Joint Channel Estimation and User Grouping for Massive MIMO Systems

Jisheng Dai [ORCID], *Member, IEEE*, An Liu [ORCID], *Senior Member, IEEE*, and Vincent K. N. Lau, *Fellow, IEEE*

Abstract—This paper addresses the problem of joint downlink channel estimation and user grouping in massive multiple-input multiple-output (MIMO) systems, where the motivation comes from the fact that the channel estimation performance can be improved if we exploit additional common sparsity among nearby users. In the literature, a commonly used group sparsity model assumes that users in each group share a uniform sparsity pattern. In practice, however, this oversimplified assumption usually fails to hold, even for physically close users. Outliers deviated from the uniform sparsity pattern in each group may significantly degrade the effectiveness of common sparsity, and hence bring limited (or negative) gain for channel estimation. To better capture the group sparse structure in practice, we provide a general model having two sparsity components: commonly shared sparsity and individual sparsity, where the additional individual sparsity accounts for any outliers. Then, we propose a novel sparse Bayesian learning based framework to address the joint channel estimation and user grouping problem under the general sparsity model. The framework can fully exploit the common sparsity among nearby users and exclude the harmful effect from outliers simultaneously. Simulation results reveal substantial performance gains over the existing state-of-the-art baselines.

Index Terms—Channel estimation, user grouping, massive multiple-input multiple-output (MIMO), sparse Bayesian learning (SBL), off-grid refinement.

I. INTRODUCTION

MASSIVE multiple-input Multiple-output (MIMO) can support high spectrum and energy efficiency, and it has been widely considered as one of the key candidate technologies to meet the capacity demand for the next generation of wireless communications [1]–[3]. To fully harvest the benefit of excessive base station (BS) antennas, knowledge of channel state information at the transmitter (CSIT) is an essential requirement [4], [5]. However, it is challenging to acquire the

accurate CSIT, since the training overhead for CSIT acquisition grows proportionally with the number of BS antennas, which can be very large in such systems. Early works sidestep this challenge by adopting a time-division duplexing (TDD) model [4], [6], where the CSIT can be obtained by exploiting channel reciprocity, and the uplink pilot-aided training overhead is only proportional to the number of mobile users.

Unfortunately, channel reciprocity does not hold for massive MIMO systems with a frequency-division duplexing (FDD) model. Compared with a TDD system, an FDD system has its own advantages; e.g., it can provide more efficient communications with low latency [7], [8]. FDD also dominates current cellular systems, and for 5G wireless networks, the radio capability for FDD remains in the specifications [9]. Therefore, it is also important to consider CSIT acquisition for FDD systems. In fact, there are situations when it is necessary to consider downlink channel estimation even in TDD systems. For example, due to random radio-frequency (RF) circuit mismatches in the uplink and downlink and limited coherence time, the channel reciprocity may no longer hold [10], [11]. Moreover, in LTE/5G systems, there exist situations when users only use some of the antennas to transmit in the uplink. In this case, the channel associated with the other user antennas has to be estimated via downlink training. In addition, the cell-edge users may suffer from very low SNR in the channel estimation phase due to the limited power budget at the mobile device. In this case, it is preferable to use downlink channel estimation because the BS can transmit pilot signals at a larger power to meet the channel estimation accuracy.

Many works have shown that the effective dimension of a massive MIMO channel is actually much less than its original dimension because of the limited local scattering effect in the propagation environment [12]–[15]. Specifically, the massive MIMO channel has an approximately sparse representation under the discrete Fourier transform (DFT) basis if the BS is equipped with a large uniform linear array (ULA) [13], [16]–[18]. As a consequence, a large number of compressive sensing (CS) algorithms that exploit the hidden sparsity under the DFT basis have been proposed for downlink channel estimation and feedback [8], [12], [13], [15], [19]–[23]. Nevertheless, there are at least two challenges of the DFT-based methods: 1) they are only applicable to ULAs because the sparse property hinges strongly on the shared structure between the DFT basis and the ULA steering; and 2) they always suffer from inevitable modeling error caused by direction mismatch. To alleviate the modeling error, a denser sampling grid covering the angular domain

with more points (named overcomplete DFT basis) was considered in [24]–[26]. However, the overcomplete DFT method is still applicable to ULAs only, and it may lead to a high performance loss if the grid is not sufficiently dense.

Recently, the sparse Bayesian learning (SBL) method has attracted significant attention for sparse signal recovery [27]–[32]. The SBL-based framework has an inherent learning capability, and hence, no prior knowledge about the sparsity level, noise variance or direction mismatch is required. Moreover, theoretical and empirical results have shown that SBL methods can achieve better performance than the l_1-norm-based methods [27], [30]. Our previous work [31] introduced an off-grid SBL-based method for downlink channel estimation, which can be applied to arbitrary 2D-array geometry and substantially reduces the modeling error caused by direction mismatch. The method in [31] overcomes all the aforementioned challenges of the DFT-based methods, and simulation results illustrated that it can achieve much better channel estimation performance than the existing state-of-the-art methods. However, [31] only focused on single-user channel estimation in massive MIMO systems.

Many studies have observed that channels of multi-user massive MIMO systems may share common sparsity structures due to the commonly shared local scattering clusters [14], [33]. To exploit the common sparsity among nearby users, a joint orthogonal matching pursuit recovery algorithm was proposed in [13]. However, the effectiveness of that approach relies on appropriate user clustering in the multi-user MIMO network. While there are various user clustering methods [7], [34], [35] in the literature, they are targeted for different purposes. It is also worth noting that the meaning of group sparsity from the perspective of compressed sensing (CS) [36], [37] is different from the one used in this paper. In CS, group sparsity is usually known as block sparsity, which means the locations of significant coefficients cluster in blocks under a known specific sorting order. To the best of our knowledge, user clustering for maximizing the common sparsity has not been investigated before. In this paper, we propose an efficient off-grid SBL-based approach for joint channel estimation and user grouping to enhance the effectiveness of common sparsity in massive MIMO systems. The following summarizes the contributions of this paper.

- **General Sparsity Model for User Grouping**

  We develop a more general sparsity model to better capture the group sparse structure in practical multi-user massive MIMO systems. In the literature, a commonly used group sparsity model assumes that users in each group share a uniform sparsity pattern [38]. This oversimplified model can simplify the procedure for user grouping; however, it usually fails to hold, even for physically close users, in practice. Outliers deviated from the uniform sparsity pattern in each group may significantly degrade the effectiveness of the common sparsity, and bring limited (or negative) gain for channel estimation. To address this issue, we propose a general model having two sparsity components: a commonly shared sparsity and an individual sparsity. Since the additional individual sparsity can account for any outliers, the new
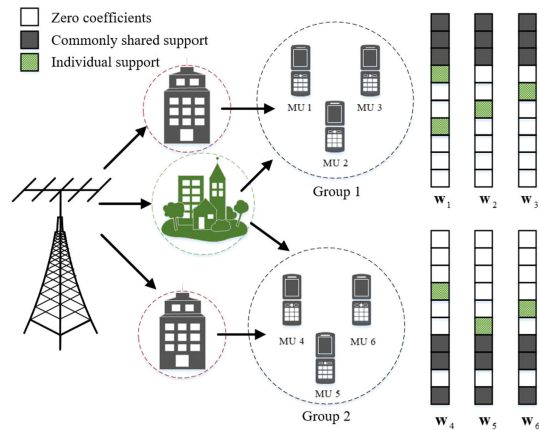

Fig. 1. Illustration of system model and general sparsity mode, where the commonly shared support sets for two groups are $\mathcal{S}_1 = \{1, 2, 3\}$ and $\mathcal{S}_2 = \{7, 8, 10\}$, respectively.

model may capture a more complex and realistic group sparse structure in real-world applications (see Fig. 1 for example).

- **SBL-based Framework for Joint Channel Estimation and User Grouping**

  We propose a novel SBL-based method to autonomously partition users into groups during the channel estimation under the general sparsity model. SBL-based methods have been widely applied to estimate the sparse channel in single-user massive MIMO systems, but they are not applicable to joint user grouping and channel estimation in multi-user massive MIMO systems. To the best of our knowledge, the method proposed for wideband direction-of-arrival estimation in [38] is the only candidate that may be tailored to solve the problem of joint channel estimation and user grouping. However, it requires the aforementioned restrictive assumption that users in each group share a uniform sparsity structure. To handle the more practical general sparsity model, we propose a novel SBL-based framework, which can fully exploit the common sparsity among nearby users and exclude the harmful effect from outliers simultaneously. Moreover, the grid-refining procedure used in [31] is also extended to the framework to efficiently combat direction mismatch with an arbitrary 2D-array geometry.

The rest of the paper is organized as follows. In Section II, we present the system model and the general sparsity model. In Section III, we provide the SBL-based method for joint channel estimation and user grouping. In Section IV, we extend the proposed method to handling direction mismatch with an arbitrary 2D-array geometry. Numerical experiments and a conclusion follow in Sections V and VI, respectively.

Notations: $\mathbb{C}$ denotes complex number, $\| \cdot \|_p$ denotes p-norm, $(\cdot)^T$ denotes transpose, $(\cdot)^H$ denotes Hermitian transpose, $(\cdot)^\dagger$ denotes pseudoinverse, $\mathbf{I}$ denotes identity matrix, $\mathbf{A}_\Omega$ denotes the sub-matrix formed by collecting the columns from Ω, $\mathcal{CN}(\cdot | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes complex Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$, $\text{supp}(\cdot)$ denotes the set of indices of nonzero

elements, $\text{tr}(\cdot)$ denotes trace operator, $\text{diag}(\cdot)$ denotes diagonal operator, and $\text{Re}(\cdot)$ denotes real part operator.

II. DATA MODEL

A. Massive MIMO Channel Model

Consider a massive MIMO system as illustrated in Fig. 1. There is one BS with N ($\gg 1$) antennas and K mobile users (MUs) with a single antenna. Assume that the BS broadcasts a sequence of T training pilot symbols, denoted by $\mathbf{X} \in \mathbb{C}^{T \times N}$, for each MU to estimate the downlink channel. Then, the downlink received signal $\mathbf{y}_k \in \mathbb{C}^{T \times 1}$ at the k-th MU is given by

$$\mathbf{y}_k = \mathbf{X}\mathbf{h}_k + \mathbf{n}_k, \tag{1}$$

where $\mathbf{h}_k \in \mathbb{C}^{N \times 1}$ stands for the downlink channel vector from the BS to the k-th MU, $\mathbf{n}_k \in \mathbb{C}^{T \times 1}$ stands for the additive complex Gaussian noise with each element being zero mean and variance σ^2 in the downlink, and $\text{tr}(\mathbf{X}\mathbf{X}^H) = PTN$, with P/σ^2 measuring the training signal-to-noise ratio (SNR). If the BS is equipped with a linear array, $\mathbf{h}_k$ can be formulated as [39]–[41]

$$\mathbf{h}_k = \sum_{c=1}^{N_c} \sum_{s=1}^{N_s} \xi_{c,s}^k \mathbf{a}(\theta_{c,s}^k), \tag{2}$$

where N_c stands for the number of scattering clusters, N_s stands for the number of sub-paths per scattering cluster, $\xi_{c,s}^k$ is the complex gain of the s-th sub-path in the c-th scattering cluster for the k-th MU, and $\theta_{c,s}^k$ is the corresponding azimuth angle-of-departure (AoD). For a linear array, the steering vector $\mathbf{a}(\theta) \in \mathbb{C}^{N \times 1}$ is in the form of

$$\mathbf{a}(\theta) = [1, e^{-j2\pi \frac{d_2}{\lambda} \sin(\theta)}, \ldots, e^{-j2\pi \frac{d_N}{\lambda} \sin(\theta)}]^T, \tag{3}$$

where λ is the wavelength of the downlink propagation, and d_n stands for the distance between the n-th antenna and the first antenna. For a ULA, $\mathbf{a}(\theta)$ can be simplified by

$$\mathbf{a}(\theta) = [1, e^{-j2\pi \frac{d}{\lambda} \sin(\theta)}, \ldots, e^{-j2\pi \frac{(N-1)d}{\lambda} \sin(\theta)}]^T, \tag{4}$$

where d stands for the distance between adjacent sensors.

For ease of notation, we denote the true AoDs for MU k as $\{\theta_l^k, l = 1, 2, \ldots, L\}$ with $L = N_c N_s$. Let $\hat{\boldsymbol{\vartheta}} = \{\hat{\vartheta}_l\}_{l=1}^{\hat{L}}$ be a fixed sampling grid that uniformly covers the angular domain $[-\frac{\pi}{2}, \frac{\pi}{2}]$, where $\hat{L}$ denotes the number of grid points. If the grid is fine enough, such that all the true AoDs θ_l^ks, $l = 1, 2, \ldots, L$, lie on (or practically close to) the grid, we have[1]

$$\mathbf{h}_k = \mathbf{A}\mathbf{w}_k, \tag{5}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{a}(\hat{\vartheta}_1), \ \mathbf{a}(\hat{\vartheta}_2), \ \ldots, \ \mathbf{a}(\hat{\vartheta}_{\hat{L}}) \end{bmatrix} \in \mathbb{C}^{N \times \hat{L}}$, and $\mathbf{w}_k \in \mathbb{C}^{\hat{L} \times 1}$ is a vector with a few non-zero elements corresponding to the true directions at $\{\theta_l, l = 1, 2, \ldots, L\}$. With (1) and (5), $\mathbf{y}_k$ can be rewritten by

$$\mathbf{y}_k = \mathbf{X}\mathbf{A}\mathbf{w}_k + \mathbf{n}_k = \boldsymbol{\Phi}\mathbf{w}_k + \mathbf{n}_k, \tag{6}$$

[1]The DFT basis becomes a special case of $\mathbf{A}$ if the BS is equipped with a ULA and there are N grid points such that $\{\sin \hat{\vartheta}_l\}_{l=1}^{\hat{L}}$ uniformly covers the range $[-1, 1]$.

where $\boldsymbol{\Phi} \triangleq \mathbf{X}\mathbf{A}$. Note that the assumption that all true AoDs are located on the predefined spatial grid is not always valid in practice [32], [42]. We will address the direction mismatch in Section IV, as well as the extension for arbitrary 2D-array geometry.

