an element of the quotient space $\mathcal{M}^s/\sim$ by $\tilde{V}$ and its matrix representation in the computational space $\mathcal{M}^s$ by V. Therefore, there is $\tilde{V} = \pi(V)$ and $[V] = \pi^{-1}(\pi(V))$, where the mapping $\pi : \mathcal{M}^s \to \mathcal{M}^s/\sim$ is called the natural projection constructing the map between the geometric ingredients of the computational space and the ones of the quotient space.

D. The Framework of Riemannian Optimization

To develop the Riemannian optimization algorithms over the quotient space, the geometric concepts in the abstract space $\mathcal{M}^s/\sim$ call for the matrix representations in the computational space $\mathcal{M}^s$. Specifically, several classical geometric concepts in the Euclidean space are required to be generalized to the geometric concepts on the manifold, such as the notion of length to the Riemannian metric, set of directional derivatives to the tangent space and motion along geodesics to the retraction. We first present the general Riemannian optimization developed on the product manifold in this subsection.

The details on the derivation of concrete optimization-related ingredients will be introduced in the next section.

Based on the Riemannian metric (17), the tangent space $T_V \mathcal{M}^s$ can be decomposed into two complementary vector spaces, given as [29]

$$T_V \mathcal{M}^s = \mathcal{V}_V \mathcal{M}^s \oplus \mathcal{H}_V \mathcal{M}^s, \qquad (24)$$

where $\oplus$ is the direct sum operator. Here, $\mathcal{V}_V \mathcal{M}^s$ is the *vertical space* where directions of vectors are tangent to the set of equivalence class (23). $\mathcal{H}_V \mathcal{M}^s$ is the *horizontal space* where the directions of vectors are orthogonal to the equivalence class (23). Thus the tangent space $T_{\tilde{V}}(\mathcal{M}^s/\sim)$ at the point $\tilde{V} \in \mathcal{M}^s/\sim$ can be represented by the horizontal space $\mathcal{H}_V \mathcal{M}^s$ at point $V \in \mathcal{M}^s$. In particular, the matrix representation of $\eta_{\tilde{V}} \in T_{\tilde{V}}(\mathcal{M}^s/\sim)$, which is also called the *horizontal lift* of $\eta_{\tilde{V}} \in T_{\tilde{V}}(\mathcal{M}^s/\sim)$ [29, Sec. 3.5.8], can be represented by a unique element $\eta_V \in \mathcal{H}_V \mathcal{M}^s$. In addition, for every $\xi_V, \eta_V \in T_V \mathcal{M}^s$, based on the Riemannian metric $g_V(\zeta_V, \eta_V)$ (17),

$$g_{\tilde{V}}(\zeta_{\tilde{V}}, \eta_{\tilde{V}}) := g_V(\zeta_V, \eta_V) \qquad (25)$$

defines a Rimannian metric on the quotient space $\mathcal{M}^s/\sim$ [29, Sec. 3.6.2], where $\zeta_{\tilde{V}}, \eta_{\tilde{V}} \in T_{\tilde{V}} \mathcal{M}^s$. Endowed with the Riemannian metric (25), the natural projection map $\pi : \mathcal{M}^s \to \mathcal{M}^s/\sim$ is a *Riemannian submersion* from the quotient manifold $\mathcal{M}^s/\sim$ to the computational space $\mathcal{M}^s$ [29, Sec. 3.6.2]. Based on the Riemannian submersion theory, several objects on the quotient manifold can be represented by corresponding objects in the computational space.

Based on the aforementioned framework, the Riemannian optimization over the computational space $\mathcal{M}^s$ can be briefly described as follows. First, search the directions η_V on the horizontal space $\mathcal{H}_V \mathcal{M}^s$. Then, retract the directional vector η_V onto the manifold $\mathcal{M}^s$ via the mapping $\mathcal{R}_V : \mathcal{H}_V \mathcal{M}^s \to \mathcal{M}^s$ called *retraction*. Since the manifold topology of the product manifold is equivalent to the product topology [29, Sec. 3.1.6], i.e., the aforementioned optimization operations can be handled individually, the Riemannian optimization developed on product manifold $\mathcal{M}^s$ can be individually processed on the manifold $\mathcal{M}$.

Specifically, the tangent space $T_V \mathcal{M}^s$ can be termed as the product of the tangent spaces $T_{M_k} \mathcal{M}$ for $k = 1, 2, \cdots, s$. In the context of individual Riemannian manifolds $\mathcal{M}$, the tangent space $T_{M_k} \mathcal{M}$ can be decomposed into two complementary vector space in the sense of the Riemannian metric g_{M_k}, given as $T_{M_k} \mathcal{M} = \mathcal{V}_{M_k} \mathcal{M} \oplus \mathcal{H}_{M_k} \mathcal{M}$, where vectors on the horizontal space $\mathcal{H}_{M_k} \mathcal{M}$ are orthogonal to the equivalence class (22). For $k = 1, 2, \cdots, s$, we individually the search direction η_{M_k} on the horizontal space $\mathcal{H}_{M_k} \mathcal{M}$ and individually retract the directions to the manifold $\mathcal{M}$ via the retraction mapping $\mathcal{R}_{M_k}$. To sum up, the schematic viewpoint of a Riemannian optimization algorithm developed on the product manifold via element-wise extension is illustrated in Fig. 2.

IV. Optimization Related Ingredients and Algorithms

In this section, we provide the optimization related ingredients on product manifold $\mathcal{M}^s$ via the elementwise extension

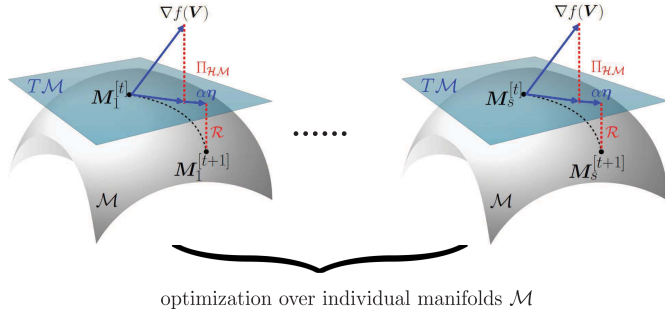optimization over individual manifolds $\mathcal{M}$

Fig. 2. Graphical representation of the concept of Riemannian optimization over individual Riemannian manifolds.

of the optimization related ingredients on the individual Riemannian manifold $\mathcal{M}$. The optimization problem is reformulated via matrix factorization $\boldsymbol{M}_k = \boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}}$, given as

$$\operatorname*{minimize}_{\boldsymbol{v}=\{\boldsymbol{w}_k\}_{k=1}^s} f(\boldsymbol{v}) := \left\| \sum_{k=1}^s \mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}}) - \boldsymbol{y} \right\|^2, \quad (26)$$

where $\boldsymbol{w}_k \in \mathbb{C}^{N+K}$, with $k = 1, \cdots, s$. Therefore, according to (19), the estimated information signal $\hat{\boldsymbol{x}}_k$ is represented by the first N rows of the estimated $\hat{\boldsymbol{w}}_k$. Compared with problem (14) of $2s$ vector variables, i.e., $\{\boldsymbol{u}_k, \boldsymbol{v}_k\}_{k=1}^s$, problem (26) endows with only s vector variables, i.e., $\{\boldsymbol{w}_k\}_{k=1}^s$, by taking the advantages of the symmetric transformation. To simplify the notations, we abuse the notions of $\boldsymbol{v} \in \mathcal{M}^s$, $\boldsymbol{w}_k \in \mathcal{M}$. For convenience, we only take the k-th variables as an example and express all the optimization related ingredients with respect to the complex vector $\boldsymbol{w}_k \in \mathcal{M} := \mathbb{C}_*^{N+K}$, where the space $\mathbb{C}_*^n$ denotes the complex Euclidean space $\mathbb{C}^n$ removing the origin.

A. Optimization Related Ingredients

In order to develop the Riemannian optimization algorithm, we need to derive the matrix representations of Riemannian gradient and Riemannian Hessian. To achieve this goal, corresponding optimization related ingredients are required, which are introduced in the following.

The Riemannian metric $g_{\boldsymbol{w}_k} : T_{\boldsymbol{w}_k}\mathcal{M} \times T_{\boldsymbol{w}_k}\mathcal{M} \to \mathbb{R}$ is an inner product between the tangent vectors on the tangent space $T_{\boldsymbol{w}_k}\mathcal{M}$ and invariable along the set of equivalence (22), which is given by [27]

$$g_{\boldsymbol{w}_k}(\boldsymbol{\zeta}_{\boldsymbol{w}_k}, \boldsymbol{\eta}_{\boldsymbol{w}_k}) = \operatorname{Tr}(\boldsymbol{\zeta}_{\boldsymbol{w}_k}^{\mathsf{H}}\boldsymbol{\eta}_{\boldsymbol{w}_k} + \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}}\boldsymbol{\zeta}_{\boldsymbol{w}_k}), \quad (27)$$

where $\boldsymbol{\zeta}_{\boldsymbol{w}_k}, \boldsymbol{\eta}_{\boldsymbol{w}_k} \in T_{\boldsymbol{w}_k}\mathcal{M}$. Based on the Riemannian metric (27), we further introduce the horizontal space, i.e., the matrix representation of the tangent space $T_{\tilde{\boldsymbol{w}}_k}(\mathcal{M}/\sim)$, the horizontal space projection, which is vital to the derivations of Riemannian gradient and Riemannian Hessian.

