

Centralized and Decentralized Channel Estimation in FDD Multi-User Massive MIMO Systems

Anupama Rajoriya, Rohit Budhiraja and Lajos Hanzo, *Fellow, IEEE*

Abstract— We design a centralized and a decentralized variational Bayesian learning (C- and D-VBL) algorithms for the base station (BS) of a frequency division duplex massive multiple input multiple output (mMIMO) cellular system, wherein users send compressed information for it to estimate their downlink channels. The BS in the decentralized algorithm consists of multiple processing units (PUs), and each PU separately estimates the channels of a group of users, by employing the proposed D-VBL algorithm. To reduce channel estimation error, the PUs exploit the structured sparsity inherent in multi-user mMIMO channels by exchanging information among themselves. We investigate the proposed C-VBL and low-complexity D-VBL algorithms and show that i) they substantially outperform the state-of-the-art centralized and decentralized algorithms in terms of the normalized mean squared error and the bit error rate. This is because they beneficially exploit the *inherent* channel sparsity, while the existing state-of-the-art solutions fail to do so. The proposed D-VBL is also robust to PU failures, and provides a similar performance as its centralized counterpart (C-VBL), but with a much reduced complexity.

Index Terms— Decentralized architecture, frequency division duplex, variational Bayesian learning.

I. INTRODUCTION

Massive multiple input multiple output (mMIMO) systems relying on a large number of antennas at the base station (BS) enable a wireless system to achieve high spectral and energy efficiency [1]. To fully realize these gains, the BS requires downlink channel state information (CSI), which is easy for it to infer in a time division duplex (TDD) system, from the uplink channel by assuming reciprocity [1, and references therein], [2]. In frequency division duplexing (FDD) systems, having non-reciprocal uplink (UL) and downlink (DL) channels, DL CSI estimation was investigated [3]–[7].

The authors of [3] showed that when different users share common scatterers at the BS, their channel matrices exhibit *common sparsity*, i.e., the angle of arrival (AoA) and angle of departure pair (AoD) of the significant multipath components is the same for all the users. On the other hand, user-specific local scatterers at the BS cause *user-specific sparsity*. Rao and Lau in [4] proposed a joint-orthogonal matching pursuit (J-OMP) algorithm to estimate channels in multi-user mMIMO systems by exploiting both user-specific and common sparsities. Tseng *et al.* in [5] developed a two-stage weighted ℓ_1 -minimization-based channel estimator to exploit both these sparsities. These algorithms, however, require knowledge of the sparsity level, which is non-trivial to acquire [3], [4]. Li *et al.* in [7] developed a deep learning based DL channel

estimation technique for FDD mMIMO systems with single-antenna users. As a further advancement, the authors of [8], [9], without considering any particular kind of shared sparsity across users, developed an orthogonal matching pursuit based DL channel estimation algorithm for FDD mMIMO systems. All of these works either capture common and/or user-specific sparsities. In addition to these sparsities, since the group of users in close proximity of each other share the same set of local scatterers at the BS, they share *cluster-specific sparsity*.

The users in [3]–[6] first compress the DL pilot observations, and then send them back to the BS for centrally estimating their channels. In mMIMO systems, centralized BS processing has a high computational complexity [19], [20]. Decentralized low-complexity BS architectures were proposed in [19]–[22], where multiple processing units (PUs) decentrally process the signals of different antenna groups by exchanging information via a central control unit (CCU). This architecture is shown to be scalable with the number of users and antennas, and is eminently suitable for the next-generation cellular systems having a large number of antennas and users.