B. General Sparsity Model

The massive MIMO channel usually has the following two important properties:

- (*Sparsity Property*): Due to the limited local scattering effect in the propagation environment [12], [17], [18], the number of scattering clusters N_c is usually small and the sub-paths associated with each scattering cluster are likely to concentrate in a small range. In other words, only a few angles are occupied in the angular domain, which, in return, brings a sparse representation $\mathbf{w}_k$.
- (*Group Property*): Some users may face a very similar scattering structure if they are physically close to each other [13], [14], [33]. Hence, the performance of the downlink channel estimation can be improved if we can exploit the common sparsity among nearby users, because it can bring additional useful information for sparse signal recovery algorithms.

Without loss of generality, assume that the K users can be partitioned into G groups $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_G\}$. The commonly used group sparsity model assumes that users in each group share a uniform sparsity pattern [38], i.e.,

$$\text{supp}(\mathbf{w}_p) = \text{supp}(\mathbf{w}_q), \quad p, q \in \mathcal{G}_g. \tag{7}$$

This assumption can simplify the procedure for user grouping, but usually fails to hold in practice, because it is a restrictive constraint requiring the same scattering structure for users in each group. The channel estimation performance may be significantly degraded by the outliers deviated from the uniform sparsity pattern in each group.

To capture a more complex and realistic group sparse structure, we partition the sparse representation vector into two parts, i.e.,

$$\mathbf{w}_k = \mathbf{w}_k^s + \mathbf{w}_k^v, \tag{8}$$

where $\mathbf{w}_k^s$ stands for the commonly shared sparse representation vector whose support corresponds to the commonly shared support, and $\mathbf{w}_k^v$ stands for the individual sparse representation vector whose support corresponds to the individual support.

Definition 1 (General Sparsity Model): Let the sparse representation vector be formulated as in (8), and there be multiple groups, with each group having a distinct commonly shared sparsity pattern; i.e.,

$$\mathcal{S}_g = \text{supp}(\mathbf{w}_p^s) = \text{supp}(\mathbf{w}_q^s), \quad \forall p, q \in \mathcal{G}_g, \tag{9}$$

$$\mathcal{S}_{\hat{g}} \neq \mathcal{S}_{\check{g}}, \quad \forall \hat{g} \neq \check{g}, \tag{10}$$

where $\mathcal{S}_g$ stands for the commonly shared support set for the g-th group.

From Definition 1, it is worth noting that 1) $\mathbf{w}_k^s$ indicates which group the k-th MU belongs to; and 2) the additional individual sparse representation vector $\mathbf{w}_k^v$ accounts for any outliers.

Clearly, the general sparsity model is more reasonable in practical implementations. It includes the commonly shared sparsity as a special case with $\mathbf{w}_k^v = 0$. Moreover, the outlier problem in the scenario of a single group has been addressed in [13], where the named joint sparsity model used can also be included as a special case of ours with $G = 1$. Fig. 1 shows an example of the general sparsity model where the commonly shared support sets for two groups are $\mathcal{S}_1 = \{1, 2, 3\}$ and $\mathcal{S}_2 = \{7, 8, 10\}$, respectively. Specifically, if $\mathbf{w}_1 = [\times, \times, \times, \times, 0, 0, \times, 0, 0, 0]^T$, with "$\times$" standing for a significant value, its corresponding general sparsity pattern is $\mathbf{w}_1^s = [\times, \times, \times, 0, 0, 0, 0, 0, 0, 0]^T$ and $\mathbf{w}_1^v = [0, 0, 0, \times, 0, 0, \times, 0, 0, 0]^T$.

The aim of this paper is to automatically partition the users into G groups according to their hidden commonly shared supports (determined by $\mathbf{w}_k^s$), and simultaneously obtain the channel estimation for each user. This expected to obtain more accurate channel estimation performance because we exploit additional information about common sparsity among nearby users, as captured by (9). The main challenge introduced by the general sparsity model is that it is difficult to directly extract the commonly shared component $\mathbf{w}_k^s$ from $\mathbf{w}_k$ because $\mathbf{w}_k^s$ and $\mathbf{w}_k^v$ are mixed. To the best of our knowledge, there lacks an efficient method for simultaneously handling $\mathbf{w}_k^s$ and $\mathbf{w}_k^v$. In the presence of a uniform sparsity model (i.e., $\mathbf{w}_k^v = 0, \forall k$), the proposed method for wideband DOA estimation in [38] is the only candidate that may be tailored to solve the problem of joint channel estimation and user grouping. However, it does not apply to the more practical general sparsity model. To overcome the challenge, in the next section, we propose a novel SBL-based framework which can fully exploit the common sparsity among nearby users and exclude the harmful effect from outliers simultaneously.

Remark 2: Although we set the number of groups G to a fixed value in the general sparsity model, this fixed value is not required to be the real number of user groups $G^\star$. When G is chosen to be larger than $G^\star$, the algorithm will automatically cluster users into $G^\star$ groups and assign zero users to the remaining $G - G^\star$ groups. Therefore, the number of groups can be "optimized" by the proposed algorithm in an implicit way. When G is smaller than the optimal $G^\star$, the outliers deviated from the uniform sparsity pattern can be mitigated by the general model. As a result, the channel estimation performance of the proposed algorithm is not sensitive to the choice of G (which will be verified in the simulations).

III. JOINT CHANNEL ESTIMATION AND USER GROUPING

In this section, we propose an efficient SBL-based method for joint channel estimation and user grouping with the general sparsity model. For ease of exposition, we proceed as follows. We begin by introducing the SBL formulation for group sparse signal recovery. Then, we resort to the variational Bayesian inference (VBI) methodology [43] and adopt an alternating optimization algorithm to perform the Bayesian inference, so as to jointly cluster the users and estimate the channel. Note that the modeling error caused by direction mismatch will be addressed in the next section.

A. Sparse Bayesian Learning Formulation

In order to separate the commonly shared support and the individual support for the k-th MU, we use (8) to rewrite the received signal $\mathbf{y}_k$ as

$$\mathbf{y}_k = \mathbf{\Phi}(\mathbf{w}_k^s + \mathbf{w}_k^v) + \mathbf{n}_k = \bar{\bar{\mathbf{\Phi}}} \bar{\mathbf{w}}_k + \mathbf{n}_k, \quad (11)$$

where $\bar{\bar{\mathbf{\Phi}}} = [\mathbf{\Phi}, \mathbf{\Phi}]$ and $\bar{\mathbf{w}}_k = [(\mathbf{w}_k^s)^T, (\mathbf{w}_k^v)^T]^T$. Following the classical sparse Bayesian model [28], we model $\mathbf{w}_k^s$ and $\mathbf{w}_k^v$ associated with user k in group g as non-stationary Gaussian prior distributions:

$$p(\mathbf{w}_k^s | \boldsymbol{\gamma}_g^*) = \mathcal{CN}(\mathbf{w}_k^s | \mathbf{0}, \mathrm{diag}\,(\boldsymbol{\gamma}_g^*)^{-1}), \forall k \in \mathcal{G}_g \quad (12)$$

and

$$p(\mathbf{w}_k^v | \boldsymbol{\gamma}_k^v) = \mathcal{CN}(\mathbf{w}_k^v | \mathbf{0}, \rho \cdot \mathrm{diag}\,(\boldsymbol{\gamma}_k^v)^{-1}), \quad (13)$$

where ρ is a small positive constant (whose function will be explained later), $\boldsymbol{\gamma}_g^* = [\gamma_{g,1}^*, \gamma_{g,2}^*, \ldots, \gamma_{g,\hat{L}}^*]^T$, $\boldsymbol{\gamma}_k^v = [\gamma_{k,1}^v, \gamma_{k,2}^v, \ldots, \gamma_{k,\hat{L}}^v]^T$, and $\gamma_{g,l}^*$ and $\gamma_{k,l}^v$ stand for the precision of the l-th elements of $\mathbf{w}_k^s$ and $\mathbf{w}_k^v$, respectively. Note that all users in group g share a common precision vector $\boldsymbol{\gamma}_g^*$ for the common sparse vector $\mathbf{w}_k^s$, which captures the common sparsity shared by the users. On the other hand, different users in group g have different precision vectors $\boldsymbol{\gamma}_k^s$ for the individual sparse vector $\mathbf{w}_k^v$, which captures the individual sparsity caused by the outliers deviated from the uniform sparsity pattern. For a given sparse vector $\mathbf{w}_k$, there are multiple ways to partition $\mathbf{w}_k$ into a common sparse vector $\mathbf{w}_k^s$ and an individual sparse vector $\mathbf{w}_k^v$, where each partition corresponds to one user grouping result. Clearly, a user grouping result is only meaningful when the users in the same user group share a large common support; i.e., we favor a denser $\mathbf{w}_k^s$ over $\mathbf{w}_k^v$. Hence, we introduce a small positive constant $\rho \in (0, 1)$ in (13) to provide a sparser prior for $\mathbf{w}_k^v$ than for $\mathbf{w}_k^s$. Empirical evidence shows that the performance of our method is not sensitive to the choice of ρ, as long as ρ is sufficiently small. In the simulations, we set $\rho = 0.001$.

To force the clustering of $\mathbf{w}_k^s$s with G groups, we introduce $\mathbf{z}_k$ of size $G \times 1$ as the assignment vector for the k-th MU. Specifically, if the k-th MU belongs to the g-th group (i.e., $k \in \mathcal{G}_g$), $\mathbf{z}_k$ is a zero vector, except for the g-th element being one. Then, the distribution of $\mathbf{w}_k^s$ conditional on $\mathbf{z}_k$ and $\boldsymbol{\gamma}_g^*$s can be expressed as

$$p(\mathbf{w}_k^s | \mathbf{z}_k, \mathbf{\Gamma}^*) = \prod_{g=1}^{G} \left\{ \mathcal{CN}(\mathbf{w}_k^s | \mathbf{0}, \mathrm{diag}(\boldsymbol{\gamma}_g^*)^{-1} \right\}^{z_{k,g}}, \quad (14)$$

where $\mathbf{\Gamma}^* = \{\boldsymbol{\gamma}_g^*\}_{g=1}^{G}$, and $z_{k,g}$ stands for the g-th element of $\mathbf{z}_k$.

For tractable inference of $\boldsymbol{\gamma}_g^*$s and $\boldsymbol{\gamma}_k^v$s, the elements of $\boldsymbol{\gamma}_g^*$ and $\boldsymbol{\gamma}_k^v$ (denoted by $\gamma_{g,l}^*$ and $\gamma_{k,l}^v$, $l = 1, 2, \ldots, \hat{L}$) are modeled as independent Gamma distributions, i.e.,

$$p(\boldsymbol{\gamma}_g^*) = \prod_{l=1}^{\hat{L}} \Gamma(\gamma_{g,l}^* | a, b) \quad (15)$$

and

$$p(\boldsymbol{\gamma}_k^v) = \prod_{l=1}^{\hat{L}} \Gamma(\gamma_{k,l}^v | a, b), \qquad (16)$$

where a and b are some small constants (e.g., $a = b = 0.0001$). Gamma distribution is a conjugate prior of Gaussian distribution, and the two-stage hierarchical prior provided by (14) and (15) [or (13) and (16)] for $\mathbf{w}_k^s$ (or $\mathbf{w}_k^v$) is recognized as encouraging sparsity, due to the heavy tails and sharp peak at zero [27], [28]. In fact, it can be shown that finding a MAP estimate of $\mathbf{w}_k^v$ (or $\mathbf{w}_k^s$) with the two-stage hierarchical prior is equivalent to finding the minimum l_0-norm solution using FOCUSS with $p \to 0$ [44], where p corresponds to the parameter of l_p-norm. It is worth noting that the precisions $\gamma_{g,l}^*$s in (12) [or $\gamma_{k,l}^v$s in (13)] directly indicate the support of $\mathbf{w}_k^s$ (or $\mathbf{w}_k^v$). For example, if $\gamma_{g,l}^*$ is large, the l-th element of $\mathbf{w}_k^s$ tends to zero; otherwise, the value of the l-th element is significant.

Under the assumption of circular symmetric complex Gaussian noise, we have

$$p(\mathbf{y}_k | \mathbf{w}_k^s, \mathbf{w}_k^v, \alpha) = \mathcal{CN}(\mathbf{y}_k | \boldsymbol{\Phi}\mathbf{w}_k, \alpha^{-1}\mathbf{I}), \qquad (17)$$

where $\alpha = \sigma^{-2}$ stands for the noise precision. Since α is usually unknown, we similarly model it as a gamma hyperprior $p(\alpha) = \Gamma(\alpha | a, b)$.

Let $\boldsymbol{\Theta} = \{\alpha, \bar{\mathbf{W}}, \boldsymbol{\Gamma}^*, \boldsymbol{\Gamma}^v, \mathbf{Z}, \}$ be the set of hidden variables to be estimated, where $\bar{\mathbf{W}} = \{\bar{\mathbf{w}}_k\}_{k=1}^K$, $\boldsymbol{\Gamma}^v = \{\boldsymbol{\gamma}_k^v\}_{k=1}^K$, and $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^K$. The user groups and channel estimation can be jointly obtained if we can calculate the maximum a *posteriori* (MAP) optimal estimate of $p(\boldsymbol{\Theta}|\mathbf{Y})$, where $\mathbf{Y} = \{\mathbf{y}_k\}_{k=1}^K$. Specifically, the user group is indicated by the MAP estimator of the group assignment vector $\mathbf{z}_k$, and the angular domain channel vector $\mathbf{w}_k$ can be calculated from the MAP estimator of $\bar{\mathbf{w}}_k$ according to (8). Unfortunately, this MAP estimate is intractable. Therefore, in the next subsection, we will resort to the VBI methodology and will adopt an alternating optimization algorithm to infer the hidden variables iteratively.