Proposition 1 (Horizontal Space): The quotient manifold $\mathcal{M}/\sim$ admits a horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M} \triangleq \{\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathbb{C}^{N+K} : \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}}\boldsymbol{w}_k = \boldsymbol{w}_k^{\mathsf{H}}\boldsymbol{\eta}_{\boldsymbol{w}_k}\}$, which is the complementary subspace of $\mathcal{V}_{\boldsymbol{w}_k}\mathcal{M}$ with respect to the Riemannian metric (27), providing the matrix representation of the abstract tangent space $T_{\tilde{\boldsymbol{w}}_k}(\mathcal{M}/\sim)$.

Proof: The proof is mainly based on [29, Sec. 3.5]. ∎

Proposition 2 (Horizontal Space Projection): The operator $\Pi_{H_{\boldsymbol{w}_k}\mathcal{M}} : T_{\boldsymbol{w}_k}\mathcal{M} \to \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$ that projects vectors on the tangent space onto the horizontal space is called horizontal space projection. It is given by $\Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\boldsymbol{\eta}_{\boldsymbol{w}_k}) = \boldsymbol{\eta}_{\boldsymbol{w}_k} - a\boldsymbol{w}_k$, where $a \in \mathbb{C}$ is derived as $a = \frac{\boldsymbol{w}_k^{\mathsf{H}}\boldsymbol{\eta}_{\boldsymbol{w}_k} - \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}}\boldsymbol{w}_k}{2\boldsymbol{w}_k^{\mathsf{H}}\boldsymbol{w}_k}$.

Proof: The proof is mainly based on [29, Sec. 3.5]. ∎

1) Riemannian Gradient: Let $\tilde{f} := f \circ \pi$ be the projection of the function f on the quotient space $\mathcal{M}/\sim$. Consider a point $\boldsymbol{w}_k \in \mathcal{M}$ and corresponding point $\tilde{\boldsymbol{w}}_k \in \mathcal{M}/\sim$, the matrix representation (horizontal lift) of $\operatorname{grad}_{\tilde{\boldsymbol{w}}_k}\tilde{f} \in T_{\tilde{\boldsymbol{w}}_k}(\mathcal{M}/\sim)$，i.e., $\operatorname{grad}_{\boldsymbol{w}_k}f \in \mathcal{H}_{\boldsymbol{w}}\mathcal{M}$, is required to develop second-order algorithm on the computational space $\mathcal{M}$. Specifically, the Riemannian gradient $\operatorname{grad}_{\boldsymbol{w}_k}f$ can be induced from the Euclidean gradient of $f(\boldsymbol{v})$ with respect to $\boldsymbol{w}_k$. The relationship between them is given by [29, Sec. 3.6]

$$g_{\boldsymbol{w}_k}(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \operatorname{grad}_{\boldsymbol{w}_k}f) = \nabla_{\boldsymbol{w}_k}f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}], \quad (28)$$

where the directional vector is $\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$ and $\nabla_{\boldsymbol{w}_k}f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \lim_{t\to 0}\frac{f(\boldsymbol{v})|_{\boldsymbol{w}_k+t\boldsymbol{\eta}_k} - f(\boldsymbol{v})|_{\boldsymbol{w}_k}}{t}$. Thus the Riemannian gradient is given by

$$\operatorname{grad}_{\boldsymbol{w}_k}f = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\tfrac{1}{2}\nabla_{\boldsymbol{w}_k}f(\boldsymbol{v})), \quad (29)$$

where $\Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\cdot)$ is the horizontal space projector operator. In particular, the Euclidean gradient of $f(\boldsymbol{v})$ with respect to $\boldsymbol{w}_k$ is represented as

$$\nabla_{\boldsymbol{w}_k}f(\boldsymbol{v}) = 2 \cdot \sum_{i=1}^L (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k, \quad (30)$$

where $c_i = \sum_{k=1}^s [\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})]_i - y_i$. Details will be demonstrated in Appendix A.

2) Riemannian Hessian: In order to perform the second-order algorithm, the matrix representation (horizontal lift) of $\operatorname{Hess}_{\tilde{\boldsymbol{w}}_k}\tilde{f}[\tilde{\boldsymbol{\eta}}_{\boldsymbol{w}_k}] \in T_{\tilde{\boldsymbol{w}}_k}\mathcal{M}$, i.e., $\operatorname{Hess}_{\boldsymbol{w}_k}f[\boldsymbol{\eta}_{\boldsymbol{w}_k}] \in \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$, needs to be computed via projecting the directional derivative of the Riemannian gradient $\operatorname{grad}_{\boldsymbol{w}_k}f$ to the horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$. Based on Propositions 1 and 2, it yields that

$$\operatorname{Hess}_{\boldsymbol{w}_k}f[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(\nabla_{\boldsymbol{\eta}_{\boldsymbol{w}_k}}\operatorname{grad}_{\boldsymbol{w}_k}f), \quad (31)$$

where $\operatorname{grad}_{\boldsymbol{w}_k}f$ (28) denotes the Riemannian gradient and $\nabla_{\boldsymbol{\eta}_{\boldsymbol{w}_k}}\operatorname{grad}_{\boldsymbol{w}_k}f$ is the *Riemannian connection*. In particular, Riemannian connection $\nabla_{\boldsymbol{\eta}_{\boldsymbol{w}_k}}\boldsymbol{\xi}_{\boldsymbol{w}_k}$ characterizes the directional derivative of the Riemannian gradient and satisfies two properties (i.e., symmetry and invariance of the Riemannian metric) [29, Sec. 5.3]. Under the structure of the manifold $\mathcal{M}$, the Riemannian connection is given as

$$\nabla_{\boldsymbol{\eta}_{\boldsymbol{w}_k}}\boldsymbol{\xi}_{\boldsymbol{w}_k} = \Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}(D\boldsymbol{\xi}_{\boldsymbol{w}_k}[\boldsymbol{\eta}_{\boldsymbol{w}_k}]), \quad (32)$$

where $D\boldsymbol{\xi}_{\boldsymbol{w}_k}[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$ is the Euclidean directional derivative of $\boldsymbol{\xi}_{\boldsymbol{w}_k}$ in the direction of $\boldsymbol{\eta}_{\boldsymbol{w}_k}$.

To derive the Riemannian Hessian (31), we first compute the directional derivative of Euclidean gradient $\nabla_{\boldsymbol{w}_k}f(\boldsymbol{v})$ (30) in the direction of $\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$, given by

$$\nabla_{\boldsymbol{w}_k}^2 f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = 2\sum_{i=1}^L (b_i \boldsymbol{J}_{ki} + b_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k \\ + (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{\eta}_{\boldsymbol{w}_k}, \quad (33)$$

<div align="center">TABLE I</div>

<div align="center">ELEMENT-WISE OPTIMIZATION-RELATED INGREDIENTS FOR PROBLEM $\mathscr{P}$</div>

	minimize $_{w_k \in \mathcal{M}} \| \sum_{k=1}^{s} \mathcal{J}_k(w_k w_k^{\mathsf{H}}) - y\|^2$
Computational space $\mathcal{M}$	$\mathbb{C}_*^{N+K}$
Quotient space $\mathcal{M}/\sim$	$\mathbb{C}_*^{N+K}/\mathrm{SU}(1)$
Riemannian metric g_{w_k}	$g_{w_k}(\zeta_{w_k}, \eta_{w_k}) = \mathrm{Tr}(\zeta_{w_k}^{\mathsf{H}} \eta_{w_k} + \eta_{w_k}^{\mathsf{H}} \zeta_{w_k})$ (27)
Horizontal space $\mathcal{H}_{w_k}\mathcal{M}$	$\eta_{w_k} \in \mathbb{C}^{N+K} : \eta_{w_k}^{\mathsf{H}} w_k = w_k^{\mathsf{H}} \eta_{w_k}$
Horizontal space projection	$\Pi_{\mathcal{H}_{w_k}\mathcal{M}}(\eta_{w_k}) = \eta_{w_k} - a w_k, a \in \mathbb{C}$
Riemannian gradient $\mathrm{grad}_{w_k} f$	$\mathrm{grad}_w f = \frac{1}{2}\nabla_{w_k} f(v)$ (43)
Riemannian Hessian $\mathrm{Hess}_{w_k} f[\eta_{w_k}]$	$\mathrm{Hess}_{w_k} f[\eta_{w_k}] = \Pi_{\mathcal{H}_{w_k}\mathcal{M}}(\frac{1}{2}\nabla_{w_k}^2 f(v)[\eta_{w_k}])$ (34)
Retraction $\mathcal{R}_{w_k} : \mathcal{H}_{w_k}\mathcal{M} \to \mathcal{M}$	$\mathcal{R}_{w_k}(\eta_{w_k}) = w_k + \eta_{w_k}$ (35)

where $b_i = \sum_{k=1}^{s}\langle J_{ki}, \eta_{w_k} w_k^{\mathsf{H}} + w_k \eta_{w_k}^{\mathsf{H}}\rangle$. According to the formulations (29), (32) and (33), the Riemannian Hessian is given as

$$\mathrm{Hess}_{w_k} f[\eta_{w_k}] = \Pi_{\mathcal{H}_{w_k}\mathcal{M}}(\frac{1}{2}\nabla_{w_k}^2 f(v)[\eta_{w_k}]). \qquad (34)$$

Details will be illustrated in Appendix A. To sum up, the element-wise optimization-related ingredients with respect to the manifold $\mathcal{M}$ for the problem (26) are provided in Table I.