The existing mMIMO and the sensor network literature have made some progress in the decentralized channel estimation [10]–[13]. Reference [10] proposed a pair of distributed algorithms for joint active user detection (AUD) and channel estimation (CE) in uplink cloud radio systems relying on such BS architectures, namely the alternating direction method of multipliers (ADMM) and hybrid block coordinate descent. The authors of [11] proposed an approximate message passing (AMP)-based algorithm for AUD and CE in a distributed cell-free mMIMO system with lot of access points (APs). The APs do not communicate with each other as the users do not share any sparsity. The authors assumed only block channel sparsity, instead of the combination of user-specific, cluster-specific and common sparsities. Our proposed solution, by contrast, develops a decentralized algorithm, which completely captures this structured sparsity and scales well with a large number of users in multi-user mMIMO. The authors of [12] proposed a consensus-based decentralized sparse Bayesian learning (SBL) algorithm, which exploited the joint sparsity by using the ADMM to solve multiple consensus optimization problems. Reference [13] proposed a fusion-based decentralized SBL algorithm, wherein each node computes a local maximum a-posteriori estimate of the sparse vector, and then refines it using the messages received from its neighboring nodes in each iteration. However, these algorithms only consider the jointly sparse structure of vectors across different sensors, rather than the user-specific, cluster-specific and common sparsities of multi-user mMIMO channels. It is difficult to extend the above algorithms to multi-user mMIMO systems associated with the aforementioned sparsities. Furthermore, the computational cost of each log-likelihood ratio test is high

Anupama Rajoriya and Rohit Budhiraja are with the Department of Electrical Engineering, IIT Kanpur, India. email: {anupama, rohitbr}@iitk.ac.in. Lajos Hanzo is with University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: lh@ecs.soton.ac.uk).

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/P034284/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028)

Table I: Summary of decentralized channel estimation mMIMO literature.

Keywords	[4] 2014	[5] 2016	[3] 2017	[6] 2019	[10] 2018	[11] 2020	[12] 2018	[13] 2019	[14] 2017	[15] 2017	[16] 2019	[17] 2020	[18] 2018	Proposed
Multi-user mMIMO	✓	✓	✓	✓				✓					✓	✓
Downlink	✓	✓	✓											✓
Structured sparsity			✓	✓										✓
Require sparsity knowledge	✓	✓												
Bayesian learning			✓	✓			✓	✓	✓	✓	✓		✓	✓
Quantization							✓				✓	✓	✓	✓
Decentralized BS architecture					✓	✓			✓	✓				✓

for a mMIMO system. Additionally, as discussed in [19], these high-resolution observations may clog up the interconnect links in a D-VBL architecture. To obviate this problem, we assume that the CCU first *quantizes* observations \mathbf{Y}_k and then sends them to each PU, which decentrally estimates the channels of a user group by employing the proposed algorithm.

Various quantization based schemes exist in literature, for low-complexity parameter estimation [14]–[16], [23]. The authors of [23] developed a quantization-constrained minimum mean squared error estimator for a general parameter estimation problem. They, however, did not consider any decentralized processing or sparsity in the unknown parameter, and thus their solution can not be extended for DL mMIMO channel estimation. As a further advancement, Jeon *et al.* in [14] studied this effect in mmWave MIMO systems by deriving log-likelihood ratios for the data bits based on the quantized observations. By contrast, Schniter *et al.* in [16] developed an AMP-based channel estimation algorithm for single-user mmWave MIMO systems considering few-bit ADCs at the receiver. These works, however, modeled the ADC noise of single-user mmWave MIMO systems. Then Jeon *et al.* in [15] also developed reinforcement learning based data detection for single-user MIMO systems employing 1-bit ADCs. However, none of the above authors estimated the sparse channel of multi-user mMIMO systems using quantized observations.

Considering the gap in the existing multi-user mMIMO literature, which can be explicitly seen from Table I, we propose centralized and decentralized channel estimation algorithms for such systems. The detailed **contributions** of this work can next be summarized as follows.

1) We propose a centralized variational Bayesian learning (C-VBL) algorithm to estimate a mMIMO channel with common, user-specific and additionally *cluster-specific sparsity*, which is shared by the users in each other’s close proximity. We propose a mixture of three Gaussian distributions as a prior, where the mixing distributions are associated with multinoulli variables. Most of the existing DL channel estimation works i.e., [3]–[5] fail to capture the cluster-specific sparsity, which occurs due to co-located users. Cheng *et al.* [17] proposed a Dirichlet-Gaussian hybrid prior which exploits user-specific, cluster-specific and common sparsities by adaptively grouping the users. However, it is difficult to extend this algorithm to a decentralized architecture, due to its adaptive user grouping.