B. Overview of the Proposed Method

The principle behind VBI is to find an approximate posterior of $\boldsymbol{\Theta}$ (denoted by $q(\boldsymbol{\Theta})$), instead of the exact posterior, to make the MAP estimate tractable, where $q(\boldsymbol{\Theta})$ is assumed to be factorized approximately as

$$q(\boldsymbol{\Theta}) = q(\alpha) \underbrace{\prod_{k=1}^K q(\bar{\mathbf{w}}_k)}_{\triangleq q(\bar{\mathbf{W}})} \underbrace{\prod_{g=1}^G q(\boldsymbol{\gamma}_g^*)}_{\triangleq q(\boldsymbol{\Gamma}^*)} \underbrace{\prod_{k=1}^K q(\boldsymbol{\gamma}_k^v)}_{\triangleq q(\boldsymbol{\Gamma}^v)} \underbrace{\prod_{k=1}^K q(\mathbf{z}_k)}_{\triangleq q(\mathbf{Z})}, \quad (18)$$

and it should be chosen to minimize the Kullback-Leibler (KL) divergence with respect to (w.r.t.) the true posterior:

$$D_{KL}(q(\boldsymbol{\Theta}) || p(\boldsymbol{\Theta}|\mathbf{Y})) = -\int q(\boldsymbol{\Theta}) \ln \frac{p(\boldsymbol{\Theta}|\mathbf{Y})}{q(\boldsymbol{\Theta})} d\boldsymbol{\Theta}. \qquad (19)$$

In other words, the corresponding optimization problem to find the "best" approximate posterior under the factorized constraint

in (18) can be formulated as

$$q^\star(\boldsymbol{\Theta}) = \arg \max_{q(\boldsymbol{\Theta})} \underbrace{\int q(\boldsymbol{\Theta}) \ln \frac{p(\mathbf{Y}, \boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} d\boldsymbol{\Theta}}_{\triangleq \mathcal{U}(q_1, q_2, q_3, q_4, q_5)}, \qquad (20)$$

where q_i denotes $q(\Theta_i)$ for simplicity, and Θ_i stands for the i-th element in $\boldsymbol{\Theta}$. Since the above objective is a high-dimensional non-convex function, it is difficult to find the optimal solution. Here, we adopt an alternating optimization algorithm to find a stationary solution instead. Specifically, we update q_is as

$$q_1^{(i+1)} = \arg \max_{q_1} \mathcal{U}(q_1, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)}), \qquad (21)$$

$$q_2^{(i+1)} = \arg \max_{q_2} \mathcal{U}(q_1^{(i+1)}, q_2, q_3^{(i)}, q_4^{(i)}, q_5^{(i)}), \qquad (22)$$

$$q_3^{(i+1)} = \arg \max_{q_3} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3, q_4^{(i)}, q_5^{(i)}), \qquad (23)$$

$$q_4^{(i+1)} = \arg \max_{q_4} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4, q_5^{(i)}), \qquad (24)$$

$$q_5^{(i+1)} = \arg \max_{q_5} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i+1)}, q_5), \qquad (25)$$

where $(\cdot)^{(i)}$ stands for the i-th iteration. Once the algorithm converges, we can obtain the approximate posteriors: $q(\alpha)$, $q(\bar{\mathbf{w}}_k)$s, $q(\boldsymbol{\gamma}_g^*)$s, $q(\boldsymbol{\gamma}_k^v)$s and $q(\mathbf{z}_k)$s. Let

$$\hat{\boldsymbol{\phi}}_k = \langle \mathbf{z}_k \rangle_{q(\mathbf{z}_k)} \qquad (26)$$

and

$$\bar{\boldsymbol{\mu}}_k = \langle \bar{\mathbf{w}}_k \rangle_{q(\bar{\mathbf{w}}_k)}, \qquad (27)$$

where $\langle \cdot \rangle_{p(x)}$ stands for the expectation operator w.r.t. $p(x)$. Then, we are able to cluster the users into G groups; e.g., user k belongs to group $g_k^\star$ if

$$g_k^\star = \arg \max_g \hat{\phi}_{k,g}, \qquad (28)$$

where $\hat{\phi}_{k,g}$ stands for the g-th element of $\hat{\boldsymbol{\phi}}_k$. Recall that $\mathbf{w}_k = \mathbf{w}_k^s + \mathbf{w}_k^v$ and $\bar{\mathbf{w}}_k = [(\mathbf{w}_k^s)^T, (\mathbf{w}_k^v)^T]^T$. Therefore, we have

$$\boldsymbol{\mu}_k \triangleq \langle \mathbf{w}_k \rangle_{q(\bar{\mathbf{w}}_k)} = \bar{\boldsymbol{\mu}}_{k,1} + \bar{\boldsymbol{\mu}}_{k,2}, \qquad (29)$$

where $\bar{\boldsymbol{\mu}}_{k,1}$ and $\bar{\boldsymbol{\mu}}_{k,2}$ stand for the first and last $\hat{L}$ elements of $\bar{\boldsymbol{\mu}}_k$, respectively. Letting $\Omega_k = \text{supp}(\boldsymbol{\mu}_k)$, the estimated downlink channels $\mathbf{h}_k^e$s can be calculated by

$$\mathbf{h}_k^e = \mathbf{A}_{\Omega_k} (\boldsymbol{\Phi}_{\Omega_k})^\dagger \mathbf{y}_k. \qquad (30)$$

The overall flow of the proposed algorithm is given in Fig. 2. In the following subsections, we will illustrate how to solve the optimization problems (21)–(25) in detail (Section III-C) and then give a convergence analysis of the alternating optimization algorithm (Section III-D).

C. Detailed Implementations

In this subsection, we focus on handling the optimization problems (21)–(25), whose solutions will be presented in Lemmas 3–7 as follows. It is worth noting that some initializations are required to trigger the iterations, which will be addressed later.
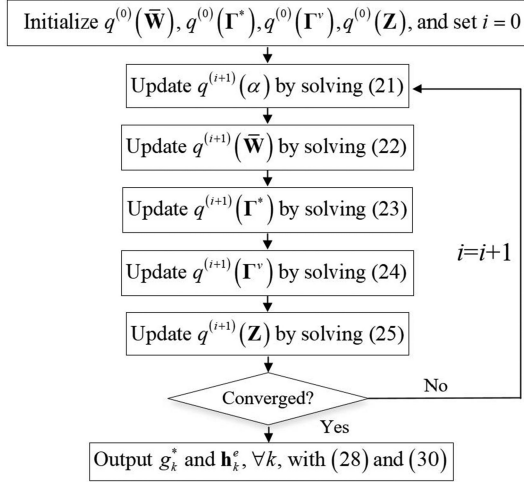
Fig. 2. The overall flow of the proposed algorithm.

1) Update for q_1: We update q_1 [or $q(\alpha)$] by solving the optimization problem (21), whose solution follows a gamma distribution.

Lemma 3: The optimization problem (21) has a unique solution:

$$q^{(i+1)}(\alpha) = \Gamma(\alpha|a_\alpha^{(i)}, b_\alpha^{(i)}), \tag{31}$$

where $a_\alpha^{(i)} = (a + KT)$ and $b_\alpha^{(i)} = b + \sum_{k=1}^K (\|\mathbf{y}_k - \mathbf{\Phi}\boldsymbol{\mu}_k^{(i)}\|_2^2 + \mathrm{tr}(\mathbf{\Phi}\mathbf{\Sigma}_k^{(i)}\mathbf{\Phi}^H))$, with $\boldsymbol{\mu}_k^{(i)}$ and $\mathbf{\Sigma}_k^{(i)}$ being the mean and variance of $\mathbf{w}_k$ at the i-th iteration [whose closed-from expressions will be given later, in (34) and (35)].

Proof: See Appendix-A. ∎

Note that the mean of α, w.r.t. the gamma distribution defined in (31), can be calculated as a byproduct:

$$\hat{\alpha}^{(i+1)} \triangleq \langle \alpha \rangle_{q^{(i+1)}(\alpha)} = \frac{a_\alpha^{(i)}}{b_\alpha^{(i)}}, \tag{32}$$

which will be used in the next lemma.

2) Update for q_2: We update q_2 [or $q(\bar{\mathbf{W}})$] by solving the optimization problem (22), whose solution follows a Gaussian distribution.

Lemma 4: The optimization problem (22) has a unique solution:

$$q^{(i+1)}(\bar{\mathbf{W}}) = \prod_{k=1}^K q^{(i+1)}(\bar{\mathbf{w}}_k) = \prod_{k=1}^K \mathcal{CN}(\bar{\mathbf{w}}_k|\bar{\boldsymbol{\mu}}_k^{(i+1)}, \bar{\mathbf{\Sigma}}_k^{(i+1)}), \tag{33}$$

where $\bar{\boldsymbol{\mu}}_k^{(i+1)} = \hat{\alpha}^{(i+1)}\bar{\mathbf{\Sigma}}_k^{(i+1)}\bar{\mathbf{\Phi}}^H \mathbf{y}_k$ and $\bar{\mathbf{\Sigma}}_k^{(i+1)} = (\hat{\alpha}^{(i+1)} \bar{\mathbf{\Phi}}^H \bar{\mathbf{\Phi}} + \mathrm{diag}([(\hat{\boldsymbol{\gamma}}_k^s)^{(i)}; \rho^{-1}(\hat{\boldsymbol{\gamma}}_k^v)^{(i)}]))^{-1}$ with $(\hat{\boldsymbol{\gamma}}_k^s)^{(i)}$ and $(\hat{\boldsymbol{\gamma}}_k^v)^{(i)}$ being the means of $\boldsymbol{\gamma}_k^s$ and $\boldsymbol{\gamma}_k^v$ at the i-th iteration (whose closed-from expressions will be given later, in (39) and (41), respectively).

Proof: See Appendix-B. ∎

Since $\mathbf{w}_k = \mathbf{w}_k^s + \mathbf{w}_k^v$, we have

$$\boldsymbol{\mu}_k^{(i+1)} \triangleq \langle \mathbf{w}_k \rangle_{q^{(i+1)}(\bar{\mathbf{w}}_k)} = \bar{\boldsymbol{\mu}}_{k,1}^{(i+1)} + \bar{\boldsymbol{\mu}}_{k,2}^{(i+1)} \tag{34}$$

and

$$\mathbf{\Sigma}_k^{(i+1)} \triangleq \left\langle (\mathbf{w}_k - \boldsymbol{\mu}_k^{(i+1)})(\mathbf{w}_k - \boldsymbol{\mu}_k^{(i+1)})^H \right\rangle_{q^{(i+1)}(\bar{\mathbf{w}}_k)}$$
$$= \bar{\mathbf{\Sigma}}_{k,1}^{(i+1)} + \bar{\mathbf{\Sigma}}_{k,2}^{(i+1)} + \bar{\mathbf{\Sigma}}_{k,3}^{(i+1)} + \bar{\mathbf{\Sigma}}_{k,4}^{(i+1)}, \tag{35}$$

where $\bar{\boldsymbol{\mu}}_{k,1}^{(i+1)} = \bar{\boldsymbol{\mu}}_k^{(i+1)}(1:\hat{L})$, $\bar{\boldsymbol{\mu}}_{k,2}^{(i+1)} = \bar{\boldsymbol{\mu}}_k^{(i+1)}(\hat{L}+1:2\hat{L})$, $\bar{\mathbf{\Sigma}}_{k,1}^{(i+1)} = \bar{\mathbf{\Sigma}}_k^{(i+1)}(1:\hat{L}, 1:\hat{L})$, $\bar{\mathbf{\Sigma}}_{k,2}^{(i+1)} = \bar{\mathbf{\Sigma}}_k^{(i+1)}(\hat{L}+1:2\hat{L}, \hat{L}+1:2\hat{L})$, $\bar{\mathbf{\Sigma}}_{k,3}^{(i+1)} = \bar{\mathbf{\Sigma}}_k^{(i+1)}(1:\hat{L}, \hat{L}+1:2\hat{L})$, and $\bar{\mathbf{\Sigma}}_{k,4}^{(i+1)} = \bar{\mathbf{\Sigma}}_k^{(i+1)}(\hat{L}+1:2\hat{L}, 1:\hat{L})$. Note that these byproducts will be required for updating both q_3 and q_4.

3) Update for q_3: We update q_3 [or $q(\mathbf{\Gamma}^*)$] by solving the optimization problem (23), whose solution follows a gamma distribution.

Lemma 5: The optimization problem (23) has a unique solution:

$$q^{(i+1)}(\mathbf{\Gamma}^*) = \prod_{g=1}^G \prod_{l=1}^{\hat{L}} q^{(i+1)}(\gamma_{g,l}^*)$$
$$= \prod_{g=1}^G \prod_{l=1}^{\hat{L}} \Gamma\left(\gamma_{g,l}^*|(a_{g,l}^*)^{(i+1)}, (b_{g,l}^*)^{(i+1)}\right), \tag{36}$$

where $(a_{g,l}^*)^{(i+1)} = a + \sum_{k=1}^K \hat{\phi}_{k,g}^{(i)}$, $(b_{g,l}^*)^{(i+1)} = b + \sum_{k=1}^K \hat{\phi}_{k,g}^{(i)}(|\bar{\mu}_{k,1,l}^{(i+1)}|^2 + \bar{\Sigma}_{k,1,l}^{(i+1)})$, $\hat{\phi}_{k,g}^{(i)} = q^{(i)}(z_{k,g} = 1)$ [whose closed-from expression will be given later, in (43)], $\bar{\mu}_{k,1,l}^{(i+1)}$ stands for the l-th element of $\bar{\boldsymbol{\mu}}_{k,1}^{(i+1)}$, and $\bar{\Sigma}_{k,1,l}^{(i+1)}$ stands for the l-th diagonal element of $\bar{\mathbf{\Sigma}}_{k,1}^{(i+1)}$.

Proof: See Appendix-C. ∎

Then, the mean of $\gamma_{g,l}^*$ at the $(i+1)$-th iteration is

$$(\hat{\gamma}_{g,l}^*)^{(i+1)} \triangleq \langle \gamma_{g,l}^* \rangle_{q^{(i+1)}(\gamma_{g,l}^*)} = \frac{(a_{g,l}^*)^{(i+1)}}{\left((b_{g,l}^*)^{(i+1)}\right)}, \tag{37}$$

and the logarithmic expectation is

$$(\widehat{\ln \gamma_{g,l}^*})^{(i+1)} \triangleq \langle \ln \gamma_{g,l}^* \rangle_{q^{(i+1)}(\gamma_{g,l}^*)}$$
$$= \Psi\left((a_{g,l}^*)^{(i+1)}\right) - \ln\left((b_{g,l}^*)^{(i+1)}\right), \tag{38}$$

where $\Psi(\cdot)$ stands for the digamma function. We define $\gamma_{k,l}^s = \sum_{g=1}^G (z_{k,g}\gamma_{g,l}^*)$, and then the mean of $\gamma_{k,l}^s$ at the $(i+1)$-th iteration is

$$(\hat{\gamma}_{k,l}^s)^{(i+1)} = \langle \gamma_{k,l}^s \rangle_{q^{(i+1)}(\mathbf{\Gamma}^*)q^{(i)}(\mathbf{z}_k)} = \sum_{g=1}^G \hat{\phi}_{k,g}^{(i)}(\hat{\gamma}_{g,l}^*)^{(i+1)}. \tag{39}$$

Note that (39) will be required for updating $q(\mathbf{Z})$.