B. Riemannian Optimization Algorithms

Based on the optimization related ingredients mentioned in Section IV-A, we develop the Riemannian gradient algorithm and Riemannian trust-region algorithm, respectively.

1) Riemannian Gradient Descent Algorithm: In the Riemannian gradient descent algorithm, i.e., Algorithm 1, the search direction is given by $\eta = -\mathrm{grad}_{w_k^{[t]}} f / g_{w_k^{[t]}}(w_k^{[t]}, w_k^{[t]})$, where $g_{w_k^{[t]}}$ is the Riemannian metric (27) and $\mathrm{grad}_{w_k^{[t]}} f \in \mathcal{H}_{w_k}\mathcal{M}$ is the Riemannian gradient (29). Therefore, the sequence of the iterates is given by $w_k^{[t+1]} = \mathcal{R}_{w_k^{[t]}}(\alpha_t \eta)$, where the stepsize $\alpha_t > 0$ and

$$\mathcal{R}_{w_k}(\xi) = w_k + \xi, \qquad (35)$$

with $\xi \in \mathcal{H}_{w_k}\mathcal{M}$ [27]. Here, the retraction map $\mathcal{R}_{w_k} : \mathcal{H}_{w_k}\mathcal{M} \to \mathcal{M}$ is an approximation of the exponential map that characterizes the motion of "moving along geodesics on the Riemannian manifold". More details on computing the retraction are available in [29, Sec. 4.1.2]. The statistical analysis of the Riemannian gradient descent algorithm will be provided in the sequel, which demonstrates the linear rate of the proposed algorithm for converging to the ground truth signals.

2) Riemannian Trust-Region Algorithm: We first consider the setting that searching the direction η_{w_k} on the horizontal space $\mathcal{H}_{w_k}\mathcal{M}$, which paves the way to search the direction on the horizontal space $\mathcal{H}_V\mathcal{M}^s$. At each iteration, let $w_k \in \mathcal{M}$, we solve the trust-region sub-problem [29]:

$$\begin{aligned}\underset{\eta_{w_k}}{\mathrm{minimize}} \quad & m(\eta_{w_k}) \\ \mathrm{subject\ to} \quad & g_{w_k}(\eta_{w_k}, \eta_{w_k}) \leq \delta^2, \qquad (36)\end{aligned}$$

Algorithm 1 Riemannian Gradient Descent With Spectral Initialization

Given: Riemannian manifold $\mathcal{M}^s$ with optimization-related ingredients, objective function f, $\{c_{ij}\}_{1 \leq i \leq s, 1 \leq j \leq m}$, $\{b_j\}_{1 \leq j \leq m}$, $\{y_j\}_{1 \leq j \leq m}$ and the stepsize α.

Output: $v = \{w_k\}_{k=1}^{s}$

1: **Spectral Initialization:**
2: **for all** $i = 1, \cdots, s$ **do in parallel**
3:    Let $\sigma_1(N_i)$, $\check{h}_i^0$ and $\check{x}_i^0$ be the leading singular value, left singular vector and right singular vector of matrix $N_i := \sum_{j=1}^{m} y_j b_j c_{ij}^{\mathsf{H}}$, respectively.
4:    Set $w_i^{[0]} = \begin{bmatrix} x_i^0 \\ h_i^0 \end{bmatrix}$ where $x_i^0 = \sqrt{\sigma_1(N_i)}\check{x}_i^0$ and $h_i^0 = \sqrt{\sigma_1(N_i)}\check{h}_i^0$.
5: **end for**
6: **for all** $t = 1, \cdots, T$
7:    **for all** $i = 1, \cdots, s$ **do in parallel**
8:        $\eta = -\frac{1}{g_{w_k^{[t]}}(w_k^{[t]}, w_k^{[t]})}\mathrm{grad}_{w_k^{[t]}} f$
9:        Update $w_k^{[t+1]} = \mathcal{R}_{w_k^{[t]}}(\alpha_t \eta)$
10:    **end for**
11: **end for**

---

where $\eta_{w_k} \in \mathcal{H}_{w_k}\mathcal{M}$, δ denotes the trust-region radius and the cost function is given by

$$m(\eta_{w_k}) = g_{w_k}(\eta_{w_k}, \mathrm{grad}_{w_k} f) + \frac{1}{2}g_{w_k}(\eta_{w_k}, \mathrm{Hess}_{w_k} f[\eta_{w_k}]), \qquad (37)$$

with $\mathrm{grad}_{w_k} f$ and $\mathrm{Hess}_{w_k} f[\eta_{w_k}]$ as the matrix representations of the Riemannian gradient and Riemannian Hessian in the quotient space, respectively. Problem (36) is solved by truncated conjugate gradient method [29, Sec. 7.3]. The Riemannian trust-region algorithm is presented in Algorithm 2.

Let $w_k^{[t]}$ denotes the t-th iterate. We introduce a quotient to determine whether updating the iterate w_k and how to select the trust-region radius implemented in the next iteration. This quotient is given by $\rho_t = (f(v)|_{w_k = w_k^{[t]}} - f(v)|_{w_k = \mathcal{R}_{w_k}(w_k^{[t]})})/(m(\mathbf{0}_{w_k^{[t]}}) - m(\eta_{w_k^{[t]}}))$ [29]. The detailed strategy is introduced in the following: reduce the trust-region radius and keep the iterate unchanged, if the quotient ρ_t is extremely small. Expand the trust-region radius and maintain

---

Algorithm 2 Riemannian Trust-Region Algorithm

Given: Riemannian manifold $\mathcal{M}^s$ with Riemannian metric $g_{\boldsymbol{v}}$, retraction mapping $\mathcal{R}_{\boldsymbol{v}} = \{\mathcal{R}_{\boldsymbol{w}_k}\}_{k=1}^s$, objective function f and the stepsize α.
Output: $\boldsymbol{v} = \{\boldsymbol{w}_k\}_{k=1}^s$
1: **Initialize:** initial point $\boldsymbol{v}^{[0]} = \{\boldsymbol{w}_k^{[0]}\}_{k=1}^s$, $t = 0$
2: **while** not converged **do**
3:   **for all** $k = 1, \cdots, s$ **do**
4:     Compute a descent direction $\boldsymbol{\eta}$ via implementing trust-region method
5:     Update $\boldsymbol{w}_k^{[t+1]} = \mathcal{R}_{\boldsymbol{w}_k^{[t]}}(\alpha \boldsymbol{\eta})$
6:     $t = t + 1$.
7:   **end for**
8: **end while**

---

the iterate when the quotient $\rho_t \gg 1$. We update the iterate if and only if the quotient ρ_t is in proper range. The new iterate is given by $\boldsymbol{w}_k^{[t+1]} = \mathcal{R}_{\boldsymbol{w}_k^{[t]}}(\boldsymbol{\eta}_{\boldsymbol{w}_k})$ [27].

C. Computational Complexity Analysis

The computational complexity of Algorithm 1 and Algorithm 2 depends on the iteration complexity (i.e., number of iterations) and iteration cost of each algorithm. In this subsection, we briefly demonstrate the computational cost in each iteration of the algorithms, which is mainly depends on the computational cost for computing the ingredients listed in Table I. More detailed and precise analysis of the iteration complexity of Algorithm 1 and Algorithm 2 are left for the future work.

Note that the linear measurement matrix $\boldsymbol{J}_{ki}$ is block and sparse, endowed with computational savings. Define $n = N + K$ and $d = NK$, then the computational cost of these components with respect to manifold $\mathcal{M}$ are showed below.

1) Objective function $f(\boldsymbol{v})$ (26): The dominant computational cost comes from computing the terms $\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})$, each of which requires a numerical cost of $O(dL)$. Other matrix operations including computing the Euclidean norm of vector of size L involve the computational cost of $O(L)$. Thus the total computational cost of $f(\boldsymbol{v})$ is $O(sdL)$.

2) Riemannian metric $g_{\boldsymbol{w}_k}$ (27): The computational cost mainly comes from computing terms $\boldsymbol{\zeta}_{\boldsymbol{w}_k}^{\mathsf{H}} \boldsymbol{\eta}_{\boldsymbol{w}_k}$ and $\boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} \boldsymbol{\zeta}_{\boldsymbol{w}_k}$. Each of them requires a numerical cost of $O(n)$.