2) We extend the C-VBL framework to design a decentralized VBL (D-VBL) for a BS having multiple PUs. Each PU separately estimates the sparse channels of a group of users assigned to it by exchanging information with other

PUs, via the CCU, to improve estimation. We show, both analytically and numerically, that the D-VBL algorithm has lesser complexity than the C-VBL algorithm. *This aspect is important for next-generation BS architectures, which employ multiple PUs to perform various signal processing tasks.*

3) We analyze the convergence of our D-VBL algorithm and show that the upper bound on the absolute error between the C- and D-VBL updates tends to zero, when the SNR-based criterion detects the non-zero support accurately.

4) We numerically show that the proposed D-VBL and C-VBL algorithms have much lower normalized mean squared error (NMSE) and bit error rate (BER) than the centralized algorithms e.g., J-OMP [4], variational expectation maximization (VEM) [3], adaptive grouping sparse Bayesian learning (AG-SBL) [17], and the decentralized ones in [13], [24]. We also show that proposed algorithms outperform the existing ones in terms of spectral efficiency (SE) and energy efficiency (EE), and that the D-VBL has better EE than the C-VBL.

Notations: The symbols $\mathbb{C}^{r \times s}$ and $\{0, 1\}^{r \times s}$ represent a complex and binary matrix of dimension $r \times s$, respectively. The point-wise product of two vectors \mathbf{a} and \mathbf{b} is denoted as $\mathbf{a} \circ \mathbf{b}$. The trace and the expectation with respect to the distribution $p(\cdot)$, are denoted by $\text{Tr}\{\cdot\}$ and $\langle \cdot \rangle_{p(\cdot)}$, respectively, and $\langle f(\mathbf{z}) \rangle$ denotes the expectation of $f(\mathbf{z})$ with respect to the posterior distribution of \mathbf{z} . The symbol \mathbf{I}_J denotes an $J \times J$ sized identity matrix and $\text{diag}(\mathbf{z})$ denotes a diagonal matrix with elements of \mathbf{z} on its diagonal. The notation $\mathcal{N}_{\mathcal{C}}(\mathbf{x}|\mathbf{a}, \mathbf{B})$ conveys that \mathbf{x} is a complex normal random vector with mean vector \mathbf{a} and covariance matrix \mathbf{B} . The notation $\text{supp}(\mathbf{a})$ denotes the index set of non-zero entries in vector \mathbf{a} . The notation $\mathbb{I}(\cdot)$ denotes the indicator function. The size of a set \mathcal{C} is denoted by $|\mathcal{C}|$. The notations $\Re\{\mathbf{y}\}$ and $\Im\{\mathbf{y}\}$ represent the real and imaginary parts of the vector \mathbf{y} , respectively. The $[n, m]$ th entry of the matrix \mathbf{X} is denoted as $\mathbf{X}^{[n, m]}$. The n th entry of the vector \mathbf{a} is denoted as $\mathbf{a}_{[n]}$.

The rest of this paper is organized as follows. We describe in Section II, the system and channel model of the downlink FDD multi-user mMIMO. In Section III and Section IV, we develop the proposed C-VBL and D-VBL algorithms, respectively, in detail. We discuss the convergence of the D-VBL algorithm in Section V, and numerically investigate the performance of the proposed scheme in Section VI.

II. SYSTEM MODEL

We consider a FDD multi-user mMIMO system having an N -antenna BS and K users, each equipped with M antennas. To estimate the DL channel of users, the FDD BS broadcasts a pilot matrix $\mathbf{Q} \in \mathbb{C}^{N \times T}$ in T time slots. Here we have