4) Update for q_4: We update q_4 [or $q(\mathbf{\Gamma}^v)$] by solving the optimization problem (24), whose solution also follows a gamma distribution.

Lemma 6: The optimization problem (24) has a unique solution:

$$q^{(i+1)}(\mathbf{\Gamma}^v) = \prod_{k=1}^{K} \prod_{l=1}^{\hat{L}} q^{(i+1)}(\gamma_{k,l}^v)$$

$$= \prod_{k=1}^{K} \prod_{l=1}^{\hat{L}} \Gamma\left(\gamma_{k,l}^v | (a_{k,l}^v)^{(i+1)}, (b_{k,l}^v)^{(i+1)}\right), \quad (40)$$

where $(a_{k,l}^v)^{(i+1)} = a + K$ and $(b_{k,l}^v)^{(i+1)} = b + \rho^{-1}(|\bar{\mu}_{k,2,l}^{(i+1)}|^2 + \bar{\Sigma}_{k,2,l}^{(i+1)})$.

Proof: The proof is similar to Lemma 3. So it is omitted for brevity. ∎

With (40), the mean of $\gamma_{k,l}^v$ at the $(i+1)$-th iteration is

$$(\hat{\gamma}_{k,l}^v)^{(i+1)} \triangleq \langle \gamma_{k,l}^v \rangle_{q^{(i+1)}(\gamma_{k,l}^v)} = \frac{(a_{k,l}^v)^{(i+1)}}{\left((b_{k,l}^v)^{(i+1)}\right)}, \quad (41)$$

which was required for updating q_2.

5) Update for q_5: We update q_5 [or $q(\mathbf{Z})$] by solving the optimization problem (25), whose solution is characterized by the following lemma.

Lemma 7: The optimization problem (25) has a unique solution:

$$q^{(i+1)}(\mathbf{Z}) = \prod_{k=1}^{K} q^{(i+1)}(\mathbf{z}_k) = \prod_{k=1}^{K} \prod_{g=1}^{G} \left(\hat{\phi}_{k,g}^{(i+1)}\right)^{z_{k,g}}, \quad (42)$$

where

$$\hat{\phi}_{k,g}^{(i+1)} = q^{(i+1)}(z_{k,g} = 1) = \frac{\exp(\varsigma_{k,g}^{(i+1)})}{\sum_{g=1}^{G} \exp(\varsigma_{k,g}^{(i+1)})} \quad (43)$$

and

$$\varsigma_{k,g}^{(i+1)} = \sum_{l=1}^{\hat{L}} (\widehat{\ln \gamma_{g,l}^*})^{(i+1)} - \sum_{l=1}^{\hat{L}} (\gamma_{g,l}^*)^{(i+1)} \left(|\bar{\mu}_{k,1,l}^{(i+1)}|^2 + \bar{\Sigma}_{k,1,l}^{(i+1)}\right). \quad (44)$$

Proof: See Appendix-D. ∎

The proposed algorithm proceeds by repeated application of (31), (33), (36), (40) and (42), and its convergence will be addressed in the next subsection. The main computational burden of the proposed method is given as follows.

- The most cost for updating $q(\alpha)$ is to calculate b_α, whose computational complexity is $\mathcal{O}(T\hat{L}^2K)$ per iteration.
- Calculating $\bar{\Sigma}_k$s and $\bar{\mu}_k$s in each iteration for updating $q(\bar{\mathbf{W}})$ is $\mathcal{O}(T\hat{L}^2K)$ and $\mathcal{O}(\hat{L}^2K)$, respectively.
- The complexities in updating $q(\mathbf{\Gamma}^*)$ and $q(\mathbf{\Gamma}^v)$ in each iteration are $\mathcal{O}(G\hat{L}K)$ and $\mathcal{O}(\hat{L}K)$, respectively.
- The complexity in updating $q(\mathbf{Z})$ is $\mathcal{O}(G\hat{L})K$ per iteration.

This suggests the total computational requirement of the proposed method is $\mathcal{O}(T\hat{L}^2K)$ per iteration.

Following are some practical implementation tips for the proposed method. In order to trigger the alternating optimization algorithm, we need some initializations for $q^{(0)}(\bar{\mathbf{W}})$, $q^{(0)}(\mathbf{\Gamma}^*)$, $q^{(0)}(\mathbf{\Gamma}^v)$ and $q^{(0)}(\mathbf{Z})$. According to the main results in Lemmas 4–7, we can simply set the initializations as follows:

- initialize a Gaussian distribution function $q^{(0)}(\bar{\mathbf{W}})$, with parameters $\bar{\mu}_k^{(i+1)} = \bar{\Sigma}_k^{(0)} \bar{\mathbf{\Phi}}^H \mathbf{y}_k$ and $\bar{\Sigma}_k^{(0)} = (\bar{\mathbf{\Phi}}^H \bar{\mathbf{\Phi}} + \text{diag}([\mathbf{1}_{\hat{L}\times 1}; \rho^{-1}\mathbf{1}_{\hat{L}\times 1}]))^{-1}$;
- initialize a gamma distribution function $q^{(0)}(\mathbf{\Gamma}^*)$, with parameters $(a_{g,l}^*)^{(0)} = (b_{g,l}^*)^{(0)} = 1, \forall g, l$;
- initialize a gamma distribution function $q^{(0)}(\mathbf{\Gamma}^v)$, with parameters $(a_{k,l}^v)^{(0)} = (b_{k,l}^v)^{(0)} = 1, \forall k, l$;
- initialize $q^{(0)}(\mathbf{Z})$, with $\varsigma_{k,g}^{(0)}$s being uniformly chosen from $[0, 1]$.

Empirical evidence shows that the proposed method remains very robust to these initializations. Moreover, we set $a = b = 0.0001$ in the simulations.

D. Convergence Analysis and Discussion

The non-decreasing property of the sequence $\mathcal{U}(q_1^{(i)}, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)})$, $i = 1, 2, 3, \ldots$, is well guaranteed by the update rules (21)–(25).

Lemma 8: The update rules (21)–(25) give a non-decreasing sequence $\mathcal{U}(q_1^{(i)}, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)})$, $i = 1, 2, 3, \ldots$.

Proof: See Appendix-E. ∎

Together with the fact that the objective function $\mathcal{U}(q_1, q_2, q_3, q_4, q_5)$ has an upper bound of 1,[2] the sequence $\mathcal{U}(q_1^{(i)}, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)})$, $i = 1, 2, 3, \ldots$, converges to a limit. The alternating algorithm does not converge to a stationary solution in general. However, the specific conditions satisfied by our problem make it possible to prove the convergence of the alternating algorithm to a stationary point. The alternating optimization algorithm can be viewed as a special case of the block MM algorithm. Hence, we have the following lemma:

Lemma 9: The iterates generated by the alternating optimization algorithm converge to a stationary solution of the optimization problem (20).

Proof: See Appendix-F. ∎

Finally, we discuss the relationship between our method and the method in [38]:

- Recall that the general sparsity model used in our method includes the commonly shared sparsity model used in [38] as a special case of $\mathbf{w}_k^v = \mathbf{0}, \forall k$. Thereore, our method designed for the general sparsity model is more general than the method in [38]. It can also handle the commonly shared sparsity model, by simply ignoring the updates for $\mathbf{w}_k^v$s and γ_k^vs.
- Our method performs Bayesian inference for the hidden variables from a new perspective of alternating optimization. Compared with the traditional Bayesian inference used in [38], our method has the following advantages: 1) its convergence is more easily proved (see Lemma 8); 2) it reveals that the convergence solution is also a stationary solution (see Lemma 9), which is a stronger convergence result since the traditional method only establishes the convergence of objective values to a certain point, without proving the converged solution is a stationary solution;

---

[2]This is because of $\int q(\mathbf{\Theta}) \ln \frac{p(\mathbf{Y}, \mathbf{\Theta})}{q(\mathbf{\Theta})} d\mathbf{\Theta} \leq \ln \int q(\mathbf{\Theta}) \frac{p(\mathbf{Y}, \mathbf{\Theta})}{q(\mathbf{\Theta})} d\mathbf{\Theta} = \ln p(\mathbf{Y})$, where the first inequality follows Jensen's inequality.

and 3) it provides a flexible framework to handle the problem of direction mismatch (see Section IV).

- In our method, each $\mathbf{z}_k$ is treated as a simple assignment vector without a prior distribution, and the number of groups G is assumed to be known, while in [38], each $\mathbf{z}_k$ is treated as a random vector that is generated from a Dirichlet process prior, and the number of groups G is automatically determined. It is worth noting that extending our method with the Dirichlet process (DP) prior and the automatically determined G is straightforward. Even without such extending, empirical results (also refer to the simulations) show that our method is still applicable to an unknown G. This is because the adopted general model can capture a much more general group sparse structure and can provide a robust result for an inexact choice of G (Remark 2). The simulation results also show that there is no performance loss by removing the DP prior.

Another motivation for choosing a fixed G comes from that fact that the user grouping result with a fixed G can be applied to some practical applications in massive MIMO systems. For example, we may combine the proposed method with Joint Spatial Division and Multiplexing (JSDM) [34], [35], where a fixed G is required. It is worth noting that we do not try to provide an improved JSDM framework, but only replace the user grouping algorithm used in JSDM with ours. This application is just a byproduct of our method. Compared with the traditional user grouping method, our method can bring some significant advantages: 1) it does not require prior knowledge about the channel covariance, where the acquisition of channel covariance may pose great challenges because it requires collecting a large number of channel samples in practical implementations; and 2) it can give a better user grouping result in the sense of Bayesian optimality, so as to alleviate the interference across different groups and enhance the sum-rate performance of JSDM systems.

IV. HANDLING DIRECTION MISMATCH WITH ARBITRARY 2D-ARRAY GEOMETRY

In the section, we extend the proposed method to handling direction mismatch with an arbitrary 2D-array geometry. Note that the steering vector $\mathbf{a}(\theta, \phi)$ for an arbitrary 2D-array geometry contains both azimuth angle θ and elevation angle ϕ [31], [45]:

$$\mathbf{a}(\theta, \phi) = [1, e^{-j2\pi \frac{d_2}{\lambda} \cos(\phi) \sin(\theta - \psi_2)},$$

$$\dots, e^{-j2\pi \frac{d_N}{\lambda} \cos(\phi) \sin(\theta - \psi_N)}]^T, \qquad (45)$$

where (d_n, ψ_n) is the coordinates of the n-th sensor. Following the convention in Section III, we adopt a fixed sampling grid $\hat{\boldsymbol{\vartheta}} = \{\hat{\vartheta}_l\}_{l=1}^{\hat{L}}$ to uniformly cover the azimuth domain $[-\pi, \pi]$. Recall that the direction mismatch between the true AoD and the grid point is unavoidable because signals usually come from random directions in practice. Here, we adopt the off-grid model proposed in [31] to handle the direction mismatch. Let θ_l^k and ϕ_l^k denote the l-th true azimuth and elevation AoDs of the k-th MU, repectively. If $\theta_l^k \notin \{\hat{\vartheta}_i\}_{i=1}^{\hat{L}}$ and $\hat{\vartheta}_{n_l}, n_l \in \{1, 2, \dots, \hat{L}\}$,

is the nearest grid point to θ_l^k, we write θ_l^k as

$$\theta_l^k = \hat{\vartheta}_{n_l} + \beta_{k,n_l}, \qquad (46)$$

where β_{k,n_l} corresponds to the azimuth direction mismatch (or off-grid gap). With (46), the received signal $\mathbf{y}_k$ can be rewritten by

$$\mathbf{y}_k = \boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)(\mathbf{w}_k^s + \mathbf{w}_k^v) + \mathbf{n}_k, \qquad (47)$$

where $\boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) = \mathbf{X}\mathbf{A}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)$, $\boldsymbol{\beta}_k = [\beta_{k,1}, \beta_{k,2}, \dots, \beta_{k,\hat{L}}]^T$, $\boldsymbol{\varphi}_k = [\varphi_{k,1}, \varphi_{k,2}, \dots, \varphi_{k,\hat{L}}]^T$, $\mathbf{A}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) = [\mathbf{a}(\hat{\vartheta}_1 + \beta_{k,1}, \varphi_{k,1}), \mathbf{a}(\hat{\vartheta}_2 + \beta_{k,2}, \varphi_{k,2}), \dots, \mathbf{a}(\hat{\vartheta}_{\hat{L}} + \beta_{k,\hat{L}}, \varphi_{k,\hat{L}})]$,

$$\beta_{k,n_l} = \begin{cases} \theta_l^k - \hat{\vartheta}_{n_l}, & l = 1, 2, \dots, L \\ 0, & \text{otherwise} \end{cases}, \quad \text{and} \quad \varphi_{k,n_l} =$$

$$\begin{cases} \phi_l^k, & l = 1, 2, \dots, L \\ 0, & \text{otherwise} \end{cases}.$$ Note that φ_{k,n_l} corresponds to elevation direction mismatch. Due to introducing the term of the off-grid gap, the direction mismatch can be significantly alleviated. Another advantage is that the commonly shared support among nearby users does not need to coincide strictly with each other. For example, let $\hat{L} = 180$ and the azimuth AoDs of two nearby MUs be $\{8.1°, 10.2°, 11.9°, 15.3°\}$ and $\{7.4°, 10.3°, 12.3°, 15.1°\}$, respectively. In this case, the nearest grid points for the first supports of the two MUs are different. However, an appropriate choice of off-grid gap can fix the commonly shared support mismatch, e.g., $8.1° = 8° + 0.1°$ and $7.4° = 8° - 0.6°$.