3) Projection on the horizontal space $\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}$ via $\Pi_{\mathcal{H}_{\boldsymbol{w}_k}\mathcal{M}}$: As computing the complex value a, it involve the multiplications between row vectors of size n and column vectors of size n, which costs $O(n)$. Other simple operations (e.g., subtraction and constant multiplication) involve cost of $O(n)$.

4) Retraction $\mathcal{R}_{\boldsymbol{w}_k}$ (35): The computational cost is $O(n)$.

5) Riemannian gradient $\text{grad}_{\boldsymbol{w}_k} f$ (29): The dominant computational cost comes from computing multiplications such as $\sum_{k=1}^s [\mathcal{J}_k(\boldsymbol{w} \boldsymbol{w}^{\mathsf{H}})]_i$ and $(c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k$ with the numerical cost of $O(sd)$ and $O(d)$ respectively.

Other operations handling with vectors of size n involve the cost of $O(n)$. Therefore, the total computational cost of computing the Riemannian gradient (29) is $O(sdL)$.

6) Riemannian Hessian $\text{Hess}_{\boldsymbol{w}_k} f[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$ (32):
- The directional derivative of Euclidean gradient $\nabla_{\boldsymbol{w}_k}^2 f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}]$ (33): The computational cost for computing this operator is $O(sdL)$, similar as the one of computing $\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})$ (30).
- Projection term: According to the above analysis, the computational cost is $O(n)$.

All the geometry related operations (e.g., projection and retraction) are of linear computational complexity in n, which is computationally efficient. The operations related to the objective problem as well endow with modest computational complexity. By exploiting the admirable geometric structure of symmetric rank-one matrices, the complexity of computing Riemannian Hessian is almost the same as the one of computing Riemannian gradient, which makes second-order Riemannian algorithm yield no extra computational cost compared with first-order algorithms. The numerical results depicted in the next section will demonstrate the computational efficiency of the algorithm.

V. MAIN RESULTS

In this section, we provide the statistical guarantees of Riemannian gradient descent, i.e., Algorithm 1, for the blind demixing problem.

Without loss of generality, we assume the ground truth $\|\boldsymbol{x}_k^{\natural}\|_2 = \|\boldsymbol{h}_k^{\natural}\|_2$ for $k = 1, \cdots, s$ and define the condition number $\kappa = \frac{\max_k \|\boldsymbol{x}_k^{\natural} \boldsymbol{h}_k^{\natural \mathsf{H}}\|_F}{\min_k \|\boldsymbol{x}_k^{\natural} \boldsymbol{h}_k^{\natural \mathsf{H}}\|_F}$ with $\max_k \|\boldsymbol{x}_k^{\natural} \boldsymbol{h}_k^{\natural \mathsf{H}}\|_F = 1$. Recall the definition of $\boldsymbol{w}_k = [\boldsymbol{x}_k^{\mathsf{H}} \ \boldsymbol{h}_k^{\mathsf{H}}]^{\mathsf{H}} \in \mathbb{C}^{N+K}$ and we define the notion $\boldsymbol{v} = [\boldsymbol{w}_1^{\mathsf{H}} \cdots \boldsymbol{w}_s^{\mathsf{H}}]^{\mathsf{H}} \in \mathbb{C}^{s(N+K)}$. In practical scenario, the reference symbol for the signal from each user can be exploited to eliminate the ambiguities for blind demixing problem. In this paper, considering the ambiguities of the estimated signals, we define the discrepancy between the estimate $\boldsymbol{v}$ and the ground truth $\boldsymbol{v}^{\natural}$ as the distance function, given as $\text{dist}(\boldsymbol{v}, \boldsymbol{v}^{\natural}) = \left(\sum_{i=1}^s \text{dist}^2(\boldsymbol{v}_i, \boldsymbol{v}_i^{\natural}) \right)^{1/2}$, where $\text{dist}^2(\boldsymbol{v}_i, \boldsymbol{v}_i^{\natural}) = \min_{\psi_i \in \mathbb{C}} (\|\frac{1}{\psi_i} \boldsymbol{h}_i - \boldsymbol{h}_i^{\natural}\|_2^2 + \|\psi_i \boldsymbol{x}_i - \boldsymbol{x}_i^{\natural}\|_2^2)/d_i$ for $i = 1, \cdots, s$. Here, $d_i = \|\boldsymbol{h}_i^{\natural}\|^2 + \|\boldsymbol{x}_i^{\natural}\|^2$ and each ψ_i is the alignment parameter. In addition, let the incoherence parameter μ be the smallest number such that $\max_{1 \leq k \leq s, 1 \leq j \leq m} \frac{|\boldsymbol{b}_j^{\mathsf{H}} \boldsymbol{h}_k^{\natural}|}{\|\boldsymbol{h}_k^{\natural}\|_2} \leq \frac{\mu}{\sqrt{m}}$. The main theorem is presented in the following.

Theorem 1: Suppose the rows of the encoding matrices, i.e., $\boldsymbol{c}_{ij}$'s, follow the i.i.d. complex Gaussian distribution, i.e., $\boldsymbol{c}_{ij} \sim \mathcal{N}(0, \frac{1}{2} \boldsymbol{I}_N) + i\mathcal{N}(0, \frac{1}{2} \boldsymbol{I}_N)$ and the step size obeys $\alpha_t > 0$ and $\alpha_t \equiv \alpha \asymp s^{-1}$, then the iterates (including the spectral initialization point) in Algorithm 1 satisfy $\text{dist}(\boldsymbol{v}^t, \boldsymbol{v}^{\natural}) \leq C_1 (1 - \frac{\alpha}{16\kappa})^t \frac{1}{\log^2 L}$ for all $t \geq 0$ and some constant $C_1 > 0$, with probability at least $1 - c_1 L^{-\gamma} - c_1 L e^{-c_2 K}$ if the number of measurements $L \geq C\mu^2 s^2 \kappa^4 \max\{K, N\} \log^8 L$ for some constants $\gamma, c_1, c_2 > 0$ and sufficiently large constant $C > 0$.

Proof: Please refer to Appendix B for details. $\square$

Theorem 1 demonstrates that number of measurements $\mathcal{O}(s^2\kappa^4 \max\{K, N\} \log^8 L)$ are sufficient for the Riemannian gradient descent algorithm (with spectral initialization), i.e., Algorithm 1, to linearly converge to the ground truth signals.

VI. SIMULATION RESULTS

In this section, we simulate our proposed Riemannian optimization algorithm for the blind demixing problem in the settings of Hadamard-type encoding matrices and Gaussian encoding matrices to demonstrate the algorithmic advantages and good performance. In the noiseless scenario (i.e., $\boldsymbol{e} = 0$), we will study the number of measurements necessary for exact recovery in order to illustrate the performance of the proposed algorithm. The convergence rate of different algorithms will be also compared. In the noisy scenario, we study the average relative construction error of different algorithms. The robustness of the proposed algorithm for noisy data is simultaneously demonstrated.

A. Simulation Settings and Performance Metric

The simulation settings are given as follows:

- Ground truth rank-one matrices $\{\hat{\boldsymbol{X}}_k\}_{k=1}^s$: In the OFDM system, the elements of symbol $\boldsymbol{q}_k \in \mathbb{R}^N$ are chosen randomly from the integer set $\{0, 1, \cdots, 14, 15\}$. The complex vector $\boldsymbol{x}_k$ is thus generated according to (5), where $\boldsymbol{s} \in \mathbb{C}^N$ is the 16-QAM symbol constellation corresponding to the symbol $\boldsymbol{q}$. In the other scenario, entries of the standard complex Gaussian vector $\boldsymbol{x}_k$ are drawn i.i.d from the standard normal distribution. With standard complex Gaussian vectors $\boldsymbol{h}_k \in \mathbb{C}^K$ and the complex vector $\boldsymbol{x}_k \in \mathbb{C}^N$, the matrices $\{\hat{\boldsymbol{X}}_k\}_{k=1}^s$ are generated as $\{\hat{\boldsymbol{X}}_k\}_{k=1}^s = \{\boldsymbol{x}_k \bar{\boldsymbol{h}}_k^{\mathsf{H}}\}_{k=1}^s$ [17].
- Measurement matrices $\{\{\boldsymbol{J}_{ik}\}_{i=1}^L\}_{k=1}^s$: We generate the normalized discrete Fourier transform (DFT) matrix $\boldsymbol{F} \in \mathbb{C}^{L \times L}$ and the Hadamard-type matrix $\boldsymbol{C}_k \in \mathbb{C}^{L \times N}$ according to (8) for $k = 1, \cdots, s$ and to construct the measurement matrices according to (11) and (18).
- Performance metric: The relative construction error is adopted to evaluate the performance of the algorithms, given as [17]

$$\text{err}(\boldsymbol{X}) = \frac{\sqrt{\sum_{k=1}^s \|\boldsymbol{X}_k - \hat{\boldsymbol{X}}_k\|_F^2}}{\sqrt{\sum_{k=1}^s \|\hat{\boldsymbol{X}}_k\|_F^2}}, \qquad (38)$$

where $\{\boldsymbol{X}_k\}_{k=1}^s$ are estimated matrices and $\{\hat{\boldsymbol{X}}_k\}$ are ground truth matrices.