$\text{Tr}(\mathbf{Q}^H \mathbf{Q}) = PT$ with P being the transmit power per slot. The $M \times T$ pilot matrix received by the k th user is

$$\mathbf{R}_k = \mathbf{H}_k \mathbf{Q} + \mathbf{N}_k, \quad (1)$$

where $\mathbf{H}_k \in \mathbb{C}^{M \times N}$ is the channel between the BS and the k th user. The noise matrix at the k th user $\mathbf{N}_k \in \mathbb{C}^{M \times T}$ has independent and identically distributed (i.i.d.) complex Gaussian entries with zero mean and σ_k^2 variance. Similar to [3]–[5], each user sends \mathbf{R}_k back to the BS, which has a decentralized architecture with L baseband PUs [19]. The BS segregates the observations received from different users into L groups and assigns each group to a PU, which separately estimates their channels, by exchanging information via CCU. Since all the PUs have the same computational capabilities, the network is assumed to be homogeneous. The l th PU is assigned the observation of the user set $\mathcal{C}_l \subseteq \{1, \dots, K\}$, where \mathcal{C}_l is the cluster/group of users so that $\bigcup_{l=1}^L \mathcal{C}_l = \{1, \dots, K\}$ and $\bigcap_{l=1}^L \mathcal{C}_l = \phi$. The l th PU thus has $|\mathcal{C}_l|$ user measurement matrices so that $\sum_{l=1}^L |\mathcal{C}_l| = K$. We next discuss the channel model. Before that, a remark is in order.

Remark 1. The wireless channel exhibits wide sense stationarity [25], i.e., its statistical properties, such that its covariance matrix, remains constant for a long time duration. The channel's covariance matrix represents its scattering structure [26], and consequently its sparsity. The authors of [26], [27] proposed advanced algorithms for estimating the covariance matrix of mMIMO channels using a limited number of sample observations. Based on [26], [27], we assume that the BS has the users' covariance information, which are grouped into multiple clusters depending on the similarity of their covariance matrices. The similarity can be quantified, e.g., using chordal distance [26].

Angular domain channel model: We assume that the BS and the users employ uniform linear arrays (ULA) for which the DL channel of the k th user can be modeled as [25]:

$$\mathbf{H}_k = \sum_{m=1}^M \sum_{n=1}^N h_{m,n,k} \mathbf{a}_R(\vartheta_{R[m]}) \mathbf{a}_T^H(\vartheta_{T[n]}). \quad (2)$$

The scalars $\vartheta_{T[n]}$ and $\vartheta_{R[m]}$ represent the n th and the m th entry of the AoD vector $\vartheta_T \in \mathbb{C}^{N \times 1}$ and the AoA vector $\vartheta_R \in \mathbb{C}^{M \times 1}$ respectively, while $h_{m,n,k}$ is the path gain for the n th AoD and the m th AoA. The AoD (AoA) is defined as the angle of elevation of the path between the BS (user) and the scatterer causing this path [25]. The vectors $\mathbf{a}_T(\vartheta_{T[n]}) \in \mathbb{C}^{N \times 1}$ and $\mathbf{a}_R(\vartheta_{R[m]}) \in \mathbb{C}^{M \times 1}$ are the steering vectors of the BS and user antenna array, respectively. For an N -element transmit/receive ULA, the array steering vector $\mathbf{a}_T(\vartheta)/\mathbf{a}_R(\vartheta)$ has the form [25]: $\frac{1}{\sqrt{N}} [1, e^{-j \frac{2\pi d \sin \vartheta}{\lambda}}, \dots, e^{-j \frac{2\pi(N-1)d \sin \vartheta}{\lambda}}]^T$, where d is the antenna spacing, λ is the carrier wavelength and $\vartheta \in [-\pi/2, \pi/2]$ is the physical AoA/AoD. With these definitions, the channel \mathbf{H}_k can be equivalently expressed as

$$\mathbf{H}_k = \mathbf{A}_R \mathbf{H}_k^a \mathbf{A}_T^H. \quad (3)$$

Here $\mathbf{H}_k^a \in \mathbb{C}^{M \times N}$ is the angular domain channel matrix. Its $[m, n]$ th element is the gain of the path with the m th AoA and the n th AoD [25]. The matrices $\mathbf{A}_R = [\mathbf{a}_R(\vartheta_{R[1]}), \dots, \mathbf{a}_R(\vartheta_{R[M]})] \in \mathbb{C}^{M \times M}$ and $\mathbf{A}_T =$