In the sparse Bayesian learning formulation for the off-grid model (47), almost all the results in Section III-B remain unchanged, except that (17) is replaced by

$$p(\mathbf{y}_k | \mathbf{w}_k^s, \mathbf{w}_k^v, \alpha, \boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) = \mathcal{CN}(\mathbf{y}_k | \boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)\mathbf{w}_k, \alpha^{-1}\mathbf{I}) \qquad (48)$$

and the optimization problem (20) is modified by

$$\{q^\star(\boldsymbol{\Theta}), \mathbf{B}^\star\} = \arg \max_{q(\boldsymbol{\Theta}), \mathbf{B}} \mathcal{U}(\boldsymbol{\Theta}, \mathbf{B}), \qquad (49)$$

where $\mathbf{B} = \{\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k\}_{k=1}^K$ is treated as a unknown parameter, rather than a random variable. Similarly, in the $(i + 1)$-th iteration, we update q_is and $\mathbf{B}$ as

$$q_1^{(i+1)} = \arg \max_{q_1} \mathcal{U}(q_1, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)}, \mathbf{B}^{(i)}), \qquad (50)$$

$$q_2^{(i+1)} = \arg \max_{q_2} \mathcal{U}(q_1^{(i+1)}, q_2, q_3^{(i)}, q_4^{(i)}, q_5^{(i)}, \mathbf{B}^{(i)}), \qquad (51)$$

$$q_3^{(i+1)} = \arg \max_{q_3} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3, q_4^{(i)}, q_5^{(i)}, \mathbf{B}^{(i)}), \qquad (52)$$

$$q_4^{(i+1)} = \arg \max_{q_4} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4, q_5^{(i)}, \mathbf{B}^{(i)}), \qquad (53)$$

$$q_5^{(i+1)} = \arg \max_{q_5} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i+1)}, q_5, \mathbf{B}^{(i)}), \qquad (54)$$

$$\mathbf{B}^{(i+1)} = \arg \max_{\mathbf{B}} \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i+1)}, q_5^{(i+1)}, \mathbf{B}). \qquad (55)$$

Applying the results in Section III-C, we can obtain the solutions to (50)–(54) directly, where the only difference is in replacing $\mathbf{\Phi}$ with $\mathbf{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)$.

What remains is to obtain the update for $\mathbf{B}$. However, the last maximization problem (55) is non-convex and it is difficult to find its optimal solution. Alternatively, we apply gradient update on the objective function of (55) and obtain a simple one-step update for each $\boldsymbol{\beta}_k$ and $\boldsymbol{\varphi}_k$ as in [31]. As shown in Appendix G, the derivative of the objective function, w.r.t. $\boldsymbol{\beta}_k$, can be calculated as

$$\boldsymbol{\zeta}_k^{(i+1)} = [\zeta^{(i+1)}(\beta_{k,1}), \zeta^{(i+1)}(\beta_{k,2}), \ldots, \zeta^{(i+1)}(\beta_{k,\hat{L}})]^T, \quad (56)$$

with

$$\zeta^{(i+1)}(\beta_{k,l})$$
$$= 2\mathrm{Re}\left((\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}^{(i)}))^H \mathbf{X}^H \mathbf{X}\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}^{(i)})\right)$$
$$\cdot c_{k1}^{(i+1)} + 2\mathrm{Re}\left((\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}^{(i)}))^H \mathbf{X}^H \mathbf{c}_{k2}^{(i+1)}\right), \quad (57)$$

where $c_{k1}^{(i+1)} = -\hat{\alpha}^{(i+1)}(\chi_{k,ll}^{(i+1)} + |\mu_{k,l}^{(i+1)}|^2)$, $\mathbf{c}_{k2}^{(i+1)} = \hat{\alpha}^{(i+1)}$ $((\mu_{k,l}^{(i+1)})^* \mathbf{y}_{k-l}^{(i+1)} - \mathbf{X}\sum_{j \neq l} \chi_{k,jl}^{(i+1)}\mathbf{a}(\hat{\vartheta}_j + \beta_{k,j}^{(i)}, \varphi_{k,j}^{(i)}))$, $\mathbf{y}_{k-l}^{(i+1)} = \mathbf{y}_k - \mathbf{X} \cdot \sum_{j \neq l}(\mu_{k,j}^{(i+1)} \cdot \mathbf{a}(\hat{\vartheta}_j + \beta_{k,j}^{(i)}, \varphi_{k,j}^{(i)}))$, $\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l})$ $= d\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l})/d\beta_{k,l}$, and $\mu_{k,l}^{(i+1)}$ and $\chi_{k,jl}^{(i+1)}$ denote the l-th element and the (j, l)-th element of $\boldsymbol{\mu}_k^{(i+1)}$ and $\boldsymbol{\Sigma}_k^{(i+1)}$, respectively. With (56), we are able to update the value of $\boldsymbol{\beta}_k$ in the derivative direction, i.e.,

$$\boldsymbol{\beta}_k^{(i+1)} = \boldsymbol{\beta}_k^{(i)} + \Delta_k \cdot \boldsymbol{\zeta}_k^{(i+1)}, \quad (58)$$

where Δ_k is the stepsize that can be optimized by backtracking line search [46]. As mentioned in Section III-D of [31], choosing the right stepsize can be time-consuming. To reduce the computational complexity, we use a fixed stepsize to update $\boldsymbol{\beta}_k$:

$$\boldsymbol{\beta}_k^{(i+1)} = \boldsymbol{\beta}_k^{(i)} + \frac{r_\theta}{100} \cdot \mathrm{sign}(\boldsymbol{\zeta}_k^{(i+1)}), \quad (59)$$

where $r_\theta = \pi/\hat{L}$ stands for the grid interval, and $\mathrm{sign}(\cdot)$ stands for the signum function.

Following similar procedures to these in Appendix VI-G, we can obtain the derivative of the objective function w.r.t $\boldsymbol{\varphi}_k$ as

$$\boldsymbol{\varsigma}_k^{(i+1)} = [\varsigma^{(i+1)}(\varphi_{k,1}), \varsigma^{(i+1)}(\varphi_{k,2}), \ldots, \varsigma^{(i+1)}(\varphi_{k,\hat{L}})]^T, \quad (60)$$

with

$$\varsigma^{(i+1)}(\varphi_{k,l})$$
$$= 2\mathrm{Re}\left((\mathbf{a}'_\varphi(\hat{\vartheta}_l + \beta_{k,l}^{(i)}, \varphi_{k,l}))^H \mathbf{X}^H \mathbf{X}\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}^{(i)}, \varphi_{k,l})\right)$$
$$\cdot c_{k1}^{(i+1)} + 2\mathrm{Re}\left((\mathbf{a}'_\varphi(\hat{\vartheta}_l + \beta_{k,l}^{(i)}, \varphi_{k,l}))^H \mathbf{X}^H \mathbf{c}_{k2}^{(i+1)}\right), \quad (61)$$

where $\mathbf{a}'_\varphi(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}) = d\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l})/d\varphi_{k,l}$. With (60), we can update $\boldsymbol{\varphi}_k$ similarly to (58). As mentioned in [31], the elevation angle ranges from $-\pi/2$ to $\pi/2$, but it is sufficient to assume that $\varphi_{k,l}$ ranges from 0 to $\pi/2$, because the steering vector contains $\cos\varphi_{k,l}$ only. Therefore, we initialize each $\varphi_{k,l}$ uniformly from $[0, \pi/2]$, and use a fixed stepsize to update $\boldsymbol{\varphi}_k$

[similarly to (59)]:

$$\boldsymbol{\varphi}_k^{(i+1)} = \boldsymbol{\varphi}_k^{(i)} + \frac{\pi}{36} \cdot \max\left\{(\varrho)^i, 0.001\right\} \cdot \mathrm{sign}(\boldsymbol{\varsigma}_k^{(i+1)}), \quad (62)$$

where $0.9474 < \varrho < 1$ is a constant [31].

Once the algorithm converges, the estimated downlink channels $\mathbf{h}_k^e$s can be calculated as

$$\mathbf{h}_k^e = \mathbf{A}_{\Omega_k}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)(\mathbf{\Phi}_{\Omega_k}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k))^\dagger \mathbf{y}_k. \quad (63)$$

V. SIMULATION RESULTS

In this section, numerical simulations are conducted to evaluate the performance of our proposed method. The proposed method is compared with the following baselines:

- **Baseline 1** (Individual-DFT): Each $\mathbf{h}_k$ is individually recovered using the l_1-norm minimization algorithm [47], [48] with a DFT basis.
- **Baseline 2** (Individual-SBL): Each $\mathbf{h}_k$ is individually recovered using the standard SBL method [28] with a DFT basis.
- **Baseline 3** (Individual-off-grid): Each $\mathbf{h}_k$ is individually recovered using the off-grid SBL method [31].
- **Baseline 4** (Joint-OMP): $\mathbf{h}_k$s are jointly recovered using the joint orthogonal matching pursuit recovery method [13].
- **Baseline 5** (Common-SBL): $\mathbf{h}_k$s are jointly recovered using the multiple measurement SBL method [32] with an off-grid basis, where $\mathbf{h}_k$s are assumed to share a uniform sparsity structure.
- **Baseline 6** (Group-SBL): $\mathbf{h}_k$s are jointly recovered using the group SBL method [38] with an off-grid basis.[3]

We first focus on simulations for ULAs, where we use the 3GPP spatial channel model (SCM) [41] to generate the channels for an urban microcell. The downlink frequency is 2170 MHz and the inter-antenna spacing is $d = c/(2f_0)$, with c being the light speed and $f_0 = 2000$ MHz. Then, we run simulations with the 3GPP 3D channel model [49], which provides a 2D array model. All the parameters of the 3D channel model follow 3D-UMa-NOLS (see Table 7.3-6 in [49]). The normalized mean square error (NMSE) is defined as

$$\frac{1}{M_c}\sum_{m=1}^{M_c}\frac{\sum_{k=1}^K \|\tilde{\mathbf{h}}_k^m - \mathbf{h}_k^m\|_2^2}{\sum_{k=1}^K \|\mathbf{h}_k^m\|_2^2}, \quad (64)$$

where $\mathbf{h}_k^m$ is the downlink channel vector for the k-th MU at the m-th Monte Carlo trial, $\tilde{\mathbf{h}}_k^m$ is the estimate of $\mathbf{h}_k^m$, and $M_c = 200$ is the number of Monte Carlo trials. Unless otherwise specified, in the following, we assume that every channel realization consists of N_c random scattering clusters, each cluster contains $N_s = 20$ sub-paths concentrated in a $\mathcal{A} = 10°$ angular spread, and the number of grid points is fixed at $\hat{L} = N$. Note that MATLAB codes have been made available online at https://sites.google.com/site/jsdaiustc/publication.

---

[3]For fairness, the off-grid refinement method used in [38] is replaced by the one used in ours.
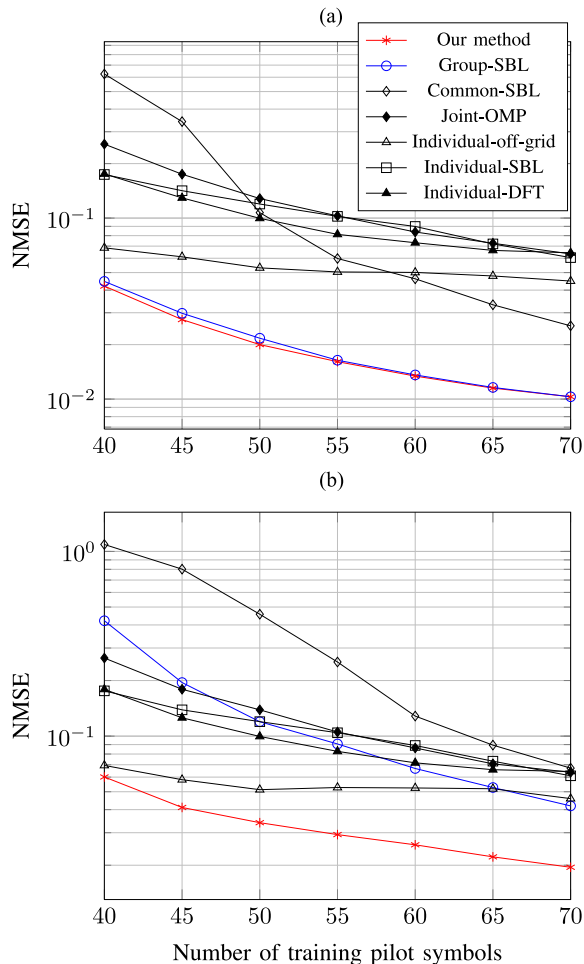
Fig. 3. NMSE of downlink channel estimate versus the number of training pilot symbols for ULA, where $N = 80$, $K = 60$, $G = 3$ and SNR $= 0$ dB. a) $L_s = 4$ and $L_v = 0$; b) $L_s = 2$ and $L_v = 2$.
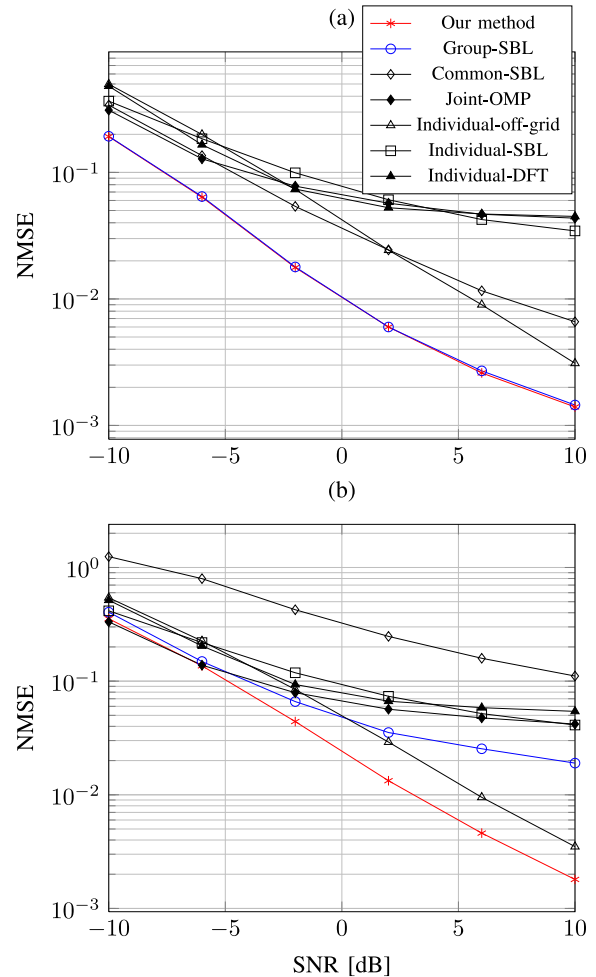

Fig. 4. NMSE of downlink channel estimate versus SNR for ULA, where $N = 80$, $G = 4$, $K = 50$ and $T = 60$. a) $L_s = 3$ and $L_v = 0$; b) $L_s = 2$ and $L_v = 1$.