The following five algorithms are compared:

- **Proposed Riemannian trust-region algorithm (PRTR)**: We use the manifold optimization toolbox *Manopt* [31] to implement the proposed Riemannian trust-region algorithm (PRTR). The initial trust region radius is $\delta = 2$.
- **Proposed Riemannian gradient descent algorithm (PRGD)**: The proposed Riemannian gradient descent algorithm (PRGD) is implemented via the manifold optimization toolbox *Manopt* [31].
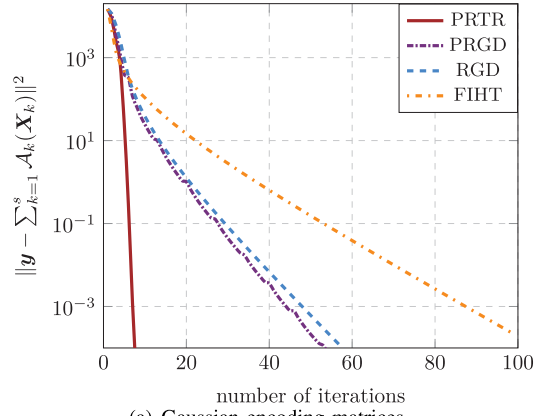
(a) Gaussian encoding matrices
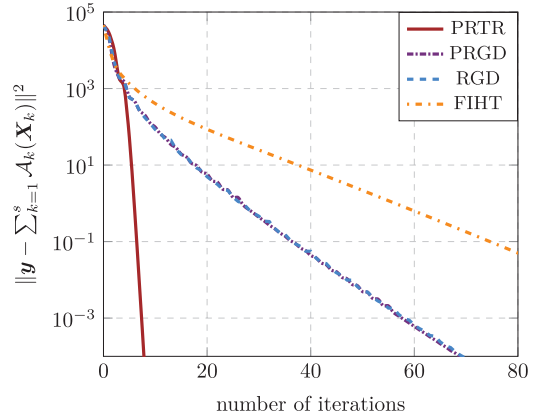
(b) Hadamard-type encoding matrices

Fig. 3.   Convergence rate of different algorithms with respect to the number of iterations.

- **Nuclear norm minimization (NNM)**: The algorithm [14] is implemented with the toolbox CVX [32] to solve the convex problem (13) with the parameter $\varepsilon = 10^{-9}$ in noiseless scenario and $\varepsilon = 10^{-2}$ in noisy scenario.
- **Regularized gradient descent (RGD)**: This algorithm [17] is implemented to solve the regularized problem (14).
- **Fast Iterative Hard Thresholding (FIHT)**: The algorithm [18] utilizes hard thresholding to solve the rank-constraint problem $\mathscr{P}$ directly.

We adopt the initialization strategy in [18] for all the non-convex optimization algorithms (i.e., PRTR, PRGD, RGD and FIHT). The PRTR and PRGD algorithm stop when the norm of Riemannian gradient falls below 10^{-8} or the number of iterations exceeds 500. The stopping criteria of RGD and FIHT are based on [17] and [18], respectively.

B. Convergence Rate

Fig. 3(a) and Fig. 3(b) illustrate the convergence rate of different non-convex algorithms with respect to the number of iterations in the setting of Gaussian encoding matrices with $N = K = 50$, $L = 1250$, $s = 5$ and in the setting of Hadamard-type encoding matrices with $N = K = 50$, $L = 1536$, $s = 5$, respectively. Under the corresponding
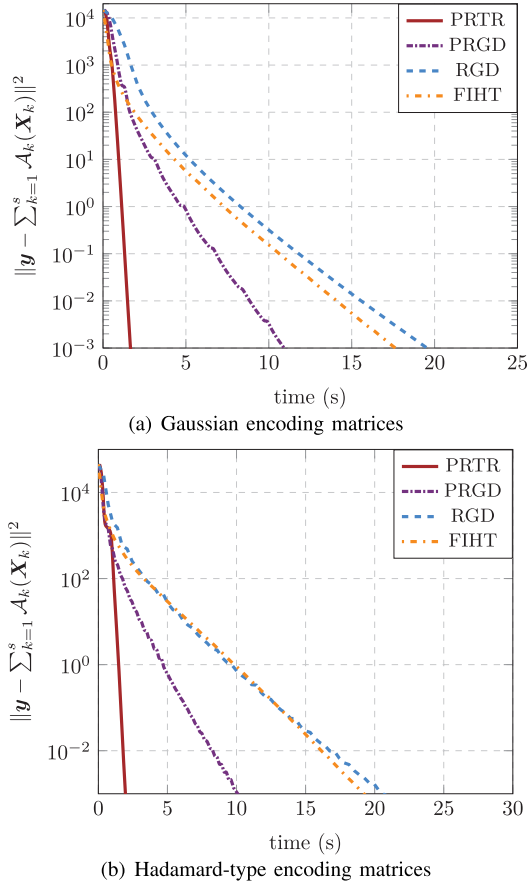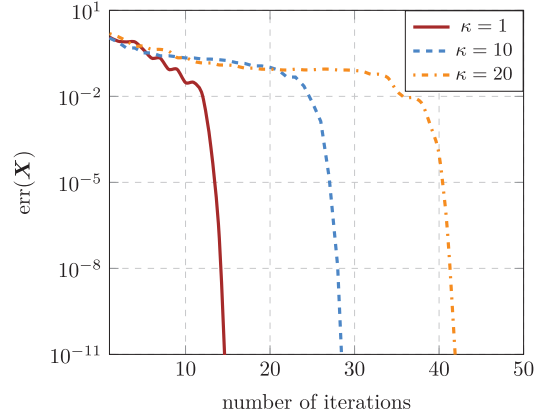
(a) Gaussian encoding matrices

(b) Hadamard-type encoding matrices

Fig. 4.   Convergence rate of different algorithms with respect to time.

Fig. 5.   Convergence rate of the proposed trust-region algorithm with respect to different κ.

settings, Fig. 4(a) and Fig. 4(b) show the convergence rate of different non-convex algorithms with respect to the time in the settings of Hadamard-type encoding matrices and Gaussian encoding matrices, respectively. From these figures, we can see that in both scenarios, the iteration complexity of the proposed Riemannian gradient descent algorithm is comparable to the regularized gradient descent and has lower time complexity that regularized gradient descent. Moreover, the Riemannian trust-region algorithm, which enjoys superlinear convergence rate, significantly converges faster than the stat-of-the-art non-convex algorithms with respect to both the number of iterations and time. The proposed algorithm thus enjoys low iteration complexity and low time complexity. Next we investigate the impact of the condition number, i.e., $\kappa = \frac{\max \|\hat{X}_k\|_F}{\min \|\hat{X}_k\|_F}$ where $\hat{X}_k$ is the ground truth, on the convergence rate of the proposed Riemannian algorithm. In this simulation, we set $s = 2$ and set the first component as $\|\hat{X}_1\|_F = 1$ and the second one as $\|\hat{X}_2\|_F = \kappa$ where $\kappa \in \{1, 10, 20\}$. Therein, $\kappa = 1$ means both sensors receive the signals with equal power and $\kappa = 10$ means the second sensor has considerably stronger received signals [17]. Fig. 5 demonstrates the relative error (38) vs. iterations for the proposed Riemannian trust-region algorithm. It shows that even though large condition number yields slightly slow convergence rate, the proposed Riemannian trust-region algorithm can still precisely recover the original signals in a few iterations. However, the gradient descent algorithm in [17] has less satisfied signal recovery performance when the

condition number is large. Therefore, our proposed second-order algorithm is robust to the condition number compared with the first-order algorithm in [17].

C. Phase Transitions

In this subsection, we investigate the empirical recovery performance of PRTR without considering the noise and compare the proposed algorithm with other algorithms. In setting of Gaussian encoding matrices, we set $N = K = 50$, $L = 1000$ with the number of devices s varying from 1 to 12. In the setting of Hadamard-type encoding matrices, we set $N = K = 16$, $L = 1536$ with s varying from 1 to 45. For each setting, 10 independent trails are performed and the recovery is treated as a success if the relative construction error $\text{err}(\boldsymbol{X}) \leq 10^{-3}$. Fig. 6(a) and Fig. 6(b) show the probability of successful recovery for different numbers of devices s in the settings of Hadamard-type encoding matrices and Gaussian encoding matrices, respectively. Based on the phase transitions results in two figures, we can see that the PRTR and PRGD algorithm outperform in terms of guaranteeing exact recovery than other three algorithms. The non-uniqueness of the factorization taken into account in the quotient manifold space plays a vital role to lead this advantage. In particular, the proposed algorithm, i.e., the Riemannian gradient algorithm and the Riemannian trust-region algorithm are more robust to the practical scenario than NNM and FIHT algorithm do.