$[\mathbf{a}_T(\vartheta_{T[1]}), \dots, \mathbf{a}_T(\vartheta_{T[N]})] \in \mathbb{C}^{N \times N}$ are the receive and transmit array response matrices respectively, which we assume to be discrete Fourier transform (DFT) matrices so that we have $\mathbf{A}_T^H \mathbf{A}_T = \mathbf{I}_N$ and $\mathbf{A}_R^H \mathbf{A}_R = \mathbf{I}_M$ [28]. The DFT array response matrices result in uniform sampling of the angles in $[-\pi/2, \pi/2]$. This choice of matrices is the same as the one used for virtual angular domain channel representation in [28]. It is shown to accurately capture the true physical angular channel [28]. Equation (1), using (3), can now be expressed as

$$\mathbf{Y}_k = \Phi \mathbf{X}_k + \mathbf{E}_k. \quad (4)$$

Here we have $\mathbf{Y}_k = \mathbf{R}_k^H \mathbf{A}_R \in \mathbb{C}^{T \times M}$, $\Phi = \mathbf{Q}^H \mathbf{A}_T \in \mathbb{C}^{T \times N}$, $\mathbf{X}_k = (\mathbf{H}_k^a)^H \in \mathbb{C}^{N \times M}$ and $\mathbf{E}_k = \mathbf{N}_k^H \mathbf{A}_R \in \mathbb{C}^{T \times M}$. The BS estimates the matrix $\mathbf{X}_k, \forall k$. In mMIMO systems, there are several clusters/scatterers present, which are active or inactive, depending on their distance from the BS and user. This is because the distance will decide the strength of this multipath component. A stronger multipath component will lead to that scatterer being active. For example, in Fig. 1, the local scatterer 1 at the BS is equi-distant from all the users. It will thus cause significant multipath components for all the users, and will be active for all the users. The multipath components from this scatterer are represented by the blue colored rows in the matrices $\mathbf{X}_k, \forall k = 1, 2, 3$ and 4.

• Furthermore, the presence of only few scatterers at the BS side engenders a low number of significant multipath components [25]. This leads to having only a few non-zero rows in the channel matrix \mathbf{X}_k . The user side is however constituted by a rich scattering environment. This results in each AoA contributing to the multipath propagation. Since all the AoAs are active, similar to [4], we refer to them as isotropic AoAs, shown in Fig. 1 using orange color dashed circles. The isotropic AoAs cause all the elements of these rows to be non-zero. Mathematically,

$$\text{supp}(\mathbf{X}_k^{[:,1]}) = \dots = \text{supp}(\mathbf{X}_k^{[:,m]}) \triangleq \Omega_k^t, \quad \forall k, \quad (5)$$

where Ω_k^t is the total sparsity support of the k th user.¹

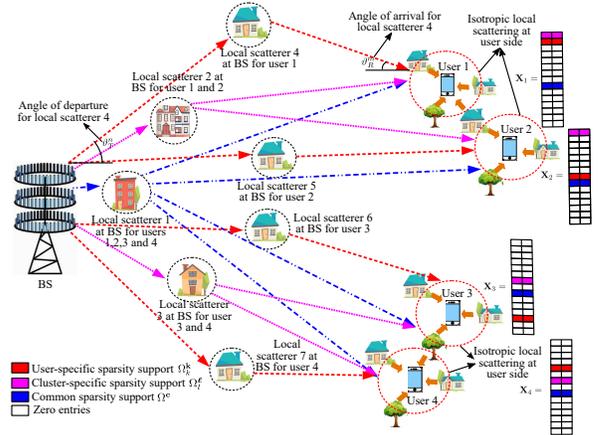


Fig. 1: Structured sparsity in multi-user mMIMO channel.

We next discuss different kind of sparsities.

• Among all limited/active local scatterers at the BS side, shown in the black dashed circles, few are specific to a particular user, resulting in active AoDs corresponding to that scatterer. For example, scatterer 4 is specific to user 1, scatterer

¹We assume $k, k' = 1, \dots, K$ users, $l, j = 1, \dots, L$ clusters, $m = 1, \dots, M$ columns and $n = 1, \dots, N$ rows.