A. Channel Estimation Performance Versus T for ULA

In Fig. 3, Monte Carlo trials are carried out to investigate the impact of the number of pilot symbols on the channel estimation performance for ULA. Assume that a ULA is equipped at the BS with $N = 80$ antennas and the system supports $K = 60$ MUs. The MUs are randomly dropped into three groups with a uniform distribution. The number of shared (unshared) scattering clusters for users in the same group is denoted by L_s (L_v).[4] If $L_s = N_c$, it means that users in the same group have a uniform scattering structure, while if $L_s = 0$, there is no group property for users. The center AoD of each scattering clusters ranges from $-90°$ to $90°$ uniformly. The training pilots are randomly generated, and the SNR is chosen as 0 dB. Fig. 3 shows the NMSE performance of the downlink channel estimate achieved by the different channel estimation strategies versus the number of training pilot symbols T. All the results are obtained by averaging over 200 Monte Carlo channel realizations. It can be seen that 1) the NMSEs of all the methods decrease as the number of training pilot symbols increases; 2) compared with the

individual recovery methods (Individual-DFT, Individual-SBL, Individual-off-grid), Joint-OMP and Common-SBL, our method and Group-SBL can improve the NMSE performance due to exploiting the common sparsity among nearby users; 3) when the uniform shared sparsity assumption holds true for each group ($L_s = 4$ and $L_v = 0$), our method and Group-SBL achieve similar channel estimation performance (Fig. 3-a), which verifies that removing the DP prior in our method does not bring any performance loss; and 4) when the uniform sparsity assumption fails to hold ($L_s = 2$ and $L_v = 2$), our method outperforms Group-SBL because our method can handle outliers but Group-SBL is only designed for the uniform sparsity assumption.

B. Channel Estimation Performance Versus SNR for ULA

In Fig. 4, we study the impact of SNR on the channel estimation performance for ULA. We consider the same scenario as in Section V-A, except that the number of training pilot symbols is fixed at 60 and the number of users is set to 50. Fig. 4 shows the NMSE performance of the downlink channel estimate achieved by the different channel estimation strategies versus SNR. All the results are obtained by averaging over
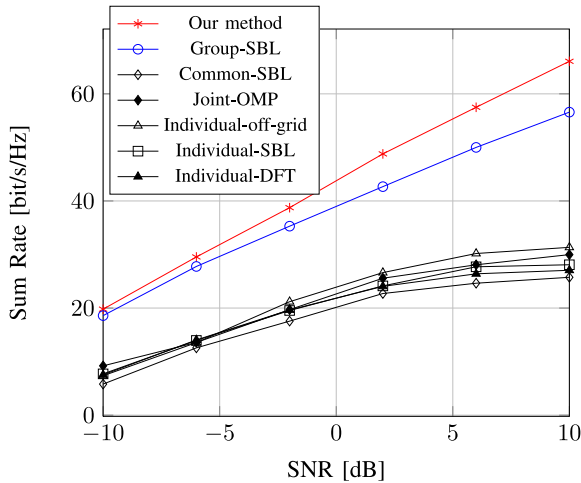
---

[4]Note that $L_s + L_v = N_c$, and two users sharing a scattering cluster means that the AoD mean of the scattering cluster is the same.

Fig. 5. Sum spectral efficiency versus SNR for JSDM, where $N = 80, G = 4$, $K = 100, T = 60, L_s = 2$ and $L_v = 1$.

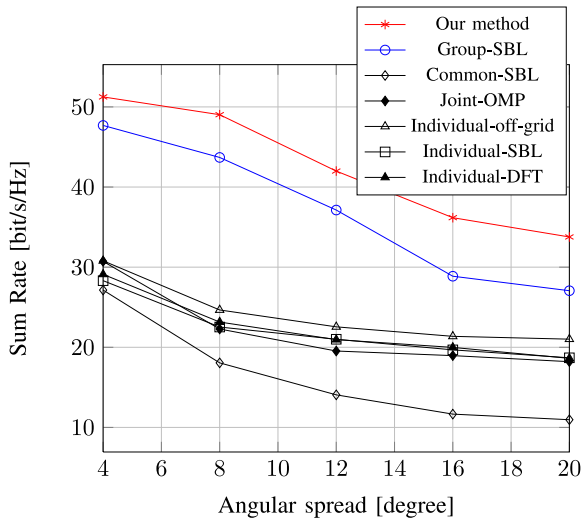Fig. 7. NMSE of downlink channel estimate versus inexact G, where $G^\star = 4$, $N = 100, K = 50, T = 60, L_s = 2, L_v = 1$, and SNR $= 0$ dB.

with a uniform distribution, and 20% of MUs will be activated in the system. Following [35], the standard K-Means algorithm is chosen to cluster users if the method (e.g., Individual-DFT, Individual-SBL, Individual-off-grid, Joint-OMP and Common-SBL) cannot provide knowledge of the user grouping, and then zero-forcing beamforming (ZFBF) with semi-orthogonal user selection (SUS) is adopted for each group in the JSDM framework. Fig. 5 shows the sum spectral efficiency achieved by the different strategies versus SNR, and Fig. 6 shows the sum spectral efficiency achieved by the different strategies versus angular spread. All the results are obtained by averaging over 200 Monte Carlo channel realizations. Compared with other methods, ours can significantly improve the sum-rate performance of JSDM systems. This is because our method can give better channel estimation and user grouping results in the sense of Bayesian optimality, so as to alleviate the interference across different groups.

Fig. 6. Sum spectral efficiency versus the angular spread for JSDM, where $N = 80, G = 4, K = 100, L_s = 2, L_v = 1, T = 50$ and SNR $= 0$ dB.

D. Channel Estimation Performance Versus Inexact G

In Fig. 7, we illustrate that our method applies to unknown real number of user groups $G^\star$. Assume that a ULA is equipped at the BS with $N = 100$ antennas and the system supports $K = 50$ MUs. The number of training pilot symbols is fixed at 60, $L_s = 2$ and $L_v = 1$. The MUs are randomly dropped into four groups with a uniform distribution, but the real number of user groups is not exactly known. Fig. 7 shows the NMSE performance of the downlink channel estimate achieved by the different channel estimation strategies versus an inexact G. It is interesting to see that most curves in the figure remain unchanged. The reason that NMSEs do not change much for the individual methods (Individual-DFT, Individual-SBL, Individual-off-grid) is because each $\mathbf{h}_k$ is estimated individually for each user, and thus its estimation performance is not related to G or $G^\star$, while Common-SBL and Joint-OMP always assume that there is just one group ($G = 1$). The reason why the NMSE of our method also does not change much is because the adopted general model

200 Monte Carlo channel realizations. It is shown that 1) the NMSEs of all the methods decrease as SNR increases; 2) when the uniform shared sparsity assumption holds true, our method and Group-off-grid achieve very similar channel estimation performance (Fig. 4-a); 3) when the uniform sparsity assumption fails to hold, Group-off-grid gives very bad performance because of outliers deviated from the group sparsity patterns (Fig. 4-b); and 4) the proposed general sparsity model can capture the true group sparse structure, and our method indeed works for the general sparsity model and can significantly improve the channel estimation performance.

C. Comparison of Sum Spectral Efficiency for JSDM

In Figs. 5 and 6, we study the sum spectral efficiency when the proposed method is integrated into JSDM. Assume that a ULA is equipped at BS with $N = 80$ antennas and the system supports $K = 100$ MUs. The MUs are randomly dropped into four groups
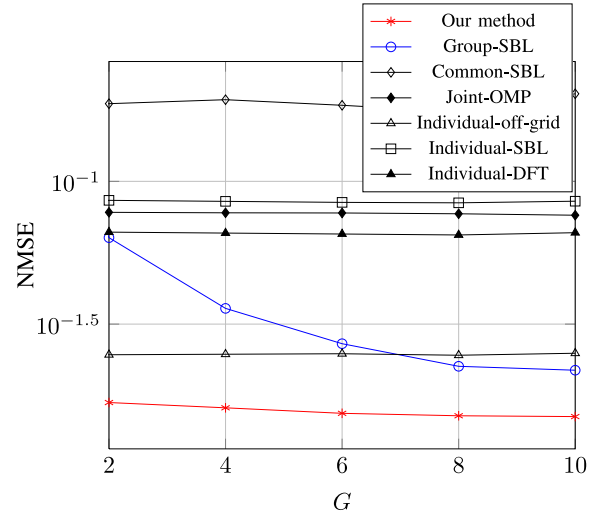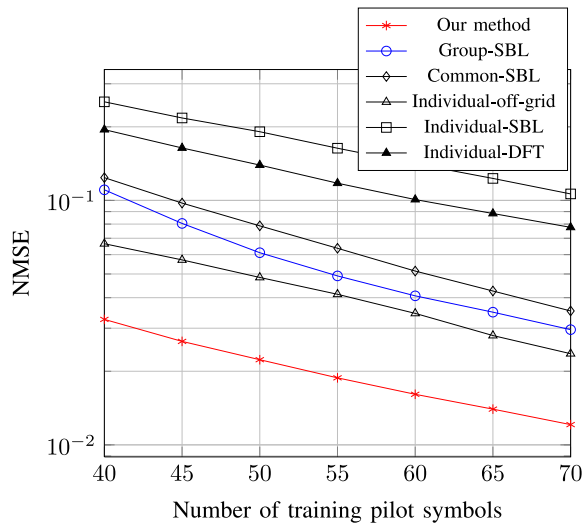
Fig. 8. NMSE of downlink channel estimate versus T for 2D array, where $N = 10 \times 10$, $G = 4$, $K = 30$, SNR = 0 dB, $L_s = 2$ and $L_v = 1$.
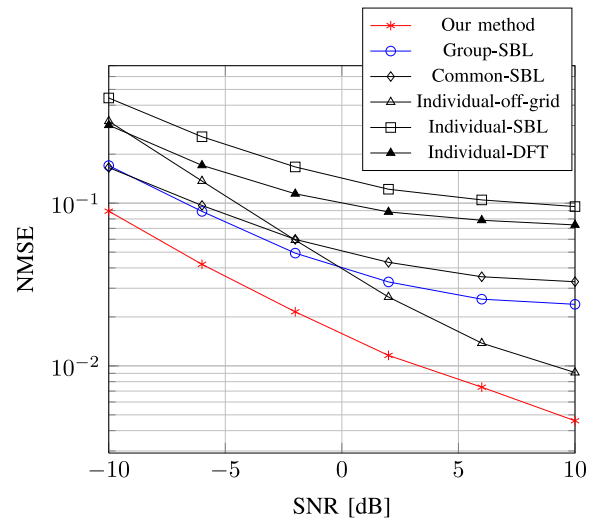

Fig. 10. NMSE of downlink channel estimate versus SNR for 2D array, where $N = 10 \times 10$, $G = 4$, $K = 30$, $T = 60$, $L_s = 2$ and $L_v = 1$.
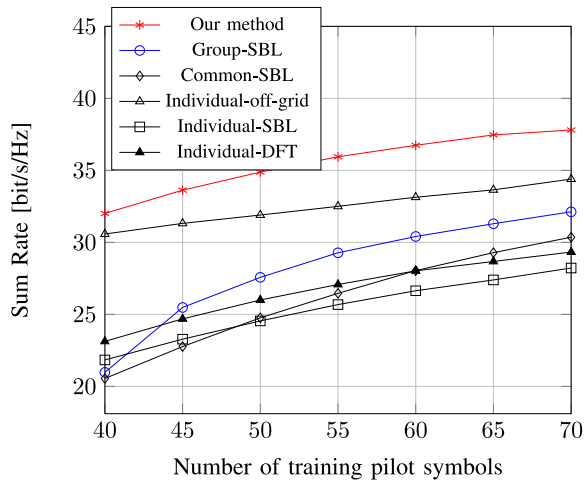

Fig. 9. Sum spectral efficiency versus T for 2D array, where $N = 10 \times 10$, $G = 4$, $K = 30$, SNR = 0 dB, $L_s = 2$ and $L_v = 1$.
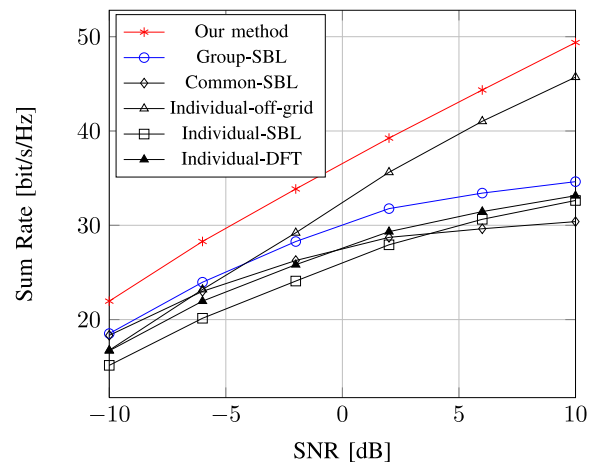

Fig. 11. Sum spectral efficiency versus SNR for 2D array, where $N = 10 \times 10$, $G = 4$, $K = 30$, $T = 60$, $L_s = 2$ and $L_v = 1$.

can capture a much more general group sparse structure and can provide a robust result for an inexact choice of G, as long as G is not much smaller than the true value $G^\star$. Hence, Fig. 7 verifies that our method works well for an unknown G.

E. Channel Estimation and Sum Rate Performance With 2D Array

In Figs. 8–11, Monte Carlo trials are carried out to investigate the channel estimation and sum rate performance with the 2D array. Assume that the 2D planar array at the BS is equipped with 10×10 antennas, where both the horizontal and vertical inter-antenna spacings are a half wavelength. Every channel realization consists of $N_c = 3$ random scattering clusters (with $L_s = 2$ and $L_v = 1$), and each cluster contains $N_s = 20$ subpaths. The AoDs are randomly generated in the 3GPP 3D channel model, where the azimuth AoDs range from $-180°$ to $180°$ and the elevation AoDs range from $-90°$ to $90°$. The system

supports $K = 30$ MUs simultaneously, and minimum meansquared error (MMSE) precoder is adopted at the BS. All the results are obtained by averaging over 200 Monte Carlo channel realizations. Figs. 8 and 9 show the NMSE and the sum spectral efficiency achieved by the different strategies versus the number of training pilot symbols T, respectively, and Figs. 10 and 11 show the NMSE and the sum spectral efficiency achieved by the different strategies versus SNR, respectively. It can be seen that our proposed method indeed works for the 2D array, and the results reverify that our method can substantially improve the channel estimation performance, as well as the sum spectral efficiency.