D. Average Relative Construction Error

We study the average relative construction error of four algorithms and explore the robustness of the proposed Riemannian trust-region algorithm against additive noise by considering the model (10). We assume the additive noise in the formulation (10) satisfies [18]

$$\boldsymbol{e} = \sigma \cdot \|\boldsymbol{y}\| \cdot \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}, \qquad (39)$$

where $\boldsymbol{\omega} \in \mathbb{C}^L$ is a standard complex Gaussian vector. We compare the four algorithms for each level of signal to noise ratio (SNR) σ in the setting of Gaussian encoding matrices with $L = 1500$, $N = K = 50$, $s = 2$ and in the
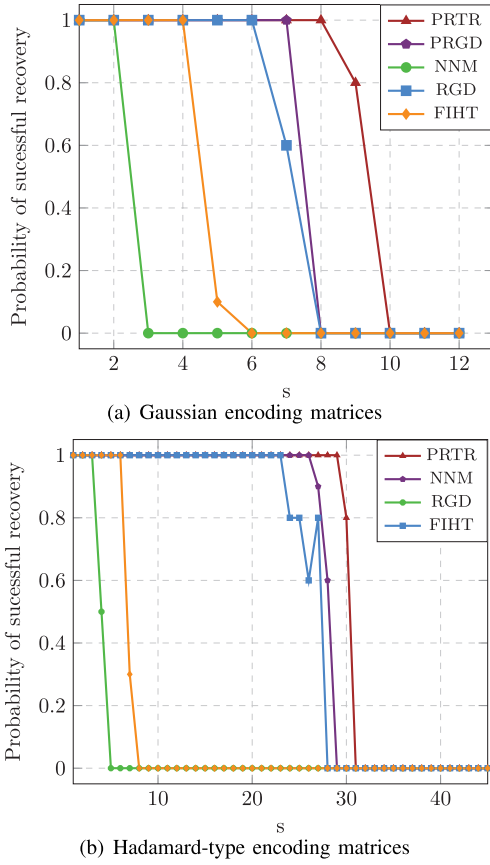
Fig. 6.   Probability of successful recovery with different numbers of devices s.

Fig. 7.   Average relative construction error vs. SNR (dB).

setting of Hadamard-type encoding matrices with $L = 1536$, $N = K = 16$ and $s = 2$. For each setting, 10 independent trails are performed and the condition of successful recovery is the same with the one aforementioned. The average relative construction error in dB against the signal to noise ratio (SNR) in the settings of Hadamard-type encoding matrices and Gaussian encoding matrices are illustrated in Fig. 7(a) and Fig. 7(b), respectively. It depicts that the relative reconstruction error of the proposed algorithm linearly scales with SNR. We conclude that the PRTR and PRGD algorithm are stable in the presence of noise as the other algorithms. The figure also shows that the proposed algorithm PRTR and PRGD achieve lower average relative construction error than other three algorithms, yielding better performance in the practical scenario.

The impressive simulation results are in favor of the quotient manifold of the product of complex symmetric rank-one matrices which is established via Hermitian reformulation. By exploiting the geometry of quotient manifold which takes into account the non-uniqueness of the factorization, both first-order and second-order Riemannian optimization algorithm are developed. Specifically, the proposed Riemannian optimization algorithms, i.e., the Riemannian gradient descent algorithm and the Riemannian trust-region algorithm outperform state-of-the-art algorithms in terms of the algorithmic advantages (i.e., fast convergence rate and low iteration cost) and performance (i.e., sample complexity). Moreover, both of them are
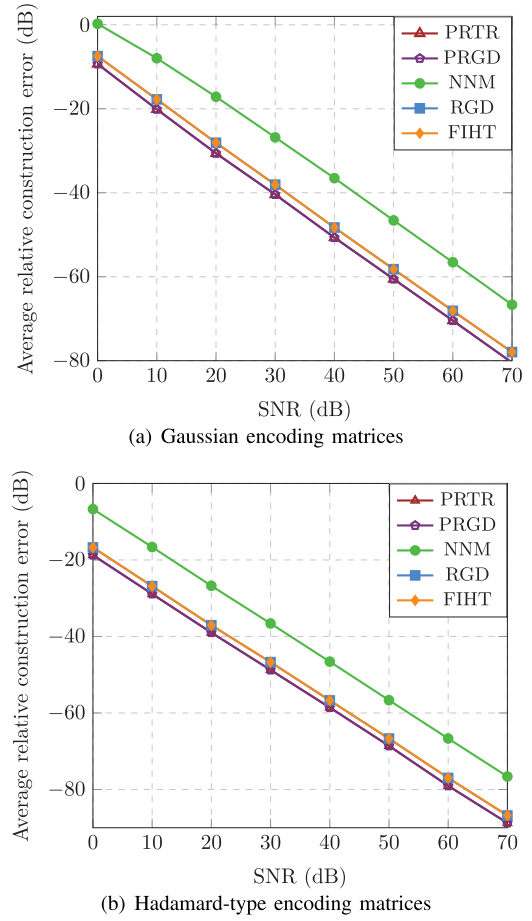
robust to the noise and the Riemannian trust-region algorithm is also robust to the condition number, i.e., $\kappa = \frac{\max \|\hat{X}_k\|_F}{\min \|\hat{X}_k\|_F}$.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented the blind demixing approach to support low-latency communication without any channel estimation in OFDM system, for which a low-rank modeling approach with respect to rank-one matrices is further developed. To address the unique challenge of multiple asymmetric complex rank-one matrices as well as develop efficient algorithm, we exploited the Riemannian quotient geometry of product of complex symmetric rank-one matrices via reformulating problems on complex asymmetric matrices to problems on Hermitian positive semidefinite (PSD) matrices. Specifically, by exploiting the admirable structure of symmetric rank-one matrices, we developed scaled Riemannian optimization algorithms, i.e., the Riemannian gradient descent algorithm and the Riemannian trust-region algorithm. We proof that the Riemannian gradient descent algorithm linearly converges to ground truth signals with high probability provided sufficient measurement. Simulation results demonstrated that the proposed algorithms are robust to the additive noise and outperforms state-of-the-art algorithms in terms of algorithmic advantages and performance in

the practical scenario. Moreover, the Riemannian trust-region algorithm is robust to the condition number.

APPENDIX A
COMPUTING THE RIEMANNIAN GRADIENT AND RIEMANNIAN HESSIAN

We first reformulate the objective function in (26) as

$$f(\boldsymbol{v}) = \sum\nolimits_{i=1}^{L} \left(\sum\nolimits_{k=1}^{s} [\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})]_i - y_i \right)^* $$
$$\times \left(\sum\nolimits_{k=1}^{s} [\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})]_i - y_i \right) = \sum\nolimits_{i=1}^{L} c_i^* \cdot c_i, \tag{40}$$

where $c_i = \sum_{k=1}^{s} [\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})]_i - y_i$. The partial gradient of $f(\boldsymbol{v})$ with respect to the complex vector $\boldsymbol{w}_k$ is given as

$$\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v}) = 2 \frac{\partial f(\boldsymbol{v})}{\partial \boldsymbol{w}_k^{\mathsf{H}}}$$
$$= 2 \sum\nolimits_{i=1}^{L} \left(\sum\nolimits_{k=1}^{s} [\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})]_i - y_i \right) \boldsymbol{J}_{ki} \boldsymbol{w}_k$$
$$+ \left(\sum\nolimits_{k=1}^{s} [\mathcal{J}_k(\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{H}})]_i - y_i \right)^* \boldsymbol{J}_{ki}^{\mathsf{H}} \boldsymbol{w}_k$$
$$= 2 \cdot \sum\nolimits_{i=1}^{L} (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k. \tag{41}$$

Furthermore, the Euclidean gradient of $f(\boldsymbol{v})$ in the direction $\boldsymbol{\eta}_{\boldsymbol{w}_k}$ with respect to $\boldsymbol{w}_k$ is derived as

$$\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \lim_{t \to 0} \frac{f(\boldsymbol{v})|_{\boldsymbol{w}_k + t\boldsymbol{\eta}_k} - f(\boldsymbol{v})|_{\boldsymbol{w}_k}}{t}$$
$$= \operatorname{Tr}\Big[\sum\nolimits_{i=1}^{L} \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \boldsymbol{w}_k$$
$$+ \boldsymbol{w}_k^{\mathsf{H}} (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \boldsymbol{\eta}_{\boldsymbol{w}_k} \Big]$$
$$= g_{\boldsymbol{w}_k} \Big(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \sum\nolimits_{i=1}^{L} (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k \Big)$$
$$= g_{\boldsymbol{w}_k} \Big(\boldsymbol{\eta}_{\boldsymbol{w}_k}, \frac{1}{2} \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v}) \Big). \tag{42}$$