5 to user 2, scatterer 6 to user 3 and scatterer 7 to user 4. The paths, due to *user-specific* scatterers and shown using red arrows, lead to *user-specific sparsity*, marked by red non-zero entries in \mathbf{X}_k . Thus, the user-specific support sets of the angular channel matrix \mathbf{X}_k , denoted by Ω_k^k , are disjoint for all k , i.e., $\cap_{k=1}^K \Omega_k^k = \phi$.

- A cluster of closely-located users share some local scatterers at the BS side, which result in a partially shared set of active AoDs, and consequently partially-shared row-sparsity in $\mathbf{X}_k, \forall k$. For example, in Fig. 1, scatterer 2 is common to users 1 and 2, and scatterer 3 is common to users 3 and 4. The user clusters here are $\{1, 2\}$ and $\{3, 4\}$. The paths, due to *cluster-specific* scatterers and shown using magenta arrows, lead to *cluster-specific sparsity*, which is denoted using magenta non-zero entries in \mathbf{X}_k , and denoted as Ω_l^ℓ for the l th cluster.

- The local scatterers closest to the BS side, are common to all the users, since their distance from the users is higher than that between the users. For example, the local scatterer 1 in Fig. 1 is farthest from all the users, which results in the same active AoD for all the users. Such paths engendered by common scatterers, and shown by blue arrows, cause *common sparsity* to all users shown using blue row entries in $\mathbf{X}_k, \forall k$. The common sparsity is mathematically denoted as Ω^c .

- The overall sparsity support of user $k \in \mathcal{C}_l$ is therefore

$$\Omega_k^t = \Omega_k^k \cup \Omega_l^\ell \cup \Omega^c. \quad (6)$$

According to the definitions of these support sets we have, i) common sparsity $\Omega^c = \cap_{k=1}^K \Omega_k^k$; and ii) $\Omega_k^k \cap \Omega_l^\ell \cap \Omega^c = \phi$.² This implies that the cluster-specific support set Ω_l^ℓ does not contain the support set of the common-sparsity, even if it is common to all the users in the l th cluster.

Given the above sparsity of the angular domain channel \mathbf{X}_k , the problem of estimating $\mathbf{X}_k, \forall k$, from the observation matrices $\mathbf{Y}_k, \forall k$, in (4) now becomes a sparse matrix recovery problem. The BS performs this task decentrally. The angular domain channel sparsity can also be used for reducing the training overhead T [29]. For an N -antenna BS, $T \ll N$ pilots are required to recover the sparse matrices $\mathbf{X}_k, \forall k$ [29].

III. PROPOSED VBL-BASED ALGORITHM

We now develop a centralized Bayesian learning framework for a FDD mMIMO BS to estimate the sparse channel matrix \mathbf{X}_k for all K users. The Bayesian framework incorporates a prior belief on \mathbf{X}_k , which should be chosen to promote its inherent sparsity. Cheng *et al.* [3] recently used a GM prior, which is a mixture of two zero-mean Gaussian distributions, to capture common and individual user sparsities in \mathbf{X}_k . This work, however, considered a centralized BS wherein a single PU estimates $\mathbf{X}_k, \forall k$ users. It also assumed that a centralized unit CCU receives and directly processes the compressed observations \mathbf{Y}_k received from all K users. These high-resolution observations, as discussed in [19] and in Section IV, clog up the interconnections in a D-VBL architecture. To circumvent this problem, we assume that the CCU first *quantizes* the observations \mathbf{Y}_k and then sends them to each PU, which separately estimates the channels of a user group by employing the proposed D-VBL algorithm. To facilitate

²We use bold superscripts ‘ \mathbf{k} ’, ‘ ℓ ’ and ‘ \mathbf{c} ’ to represent the user-specific, cluster-specific and common parameters, respectively.

our D-VBL design, we first propose a centralized C-VBL algorithm, which specifically groups the users to exploit their shared sparsity using the quantized observations, a design aspect [3] did not consider. We assume a GM prior of three Gaussian distributions to capture user-specific, cluster-specific and common sparsities. We also note that references [3], [17] have not developed a decentralized VBL algorithm. Before discussing the GM prior, we briefly discuss the Bayesian learning framework and the expectation maximization (EM) algorithm, which are used by the proposed algorithms.