VI. CONCLUSION

The problem of joint downlink channel estimation and user grouping in massive MIMO systems is addressed in this paper. We first provide a general model to capture a more general sparse structure for user grouping. Then, we propose an

SBL-based framework to handle the general sparsity model, which can fully exploit the common sparsity to cluster nearby users and exclude the harmful effect from outliers simultaneously. To the best of our knowledge, our work is the first to utilize an off-grid SBL-based framework to jointly estimate the channel and cluster the users. Simulation results demonstrate that our method indeed works for the general sparsity model and can significantly improve the channel estimation performance when the uniform sparsity assumption fails to hold. Moreover, it is worth noting that extending our method with the DP prior and an automatically determined G is straightforward.

APPENDIX

A. Proof of Lemma 3

The objective function in (21) can be rewritten as

$$
\mathcal{U}(q_1, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)})
$$

$$
= \int q_1 q_2^{(i)} q_3^{(i)} q_4^{(i)} q_5^{(i)} \ln \frac{p(\mathbf{Y}, \boldsymbol{\Theta})}{q_1 q_2^{(i)} q_3^{(i)} q_4^{(i)} q_5^{(i)}} d\boldsymbol{\Theta}) \tag{65}
$$

$$
\propto \int q_1 \left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i)}(\boldsymbol{\Theta} \backslash \Theta_1)} d\Theta_1 - \int q_1 \ln q_1 d\Theta_1 \tag{66}
$$

$$
= \int q_1 \ln \frac{\exp\left(\left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i)}(\boldsymbol{\Theta} \backslash \Theta_1)} \right)}{q_1} d\Theta_1 \tag{67}
$$

$$
\leq \ln \int q_1 \frac{\exp\left(\left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i)}(\boldsymbol{\Theta} \backslash \Theta_1)} \right)}{q_1} d\Theta_1 \tag{68}
$$

$$
= \ln \int \exp\left(\left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i)}(\boldsymbol{\Theta} \backslash \Theta_1)} \right) d\Theta_1, \tag{69}
$$

where $\boldsymbol{\Theta} \backslash \Theta_j$ stands for the set $\boldsymbol{\Theta}$ excluding Θ_j, and Jensen's inequality is applied in (68). Clearly, the objective function in (21) is maximized, if the inequality in (68) holds strictly, which means the optimization problem (21) has a unique solution:

$$
\ln q^{(i+1)}(\alpha)
$$

$$
\propto \left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i)}(\bar{\mathbf{W}}) q^{(i)}(\boldsymbol{\Gamma}^*) q^{(i)}(\boldsymbol{\Gamma}^v) q^{(i)}(\mathbf{Z})} \tag{70}
$$

$$
\propto \left\langle \ln p(\mathbf{Y}|\bar{\mathbf{W}}, \alpha) \right\rangle_{q^{(i)}(\bar{\mathbf{W}})} + \ln p(\alpha)
$$

$$
\propto (a + KT - 1) \ln \alpha
$$

$$
- \alpha \left(b + \sum_{k=1}^{K} \left(\|\mathbf{y}_k - \boldsymbol{\Phi} \boldsymbol{\mu}_k^{(i)}\|_2^2 + \operatorname{tr}(\boldsymbol{\Phi} \boldsymbol{\Sigma}_k^{(i)} \boldsymbol{\Phi}^H) \right) \right), \tag{71}
$$

where $\boldsymbol{\mu}_k^{(i)} \triangleq \langle \mathbf{w}_k \rangle_{q^{(i)}(\bar{\mathbf{w}}_k)}$ and $\boldsymbol{\Sigma}_k^{(i)} \triangleq \langle (\mathbf{w}_k - \boldsymbol{\mu}_k^{(i)})(\mathbf{w}_k - \boldsymbol{\mu}_k^{(i)})^H \rangle_{q^{(i)}(\bar{\mathbf{w}}_k)}$ [whose closed-form expressions are given in (34) and (35)]. Hence, $q^{(i+1)}(\alpha)$ obeys a gamma distribution:

$$
q^{(i+1)}(\alpha) = \Gamma(\alpha | a_\alpha^{(i)}, b_\alpha^{(i)}), \tag{72}
$$

where $a_\alpha^{(i)} = (a + KT)$ and $b_\alpha^{(i)} = b + \sum_{k=1}^{K} (\|\mathbf{y}_k - \boldsymbol{\Phi} \boldsymbol{\mu}_k^{(i)}\|_2^2 + \operatorname{tr}(\boldsymbol{\Phi} \boldsymbol{\Sigma}_k^{(i)} \boldsymbol{\Phi}^H))$.

B. Proof of Lemma 4

Following a similar derivation to (65)–(69), the optimization problem (22) has a unique solution:

$$
\ln q^{(i+1)}(\bar{\mathbf{W}})
$$

$$
\propto \left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i+1)}(\alpha) q^{(i)}(\boldsymbol{\Gamma}^*) q^{(i)}(\boldsymbol{\Gamma}^v) q^{(i)}(\mathbf{Z})} \tag{73}
$$

$$
\propto \sum_{k=1}^{K} \left\langle \ln p(\mathbf{y}_k, \boldsymbol{\Theta}) \right\rangle_{q^{(i+1)}(\alpha) q^{(i)}(\boldsymbol{\Gamma}^*) q^{(i)}(\boldsymbol{\gamma}_k^v) q^{(i)}(\mathbf{Z})} \cdot \tag{74}
$$

For each term in (74), we have

$$
\left\langle \ln p(\mathbf{y}_k, \boldsymbol{\Theta}) \right\rangle_{q^{(i+1)}(\alpha) q^{(i)}(\boldsymbol{\Gamma}^*) q^{(i)}(\boldsymbol{\gamma}_k^v) q^{(i)}(\mathbf{z}_k)}
$$

$$
\propto \left\langle \ln p(\mathbf{y}_k|\bar{\mathbf{w}}_k) \right\rangle_{q^{(i+1)}(\alpha)} + \left\langle \ln p(\mathbf{w}_k^v|\boldsymbol{\gamma}_k^v) \right\rangle_{q^{(i)}(\boldsymbol{\gamma}_k^v)}
$$

$$
+ \left\langle \ln p(\mathbf{w}_k^s|\mathbf{z}_k, \boldsymbol{\Gamma}^*) \right\rangle_{q^{(i)}(\boldsymbol{\Gamma}^*) q^{(i)}(\mathbf{z}_k)} \tag{75}
$$

$$
\propto - \hat{\alpha}^{(i+1)} \|\mathbf{y}_k - \bar{\boldsymbol{\Phi}} \bar{\mathbf{w}}_k\|_2^2 - \rho^{-1} (\mathbf{w}_k^v)^H \operatorname{diag}\left((\hat{\boldsymbol{\gamma}}_k^v)^{(i)} \right) \mathbf{w}_k^v
$$

$$
- (\mathbf{w}_k^s)^H \operatorname{diag}\left(\underbrace{\sum_{g=1}^{G} \hat{\phi}_{k,g}^{(i)} (\hat{\boldsymbol{\gamma}}_g^*)^{(i)}}_{\triangleq (\hat{\boldsymbol{\gamma}}_k^s)^{(i)}} \right) \mathbf{w}_k^s, \tag{76}
$$

where $(\hat{\boldsymbol{\gamma}}_k^v)^{(i)} = \langle \boldsymbol{\gamma}_k^v \rangle_{q^{(i)}(\boldsymbol{\gamma}_k^v)}$ and $(\hat{\boldsymbol{\gamma}}_g^*)^{(i)} = \langle \boldsymbol{\gamma}_g^* \rangle_{q^{(i)}(\boldsymbol{\gamma}_g^*)}$.

This equality shows that $q^{(i+1)}(\bar{\mathbf{W}})$ is separable for each $\bar{\mathbf{w}}_k$, and $q^{(i+1)}(\bar{\mathbf{w}}_k)$ follows a Gaussian distribution:

$$
q^{(i+1)}(\bar{\mathbf{w}}_k) = \mathcal{CN}(\bar{\mathbf{w}}_k | \bar{\boldsymbol{\mu}}_k^{(i+1)}, \bar{\boldsymbol{\Sigma}}_k^{(i+1)}), \tag{77}
$$

where $\bar{\boldsymbol{\mu}}_k^{(i+1)} = \hat{\alpha}^{(i+1)} \bar{\boldsymbol{\Sigma}}_k^{(i+1)} \bar{\boldsymbol{\Phi}}^H \mathbf{y}_k$ and $\bar{\boldsymbol{\Sigma}}_k^{(i+1)} = (\hat{\alpha}^{(i+1)} \bar{\boldsymbol{\Phi}}^H \bar{\boldsymbol{\Phi}} + \operatorname{diag}([(\hat{\boldsymbol{\gamma}}_k^s)^{(i)}; \rho^{-1}(\hat{\boldsymbol{\gamma}}_k^v)^{(i)}]))^{-1}$.

C. Proof of Lemma 5

Following a similar derivation to (65)–(69), the optimization problem (23) has a unique solution:

$$
\ln q^{(i+1)}(\boldsymbol{\Gamma}^*)
$$

$$
\propto \left\langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \right\rangle_{q^{(i+1)}(\alpha) q^{(i+1)}(\bar{\mathbf{W}}) q^{(i)}(\boldsymbol{\Gamma}^v) q^{(i)}(\mathbf{Z})} \tag{78}
$$

$$
\propto \sum_{k=1}^{K} \left\langle \ln p(\mathbf{w}_k^s|\mathbf{z}_k, \boldsymbol{\Gamma}^*) \right\rangle_{q^{(i+1)}(\bar{\mathbf{W}}) q^{(i)}(\mathbf{Z})} + \sum_{g=1}^{G} \ln p(\boldsymbol{\gamma}_g^*) \tag{79}
$$

$$
\propto - \sum_{g=1}^{G} \sum_{k=1}^{K} \sum_{l=1}^{\hat{L}} \hat{\phi}_{k,g}^{(i)} \gamma_{g,l}^* \left\langle (w_{k,l}^s)^* w_{k,l}^s \right\rangle_{q^{(i+1)}(\bar{\mathbf{w}}_k)}
$$

$$
+ \sum_{g=1}^{G} \sum_{k=1}^{K} \sum_{l=1}^{\hat{L}} \hat{\phi}_{k,g}^{(i)} \ln \gamma_{g,l}^* + \sum_{g=1}^{G} \sum_{l=1}^{\hat{L}} ((a - 1) \ln \gamma_{g,l}^* - b \gamma_{g,l}^*). \tag{80}
$$

Clearly, $q^{(i+1)}(\boldsymbol{\Gamma}^*)$ is separable for each $\gamma_{g,l}^*$, and we obtain

$$\ln q^{(i)}(\gamma_{g,l}^*) \propto -\gamma_{g,l}^* \left(b + \sum_{k=1}^{K} \hat{\phi}_{k,g}^{(i)} \left(|\bar{\mu}_{k,1,l}^{(i+1)}|^2 + \bar{\Sigma}_{k,1,l}^{(i+1)} \right) \right)$$
$$+ \left(a - 1 + \sum_{k=1}^{K} \hat{\phi}_{k,g}^{(i)} \right) \ln \gamma_{g,l}^*, \tag{81}$$

where $\hat{\phi}_{k,g}^{(i)} \triangleq q^{(i)}(z_{k,g} = 1)$, $\bar{\mu}_{k,1,l}^{(i+1)}$ stands for the l-th element of $\bar{\boldsymbol{\mu}}_{k,1}^{(i+1)}$, and $\bar{\Sigma}_{k,1,l}^{(i+1)}$ stands for the l-th diagonal element of $\bar{\boldsymbol{\Sigma}}_{k,1}^{(i+1)}$. Hence, $q^{(i+1)}(\gamma_{g,l}^*)$ obeys a gamma distribution:

$$q^{(i+1)}(\gamma_{g,l}^*) = \Gamma\left(\gamma_{g,l}^* | (a_{g,l}^*)^{(i+1)}, (b_{g,l}^*)^{(i+1)} \right) \tag{82}$$

with $(a_{g,l}^*)^{(i+1)} = a + \sum_{k=1}^{K} \hat{\phi}_{k,g}^{(i)}$ and $(b_{g,l}^*)^{(i+1)} = b + \sum_{k=1}^{K} \hat{\phi}_{k,g}^{(i)} (|\bar{\mu}_{k,1,l}^{(i+1)}|^2 + \bar{\Sigma}_{k,1,l}^{(i+1)})$.

D. Proof of Lemma 7

Following a similar derivation to (65)–(69), the optimization problem (25) has a unique solution:

$$\ln q^{(i+1)}(\mathbf{Z})$$
$$\propto \langle \ln p(\mathbf{Y}, \boldsymbol{\Theta}) \rangle_{q^{(i+1)}(\alpha)q^{(i+1)}(\bar{\mathbf{W}})q^{(i+1)}(\boldsymbol{\Gamma}^*)q^{(i+1)}(\boldsymbol{\Gamma}^v)} \tag{83}$$
$$\propto \sum_{k=1}^{K} \langle \ln p(\mathbf{w}_k^s | \mathbf{z}_k, \boldsymbol{\Gamma}^*) \rangle_{q^{(i+1)}(\bar{\mathbf{W}})q^{(i+1)}(\boldsymbol{\Gamma}^*)}. \tag{84}$$

From (84) and the fact that $\mathbf{z}_k$ is a discrete vector, we are able to exhaustively calculate the value of $\ln q^{(i+1)}(z_{k,g} = 1)$, $\forall k, g$ as

$$\ln q^{(i+1)}(z_{k,g} = 1)$$
$$\propto \underbrace{\sum_{l=1}^{\hat{L}} (\widehat{\ln \gamma_{g,l}^*})^{(i+1)} - \sum_{l=1}^{\hat{L}} (\gamma_{g,l}^*)^{(i+1)} \left(|\bar{\mu}_{k,1,l}^{(i+1)}|^2 + \bar{\Sigma}_{k,1,l}^{(i+1)} \right)}_{=\varsigma_{k,g}^{(i+1)}}.$$

Since $\sum_{g=1}^{G} q^{(i+1)}(z_{k,g} = 1) = 1$, we obtain

$$\hat{\phi}_{k,g}^{(i+1)} = q^{(i+1)}(z_{k,g} = 1) = \frac{\exp(\varsigma_{k,g}^{(i+1)})}{\sum_{g=1}^{G} \exp(\varsigma_{k,g}^{(i+1)})}. \tag{85}$$

E. Proof of Lemma 8

The non-decreasing property can be achieved by

$$\mathcal{U}(q_1^{(i)}, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)})$$
$$\leq \mathcal{U}(q_1^{(i+1)}, q_2^{(i)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)}) \tag{86}$$
$$\leq \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i)}, q_4^{(i)}, q_5^{(i)}) \tag{87}$$
$$\leq \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i)}, q_5^{(i)}) \tag{88}$$
$$\leq \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i+1)}, q_5^{(i)}) \tag{89}$$
$$\leq \mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i+1)}, q_5^{(i+1)}), \tag{90}$$

where (86), (87), (88), (89) and (90) follow (21), (22), (23), (24) and (25), respectively.