Thus, according to (28), there is $\operatorname{grad}_{\boldsymbol{w}_k} f = \frac{1}{2} \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})$. With the fact that $(\operatorname{grad}_{\boldsymbol{w}_k} f)^{\mathsf{H}} \boldsymbol{w}_k = \boldsymbol{w}_k^{\mathsf{H}} \operatorname{grad}_{\boldsymbol{w}_k} f$, we conclude that $\operatorname{grad}_{\boldsymbol{w}_k} f$ is already in the horizontal space $\mathcal{V}_{\boldsymbol{w}_k} \mathcal{M}$. Therefore, the matrix representation of Riemannian gradient is written as

$$\operatorname{grad}_{\boldsymbol{w}_k} f = \sum\nolimits_{i=1}^{L} (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k. \tag{43}$$

To compute the Riemannian Hessian, we first derive the directional derivative of Euclidean gradient $\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})$ (30) in the direction of $\boldsymbol{\eta}_{\boldsymbol{w}_k} \in \mathcal{H}_{\boldsymbol{w}_k} \mathcal{M}$, given by

$$\nabla_{\boldsymbol{w}_k}^2 f(\boldsymbol{v})[\boldsymbol{\eta}_{\boldsymbol{w}_k}] = \lim_{t \to 0} \frac{\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})|_{\boldsymbol{w}_k + t\boldsymbol{\eta}_k} - \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})|_{\boldsymbol{w}_k}}{t}$$
$$= \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0} \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})|_{\boldsymbol{w}_k + t\boldsymbol{\eta}_k}$$
$$= 2 \sum\nolimits_{i=1}^{L} (b_i \boldsymbol{J}_{ki} + b_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{w}_k$$
$$+ (c_i \boldsymbol{J}_{ki} + c_i^* \boldsymbol{J}_{ki}^{\mathsf{H}}) \cdot \boldsymbol{\eta}_{\boldsymbol{w}_k}, \tag{44}$$

where $b_i = \sum_{k=1}^{s} \langle \boldsymbol{J}_{ki}, \boldsymbol{\eta}_{\boldsymbol{w}_k} \boldsymbol{w}_k^{\mathsf{H}} + \boldsymbol{w}_k \boldsymbol{\eta}_{\boldsymbol{w}_k}^{\mathsf{H}} \rangle$. Thus, the matrix representation of Riemannian Hessian is derived according to (34).

APPENDIX B
PROOF OF THEOREM 1

According to the definition of the horizontal space in Proposition 1, we know that $\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})$ is in the horizontal space due to $\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})^{\mathsf{H}} \boldsymbol{w}_k = \boldsymbol{w}_k^{\mathsf{H}} \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})$. Based on this fact, the update rule in the Riemannian gradient descent algorithm, i.e., Algorithm 1, can be reformulated as

$$\boldsymbol{w}_k^{[t+1]} = \boldsymbol{w}_k^{[t]} - \frac{\alpha_t}{2\|\boldsymbol{w}_k^{[t]}\|_2^2} \nabla_{\boldsymbol{w}_k} f(\boldsymbol{v})\big|_{\boldsymbol{w}_k^{[t]}}, \tag{45}$$

according to the definition of the Riemannian metric $g_{\boldsymbol{w}_k}$ (27) and the retraction $\mathcal{R}_{\boldsymbol{w}_k}$ (35). The update rule (45) can be further modified as

$$\begin{bmatrix} \boldsymbol{w}_k^{[t+1]} \\ \boldsymbol{w}_k^{[t+1]} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_k^{[t]} \\ \boldsymbol{w}_k^{[t]} \end{bmatrix} - \frac{\alpha_t}{\|\boldsymbol{w}_k^{[t]}\|_2^2} \begin{bmatrix} \frac{\partial f}{\partial \boldsymbol{w}_k^{\mathsf{H}}}\big|_{\boldsymbol{w}_k^{[t]}} \\ \frac{\partial f}{\partial \boldsymbol{w}_k^{\mathsf{H}}}\big|_{\boldsymbol{w}_k^{[t]}} \end{bmatrix}, \tag{46}$$

based on the fact that $\nabla_{\boldsymbol{w}_k} f(\boldsymbol{v}) = 2 \frac{\partial f(\boldsymbol{v})}{\partial \boldsymbol{w}_k^{\mathsf{H}}}$.

To proof Theorem 1, under the assumption that the rows of the encoding matrices $\boldsymbol{c}_{ij} \sim \mathcal{N}(0, \frac{1}{2}\boldsymbol{I}_N) + i\mathcal{N}(0, \frac{1}{2}\boldsymbol{I}_N)$, we first characterize the local geometry in the region of incoherence and contraction (RIC) where the objective function enjoys restricted strong convexity and smoothness near the ground truth $\boldsymbol{v}^{\natural}$, please refer to Lemma 1. The error contraction, i.e., convergence analysis, is further established in Lemma 2 based on the property of the local geometry. We then exploit the induction arguments to demonstrate that the iterates of Algorithm 1, including the spectral initialization point, stay within the RIC, please refer to Lemma 3.

Definition 1 $((\phi, \beta, \gamma, \boldsymbol{z}^{\natural}) - \mathcal{R}$ the Region of Incoherence and Contraction): Define $\boldsymbol{z}_i = [\boldsymbol{x}_i^{\mathsf{H}} \ \boldsymbol{h}_i^{\mathsf{H}}]^{\mathsf{H}} \in \mathbb{C}^{N+K}$ and $\boldsymbol{z} = [\boldsymbol{z}_1^{\mathsf{H}} \cdots \boldsymbol{z}_s^{\mathsf{H}}]^{\mathsf{H}} \in \mathbb{C}^{s(N+K)}$. *If $\boldsymbol{z}$ is in the region of incoherence and contraction (RIC), i.e., $\boldsymbol{z} \in (\phi, \beta, \gamma, \boldsymbol{z}^{\natural}) - \mathcal{R}$, it holds that*

$$\operatorname{dist}(\boldsymbol{z}^t, \boldsymbol{z}^{\natural}) \leq \phi, \tag{47a}$$
$$\max_{1 \leq i \leq s, 1 \leq j \leq m} \left| \boldsymbol{c}_{ij}^{\mathsf{H}} \left(\widetilde{\boldsymbol{x}}_i^t - \boldsymbol{x}_i^{\natural} \right) \right| \cdot \|\boldsymbol{x}_i^{\natural}\|_2^{-1} \leq C_3 \beta, \tag{47b}$$
$$\max_{1 \leq i \leq s, 1 \leq j \leq m} \left| \boldsymbol{b}_j^{\mathsf{H}} \widetilde{\boldsymbol{h}}_i^t \right| \cdot \|\boldsymbol{h}_i^{\natural}\|_2^{-1} \leq C_4 \gamma, \tag{47c}$$

for some constants $C_3, C_4 > 0$ and some sufficiently small constants $\phi, \beta, \gamma > 0$. Here, $\widetilde{\boldsymbol{h}}_i^t$ and $\widetilde{\boldsymbol{x}}_i^t$ are defined as $\widetilde{\boldsymbol{h}}_i^t = \frac{1}{\psi_i^t} \boldsymbol{h}_i^t$ and $\widetilde{\boldsymbol{x}}_i^t = \psi_i^t \boldsymbol{x}_i^t$ for $i = 1, \cdots, s$, where ψ_i^t is the alignment parameter.

The Riemannian Hessian is denoted as $\operatorname{Hess} f(\boldsymbol{v}) := \operatorname{diag}(\{\operatorname{Hess}_{\boldsymbol{w}_i} f\}_{i=1}^{s})$.

Lemma 1: Suppose there is a sufficiently small constant $\delta > 0$. If the number of measurements obeys $m \gg \mu^2 s^2 \kappa^2 \max\{N, K\} \log^5 m$, then with probability at least $1 - \mathcal{O}(m^{-10})$, the Riemannian Hessian $\operatorname{Hess} f(\boldsymbol{v})$ obeys

$$\boldsymbol{u}^{\mathsf{H}} [\boldsymbol{D} \operatorname{Hess} f(\boldsymbol{v}) + \operatorname{Hess} f(\boldsymbol{v}) \boldsymbol{D}] \boldsymbol{u} \geq \frac{1}{4\kappa} \|\boldsymbol{u}\|_2^2$$
$$\text{and } \|\operatorname{Hess} f(\boldsymbol{v})\| \leq 2 + s \tag{48}$$

simultaneously for all $\boldsymbol{u} = [\boldsymbol{u}_1^{\mathsf{H}} \cdots \boldsymbol{u}_s^{\mathsf{H}}]^{\mathsf{H}}$ with $\boldsymbol{u}_i = [(\boldsymbol{x}_i - \boldsymbol{x}_i')^{\mathsf{H}} \ (\boldsymbol{h}_i - \boldsymbol{h}_i')^{\mathsf{H}} \ (\boldsymbol{x}_i - \boldsymbol{x}_i')^{\top} \ (\boldsymbol{h}_i - \boldsymbol{h}_i')^{\top}]^{\mathsf{H}}$, and $\boldsymbol{D} =$