A. Bayesian learning and EM algorithm

We begin by denoting the observation and the unknown parameter set as \mathcal{Y} and \mathcal{X} , respectively [30]. The unknown parameter \mathcal{X} is assigned a prior distribution $p(\mathcal{X}|\boldsymbol{\theta})$ to incorporate the prior belief. The hyperparameter $\boldsymbol{\theta}$ governing the prior distribution is estimated as $\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} p(\mathcal{Y}|\boldsymbol{\theta})$ [30]. Here $p(\mathcal{Y}|\boldsymbol{\theta})$ is the marginal likelihood of the observation \mathcal{Y} . This maximization does not have a closed form solution even for *simple* cases, where the prior and posterior distributions are conjugates [30]. The iterative EM algorithm is generally used to calculate $\hat{\boldsymbol{\theta}}$ [30]. It treats the unknown parameter \mathcal{X} as a hidden variable, and iteratively calculates its posterior distribution using the likelihood, prior and the hyperparameter estimates. For any set of observations, hidden variables and parameters given by $\{\mathcal{Y}, \mathcal{X}, \boldsymbol{\theta}\}$, the EM algorithm works as follows. The log-likelihood function which we wish to maximize w.r.t. $\boldsymbol{\theta}$ is decomposed as [30]

$$\ln p(\mathcal{Y}; \boldsymbol{\theta}) = F(q, \boldsymbol{\theta}) + KL(q||p), \quad \text{where}$$

$$F(q, \boldsymbol{\theta}) = \int q(\mathcal{X}) \ln \left(\frac{p(\mathcal{Y}, \mathcal{X}; \boldsymbol{\theta})}{q(\mathcal{X})} \right) d\mathcal{X},$$

$$KL(q||p) = - \int q(\mathcal{X}) \ln \left(\frac{p(\mathcal{X}|\mathcal{Y}; \boldsymbol{\theta})}{q(\mathcal{X})} \right) d\mathcal{X}. \quad (7)$$

Here $q(\mathcal{X})$ is an arbitrary distribution on \mathcal{X} . The EM algorithm consist of E and M steps, wherein the E step maximizes the lower bound on log-likelihood, i.e., $F(q, \boldsymbol{\theta})$, iteratively w.r.t. $q(\mathcal{X})$, by fixing $\boldsymbol{\theta}$. This results in $KL(q||p) = 0$, which yields $q(\mathcal{X}) = p(\mathcal{X}|\mathcal{Y}; \boldsymbol{\theta})$ [30]. Note that the E-step equivalently calculates the posterior distribution of the hidden variable \mathcal{X} , given the old value of the hyperparameter $\boldsymbol{\theta}$. In the M-step, the lower bound $F(q, \boldsymbol{\theta})$ is maximized w.r.t. $\boldsymbol{\theta}$, keeping the distribution $q(\mathcal{X})$ constant; its value equals to that obtained in the E-step. The E and M steps are repeated till convergence.

B. Centralized variational Bayesian learning

We now develop the C-VBL algorithm to estimate \mathbf{X}_k for all K users in (4) by exploiting its sparsity. In the centralized framework considered in [3]–[5], the BS processes compressed and unquantized observations $[\mathbf{Y}_1, \dots, \mathbf{Y}_K]$ to recover the overall channel matrix $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_K]$. We, by contrast, assume that the CCU first quantizes the real and imaginary parts of observations \mathbf{Y}_k using a q -bit quantizer $\mathcal{Q}(\cdot, q)$ as $\mathbf{D}_k = \mathcal{Q}(\mathbf{Y}_k, q)$ and then processes them. This, as discussed earlier, is desirable for avoiding communication link bottlenecks in decentralized BS architectures [19]. In the C-VBL algorithm, CCU processes the compressed and quantized observation $\mathbf{D} = [\mathbf{D}_1, \dots, \mathbf{D}_K]$ to recover the overall channel matrix $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_K]$. We assume a GM prior over the unknown channel matrix \mathbf{X} , so that the (n, m) th entry