F. Proof of Lemma 9

From Section III-C, it is clear that $q(\boldsymbol{\Theta}) = q(\alpha)q(\bar{\mathbf{W}})q(\boldsymbol{\Gamma}^*)q(\boldsymbol{\Gamma}^v)q(\mathbf{Z})$ can be considered as some parameterized functions, e.g., a gamma distribution function with parameters a_α and b_α for $q(\alpha)$, a Gaussian distribution function with parameters $\bar{\boldsymbol{\mu}}_k$s and $\bar{\boldsymbol{\Sigma}}_k$s for $q(\bar{\mathbf{W}})$, and so on. As a result, the optimization problem (20) which is optimized over function spaces can be converted into a conventional parameter optimization problem. Therefore, the definition and convergence result for the conventional stationary point can be applied.

Let the surrogate function be chosen as the objective function itself, and then, according to Theorem 2-b in [50], the proposed algorithm converges to a stationary solution because the problems in (21)–(25) have a unique solution.

G. Derivation for Eq. (56)

Ignoring the independent terms, the objective function in (55) becomes

$$\mathcal{U}(q_1^{(i+1)}, q_2^{(i+1)}, q_3^{(i+1)}, q_4^{(i+1)}, q_5^{(i+1)}, \mathbf{B})$$
$$\propto \left\langle \sum_{k=1}^{K} \ln p(\mathbf{y}_k | \mathbf{w}_k^s, \mathbf{w}_k^v, \alpha, \boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) \right\rangle_{q^{(i+1)}(\alpha)q^{(i+1)}(\bar{\mathbf{W}})}$$
$$\propto -\hat{\alpha}^{(i+1)} \sum_{k=1}^{K} \left\| \mathbf{y} - \boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) \boldsymbol{\mu}_k^{(i+1)} \right\|_2^2$$
$$- \hat{\alpha}^{(i+1)} \sum_{k=1}^{K} \text{tr}\left(\boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) \boldsymbol{\Sigma}_k^{(i+1)} \boldsymbol{\Phi}^H(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) \right).$$

Obviously, the objective function is separable for each $\boldsymbol{\beta}_k$. Calculating the derivative of each term in the above equality w.r.t. $\beta_{k,l}$, we obtain

$$\frac{\partial \left\| \mathbf{y} - \boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k) \boldsymbol{\mu}_k^{(i+1)} \right\|_2^2}{\partial \beta_{k,l}}$$
$$= \frac{\partial \left\| \mathbf{y}_{k-l}^{(i+1)} - \mu_{k,l}^{(i+1)} \cdot \mathbf{X}(\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l})) \right\|_2^2}{\partial \beta_{k,l}}$$
$$= 2\text{Re}\left((\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}))^H \mathbf{X}^H \mathbf{X} \mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}) \right)$$
$$\cdot |\mu_{k,l}^{(i+1)}|^2 - 2\text{Re}\left((\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}))^H \mathbf{X}^H \right.$$
$$\left. \cdot (\mu_{k,l}^{(i+1)})^* \mathbf{y}_{k-l}^{(i+1)} \right)$$

and

$$\frac{\partial \text{tr}\left(\boldsymbol{\Phi}(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)\boldsymbol{\Sigma}_k^{(i+1)}\boldsymbol{\Phi}^H(\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)\right)}{\partial \beta_{k,l}}$$

$$= 2\text{Re}\left((\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}))^H \mathbf{X}^H \mathbf{X}\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l})\right)$$

$$\cdot \chi_{k,ll}^{(i+1)} + 2\text{Re}\left((\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}))^H \mathbf{X}^H \mathbf{X}\right.$$

$$\left. \times \sum_{j \neq l} \chi_{k,jl}^{(i+1)}\mathbf{a}(\hat{\vartheta}_j + \beta_{k,j}, \varphi_{k,j})\right),$$

where $\mathbf{y}_{k-l}^{(i+1)} = \mathbf{y}_k - \mathbf{X} \cdot \sum_{j \neq l}(\mu_{k,j}^{(i+1)} \cdot \mathbf{a}(\hat{\vartheta}_j + \beta_{k,j}, \varphi_{k,l}))$, $\mathbf{a}'(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l}) = d\mathbf{a}(\hat{\vartheta}_l + \beta_{k,l}, \varphi_{k,l})/d\beta_{k,l}$, and $\mu_{k,l}^{(i+1)}$ and $\chi_{k,jl}^{(i+1)}$ denote the l-th element and the (j,l)-th element of $\boldsymbol{\mu}_k^{(i+1)}$ and $\boldsymbol{\Sigma}_k^{(i+1)}$, respectively. Hence, the derivative element $\zeta^{(i+1)}(\beta_{k,l})$ in (57) is achieved.
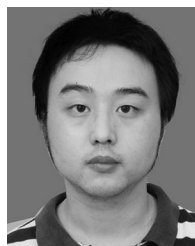
REFERENCES

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[2] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[3] 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.

[4] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive MIMO: Benefits and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 742–758, Oct. 2014.

[5] J.-C. Shen, J. Zhang, K.-C. Chen, and K. B. Letaief, "High-dimensional CSI acquisition in massive MIMO: Sparsity-inspired approaches," *IEEE Syst. J.*, vol. 11, no. 1, pp. 32–40, Mar. 2017.

[6] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 160–171, Feb. 2013.

[7] Y. Xu, G. Yue, and S. Mao, "User grouping for massive MIMO in FDD systems: New design methods and analysis," *IEEE Access*, vol. 2, pp. 947–959, 2014.

[8] Z. Gao, L. Dai, W. Dai, B. Shim, and Z. Wang, "Structured compressive sensing-based spatio-temporal joint channel estimation for FDD massive MIMO," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 601–617, Feb. 2016.

[9] 3GPP, "First 5G NR Specs Approved," 2017. [Online]. Available: http://www.3gpp.org/newsevents/3gpp-news/1929-nsa_nr_5g.

[10] D. Mi, M. Dianati, L. Zhang, S. Muhaidat, and R. Tafazolli, "Massive MIMO performance with imperfect channel reciprocity and channel estimation error," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3734–3749, Sep. 2017.

[11] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 7112–7139, Nov. 2014.

[12] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169–6183, Dec. 2015.

[13] X. Rao and V. K. Lau, "Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261–3271, Jun. 2014.

[14] J. Hoydis, C. Hoek, T. Wild, and S. ten Brink, "Channel measurements for large antenna arrays," in *Proc. Int. Symp. Wireless Commun. Syst.*, 2012, pp. 811–815.

[15] A. Liu, F. Zhu, and V. K. Lau, "Closed-loop autonomous pilot and compressive CSIT feedback resource adaptation in multi-user FDD massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 173–183, Jan. 2017.

[16] C.-K. Wen, S. Jin, K.-K. Wong, J.-C. Chen, and P. Ting, "Channel estimation for massive MIMO using Gaussian-mixture Bayesian learning," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1356–1368, Mar. 2015.

[17] Z. Chen and C. Yang, "Pilot decontamination in wideband massive MIMO systems by exploiting channel sparsity," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5087–5100, Jul. 2016.

[18] J.-C. Shen, J. Zhang, E. Alsusa, and K. B. Letaief, "Compressed CSI acquisition in FDD massive MIMO: How much training is needed?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4145–4156, Jun. 2016.

[19] X. Rao and V. K. Lau, "Compressive sensing with prior support quality information and application to massive MIMO channel estimation with temporal correlation," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4914–4924, Sep. 2015.

[20] J. Choi, D. J. Love, and P. Bidigare, "Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 802–814, Oct. 2014.

[21] L. You, X. Gao, A. L. Swindlehurst, and W. Zhong, "Channel acquisition for massive MIMO-OFDM with adjustable phase shift pilots," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1461–1476, Oct. 2016.

[22] Z. Gao, C. Zhang, Z. Wang, and S. Chen, "Priori-Information aided iterative hard threshold: A low-complexity high-accuracy compressive sensing based channel estimation for TDS-OFDM," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 242–251, Jan. 2015.

[23] A. Liu, V. K. Lau, and W. Dai, "Exploiting burst-sparsity in massive MIMO with partial channel support Information," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7820–7830, Nov. 2016.

[24] Y. Ding and B. D. Rao, "Channel estimation using joint dictionary learning in FDD massive MIMO systems," in *Proc. IEEE GlobalSIP*, 2015, pp. 185–189.

[25] Y. Ding and B. D. Rao, "Compressed downlink channel estimation based on dictionary learning in FDD massive MIMO systems," in *Proc. IEEE GLOBECOM*, 2015, pp. 1–6.

[26] Y. Ding and B. D. Rao, "Dictionary learning based sparse channel representation and estimation for FDD massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5437–5451, Aug. 2018.

[27] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[28] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, no. Jun, pp. 211–244, 2001.

[29] J. Dai and H. C. So, "Sparse Bayesian learning approach for outlier-resistant direction-of-arrival estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 744–756, Feb. 2018.

[30] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[31] J. Dai, A. Liu, and V. K. Lau, "FDD massive MIMO channel estimation with arbitrary 2D-array geometry," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2584–2599, May 2018.

[32] Z. Yang, L. Xie, and C. Zhang, "Off-grid direction of arrival estimation using sparse Bayesian inference," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 38–43, Jan. 2013.

[33] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson, "Linear pre-coding performance in measured very-large MIMO channels," in *Proc. Veh. Technol. Conf.*, 2011, pp. 1–5.

[34] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexingłThe large-scale array regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6441–6463, Oct. 2013.

[35] J. Nam, A. Adhikary, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 876–890, 2014.

[36] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Statist.*, vol. 22, no. 2, pp. 231–245, 2013.

[37] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[38] L. Wang, L. Zhao, G. Bi, C. Wan, L. Zhang, and H. Zhang, "Novel wideband DOA estimation based on sparse Bayesian learning with Dirichlet process priors," *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 275–289, Jan. 2016.

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

[40] *Universal Mobile Telecommunications System (UMTS); Spatial Channel Model for Multiple Input Multiple Output (MIMO) Simulations*, 3GPP TR 25.996 version 11.0.0 Release 11, 2012.

[41] A. F. Molisch, A. Kuchar, J. Laurila, K. Hugl, and R. Schmalenberger, "Geometry-based directional model for mobile radio channelsłprinciples and implementation," *Trans. Emerging TeleCommun. Technol.*, vol. 14, no. 4, pp. 351–359, 2003.

[42] J. Dai, X. Bao, W. Xu, and C. Chang, "Root sparse Bayesian learning for off-grid DOA estimation," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 46–50, Jan. 2017.

[43] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Jun. 2008.

[44] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 187–200, Jan. 1999.

[45] W. Shen, L. Dai, B. Shim, Z. Wang, and R. W. Heath, "Channel feedback based on aod-adaptive subspace codebook in FDD massive MIMO systems," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5235–5248, Nov. 2018.

[46] S. J. Wright and J. Nocedal, *Numerical Optimization*. Berlin, Germany: Springer Science 2006.

[47] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[48] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[49] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on 3D Channel Model for LTE*, 3GPP TR 36.873 version 12.2.0 Release 12, 2015.

[50] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, Dep. Elect. Comput. Eng., Univ. Minnesota, Minneapolis, MN, USA, 2014.
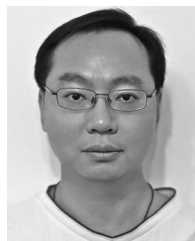
An Liu (S'07–M'09–SM'17) received the B.S. and Ph.D. degrees in electrical engineering from Peking University, Beijing, China, in 2011 and 2004, respectively. From 2008 to 2010, he was a Visiting Scholar at the Department of Electrical, Computer, and Energy Engineering, University of Colorado at Boulder. He was a Postdoctoral Research Fellow during 2011–2013, Visiting Assistant Professor in 2014, and Research Assistant Professor during 2015–2017, with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology. He is currently a Distinguished Research Fellow with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. His research interests include wireless communications, stochastic optimization, and compressive sensing.

Jisheng Dai (S'08–M'11) received the B.Eng. degree in electronic engineering from the Nanjing University of Technology, Nanjing, China, in 2005, and the Ph.D. degree in information and communication engineering from the University of Science and Technology of China, Hefei, China, in 2010. He was a Research Assistant with the Department of Electrical Engineering, University of Hong Kong, in 2009, and Visiting Scholar with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, during 2017–2018. He is currently a Research Associate with the School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, China. His research interests include convex optimization theory, wireless communications, machine learning, and bioinformatics.

Vincent K. N. Lau (SM'04–F'12) received the B.Eng. (Distinction 1st Hons.) from the University of Hong Kong, Hong Kong, in 1992 and the Ph.D. from the Cambridge University, Cambridge, U.K., in 1997. He was with Bell Labs from 1997 to 2004 and the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology (HKUST), in 2004. He is currently a Chair Professor and the Founding Director of Huawei-HKUST Joint Innovation Lab, HKUST. His current research focus includes robust and delay-optimal cross layer optimization for MIMO/OFDM wireless systems, and interference mitigation techniques for wireless networks, massive MIMO, M2M, and network control systems.