$\text{diag}\left(\{\boldsymbol{W}_i\}_{i=1}^s\right)$ with $\boldsymbol{W}_i = \text{diag}([\overline{\beta}_{i1}\boldsymbol{I}_K \ \overline{\beta}_{i2}\boldsymbol{I}_N \ \overline{\beta}_{i1}\boldsymbol{I}_K \ \overline{\beta}_{i2}\boldsymbol{I}_N]^*).$

Here $\boldsymbol{v}$ is in the region $(\delta, \frac{1}{\sqrt{s}\log^{3/2}m}, \frac{\mu}{\sqrt{m}}\log^2 m, \boldsymbol{v}^\natural) - \mathcal{R}$, and one has $\max\{\|\boldsymbol{h}_i - \boldsymbol{h}_i^\natural\|_2, \|\boldsymbol{h}_i' - \boldsymbol{h}_i^\natural\|_2, \|\boldsymbol{x}_i - \boldsymbol{x}_i^\natural\|_2, \|\boldsymbol{x}_i' - \boldsymbol{x}_i^\natural\|_2\} \leq \delta/(\kappa\sqrt{s})$, for $i = 1, \cdots, s$ and $\boldsymbol{W}_i$'s satisfy that for $\beta_{i1}, \beta_{i2} \in \mathbb{R}$, for $i = 1, \cdots, s$ $\max_{1 \leq i \leq s} \max\left\{|\beta_{i1} - \frac{1}{\kappa}|, |\beta_{i2} - \frac{1}{\kappa}|\right\} \leq \frac{\delta}{\kappa\sqrt{s}}$. Therein, $C_3, C_4 \geq 0$ are numerical constants.

Lemma 2: Suppose the number of measurements satisfies $m \gg \mu^2 s^2 \kappa^4 \max\{N, K\}\log^5 m$ and the step size obeys $\alpha_t > 0$ and $\alpha_t \equiv \alpha \asymp s^{-1}$. Then with probability at least $1 - \mathcal{O}(m^{-10})$,

$$\text{dist}(\boldsymbol{v}^{t+1}, \boldsymbol{v}^\natural) \leq (1 - \frac{\alpha}{16\kappa})\text{dist}(\boldsymbol{v}^t, \boldsymbol{v}^\natural), \tag{49}$$

provided that $\boldsymbol{v}$ is in the region $(\delta, \frac{1}{\sqrt{s}\log^{3/2}m}, \frac{\mu}{\sqrt{m}}\log^2 m, \boldsymbol{v}^\natural) - \mathcal{R}$.

Lemma 3: The spectral initialization point $\boldsymbol{v}^0$ is in the region $(\frac{1}{\log m}, \frac{1}{\sqrt{s}\log^{3/2}m}, \frac{\mu}{\sqrt{m}}\log^2 m, \boldsymbol{v}^\natural) - \mathcal{R}$ with probability at least $1 - \mathcal{O}(m^{-9})$, provided $m \gg \mu^2 s^2 \kappa^2 \max\{K, N\}\log^6 m$. Suppose t-th iteration $\boldsymbol{v}^t$ is in the region $(\frac{1}{\log m}, \frac{1}{\sqrt{s}\log^{3/2}m}, \frac{\mu}{\sqrt{m}}\log^2 m, \boldsymbol{v}^\natural) - \mathcal{R}$ and the number of measurements satisfy $m \gg \mu^2 s^2 \kappa^2 \max\{K, N\}\log^8 m$. Then with probability at least $1 - \mathcal{O}(m^{-9})$, the $(t+1)$-th iteration $\boldsymbol{v}^{t+1}$ is also in the region $(\frac{1}{\log m}, \frac{1}{\sqrt{s}\log^{3/2}m}, \frac{\mu}{\sqrt{m}}\log^2 m, \boldsymbol{v}^\natural) - \mathcal{R}$, provided that the step size satisfies $\alpha_t > 0$ and $\alpha_t \equiv \alpha \asymp s^{-1}$.

Remark 1: The proofs of Lemma 1, Lemma 2 and Lemma 3 are mainly based on the proofs of [33, Lemma 1–Lemma 7].

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th. Quart. 2015.

[2] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile Internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, Mar. 2016.

[3] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[4] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[5] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th. Quart. 2018.

[6] Y. Shi, J. Zhang, K. B. Letaief, B. Bai, and W. Chen, "Large-scale convex optimization for ultra-dense cloud-RAN," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 84–91, Jun. 2015.

[7] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[8] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[9] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[10] Y. Shi, J. Zhang, and K. B. Letaief, "Low-rank matrix completion for topological interference management by Riemannian pursuit," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4703–4717, Jul. 2016.

[11] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[12] S. Ling and T. Strohmer, "Blind deconvolution meets blind demixing: Algorithms and performance bounds," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4497–4520, Jul. 2017.

[13] D. Stoeger, P. Jung, and F. Krahmer, "Blind demixing and deconvolution with noisy data: Near-optimal rate," in *Proc. 21st Int. ITG Workshop Smart Antennas*, Mar. 2017, pp. 1–5.

[14] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1711–1732, Mar. 2014.

[15] M. B. McCoy and J. A. Tropp, "Sharp recovery bounds for convex demixing, with applications," *Found. Comput. Math.*, vol. 14, no. 3, pp. 503–567, Jun. 2014.

[16] M. B. McCoy and J. A. Tropp, "The achievable performance of convex demixing," Caltech, Pasadena, CA, UAS, ACM Rep. 2017-02, Feb. 2017.

[17] S. Ling and T. Strohmer, "Regularized gradient descent: A non-convex recipe for fast joint blind deconvolution and demixing," *Inf. Inference J. IMA*, pp. 1–49, Mar. 2018.

[18] T. Strohmer and K. Wei, "Painless breakups—Efficient demixing of low rank matrices," *J. Fourier Anal. Appl.*, pp. 1–31, Sep. 2017.

[19] N. Boumal, P.-A. Absil, and C. Cartis, "Global rates of convergence for nonconvex optimization on manifolds," *IMA J. Numer. Anal.*, pp. 1–33, Feb. 2018.

[20] N. Boumal and P.-A. Absil, "RTRMC: A Riemannian trust-region method for low-rank matrix completion," in *Proc. NIPS*, Dec. 2011, pp. 406–414.

[21] W. Huang and P. Hand. (2017). "Blind deconvolution by a steepest descent algorithm on a quotient manifold." [Online]. Available: https://arxiv.org/abs/1710.03309

[22] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, May 2005.

[23] J. V. D. Beek, "Synchronization channel estimation OFDM syst," Ph.D. dissertation, Dept. Comput. Sci., Elect. Space Eng., Signals Syst., Luleå Univ. Technol., Luleå, Sweden, 1998.

[24] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, Aug. 2005.

[25] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, Jun. 2013.

[26] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, "Fixed-rank matrix factorizations and Riemannian low-rank optimization," *Comput. Statist.*, vol. 29, nos. 3–4, pp. 591–621, Jun. 2014.

[27] S. Yatawatta, "Radio interferometric calibration using a Riemannian manifold," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3866–3870.

[28] S. Yatawatta, "On the interpolation of calibration solutions obtained in radio interferometry," *Monthly Notices Roy. Astronomical Soc.*, vol. 428, no. 1, pp. 828–833, Jan. 2013.

[29] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, Apr. 2009.

[30] R. Ge, C. Jin, and Y. Zheng, "No spurious local minima in nonconvex low rank problems: A unified geometric analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 70, Aug. 2017, pp. 1233–1242.

[31] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a MATLAB toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, Jan. 2014.

[32] M. Grant, S. Boyd, and Y. Ye. (Mar. 2014). *CVX: MATLAB Software for Disciplined Convex Programming*. [Online]. Available: http//cvxr.com/cvx

[33] J. Dong and Y. Shi, "Nonconvex demixing from bilinear measurements," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5152–5166, Oct. 2018.

Jialin Dong (S'17) received the B.S. degree in communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2017. She is currently pursuing the degree with the School of Information Science and Technology, ShanghaiTech University. Her research interests include mathematical optimization and high-dimensional probability.

Kai Yang (S'16) received the B.S. degree in electronic engineering from the Dalian University of Technology, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Information Science and Technology, ShanghaiTech University. His research interests are in big data processing, mobile edge/fog computing, mobile artificial intelligence, and dense communication networking.
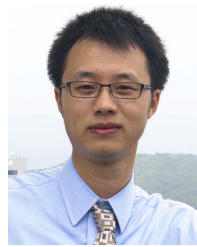
Yuanming Shi (S'13–M'15) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology, in 2015. From 2016 to 2017, he held a visiting position with the University of California, Berkeley, CA, USA. Since 2015, he has been a Tenure-Track Assistant Professor with the School of Information Science and Technology, ShanghaiTech University. His research areas include optimization, statistics, learning, and their applications to data analysis, mobile AI, and intelligent Internet of Things. He was a recipient of the 2016 IEEE Marconi Prize Paper Award in Wireless Communications. He received the 2016 Young Author Best Paper Award from the IEEE Signal Processing Society.