of \mathbf{X}_k i.e., $\mathbf{X}_k^{[n,m]}$, is generated from a mixture of three Gaussian distributions $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}})^{-1})$, $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{l[n]}^{\ell})^{-1})$ and $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{[n]}^{\mathbf{c}})^{-1})$ with the mixing proportions $\rho_{[n]}^{\mathbf{k}}$, $\rho_{[n]}^{\ell}$ and $\rho_{[n]}^{\mathbf{c}}$ respectively, so that $\rho_{[n]}^{\mathbf{k}} + \rho_{[n]}^{\ell} + \rho_{[n]}^{\mathbf{c}} = 1$. Here $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$, $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ and $\rho_{[n]}^i$ represent the n th entry of the N -dimensional vectors $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$, $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ and $\rho_{[n]}^i$, $\forall i = \mathbf{k}, \ell, \mathbf{c}$, respectively. We see that the hyperparameters $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$, $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$, $\rho_{[n]}^{\mathbf{k}}$, $\rho_{[n]}^{\ell}$ and $\rho_{[n]}^{\mathbf{c}}$ are independent of the column index m . This is because all M columns in \mathbf{X}_k share the row sparsity due to rich user side scattering sparsity, given by Ω_k^{ℓ} in (5). The joint column sparsity reduces the estimation error, if appropriately harnessed [24]. Our algorithm appropriately harnesses this row sparsity also, and as shown in Section VI yields better performance than the existing ones.

The first distribution captures the user-specific sparsity of the k th user; its precision hyperparameter $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$ is thus different for all K , but remains the same for all M columns. The second distribution captures the cluster-specific sparsity, and consequently $\boldsymbol{\alpha}_{l[n]}^{\ell}$ is same for all users $k \in \mathcal{C}_l$ of a cluster and for all M columns. The third distribution captures the common sparsity, and thus $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ is the same for all k and m . The prior distribution over channel \mathbf{X} can now be written as

$$P(\mathbf{X} | \boldsymbol{\alpha}_{k[n]}^{\mathbf{k}} \forall k, \boldsymbol{\alpha}_{l[n]}^{\ell} \forall l, \boldsymbol{\alpha}_{[n]}^{\mathbf{c}}, \rho_{[n]}^{\mathbf{k}}, \rho_{[n]}^{\ell}, \rho_{[n]}^{\mathbf{c}}) = \prod_{l=1}^L \prod_{k \in \mathcal{C}_l} \prod_{n=1}^N \prod_{m=1}^M$$

$$P(\mathbf{X}_k^{[n,m]} | \boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}, \boldsymbol{\alpha}_{l[n]}^{\ell}, \boldsymbol{\alpha}_{[n]}^{\mathbf{c}}, \rho_{[n]}^{\mathbf{k}}, \rho_{[n]}^{\ell}, \rho_{[n]}^{\mathbf{c}}), \quad (8)$$

where $P(\mathbf{X}_k^{[n,m]} | \boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}, \boldsymbol{\alpha}_{l[n]}^{\ell}, \boldsymbol{\alpha}_{[n]}^{\mathbf{c}}, \rho_{[n]}^{\mathbf{k}}, \rho_{[n]}^{\ell}, \rho_{[n]}^{\mathbf{c}}) =$

$$\rho_{[n]}^{\mathbf{k}} \mathcal{N}_C(\mathbf{X}_k^{[n,m]} | 0, (\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}})^{-1}) + \rho_{[n]}^{\ell} \mathcal{N}_C(\mathbf{X}_k^{[n,m]} | 0, (\boldsymbol{\alpha}_{l[n]}^{\ell})^{-1}) + \rho_{[n]}^{\mathbf{c}} \mathcal{N}_C(\mathbf{X}_k^{[n,m]} | 0, (\boldsymbol{\alpha}_{[n]}^{\mathbf{c}})^{-1}), \forall k \in \mathcal{C}_l, \forall m, n, l. \quad (9)$$

We define a multinoulli vector $\mathbf{z}_n = [\mathbf{z}_{[n]}^{\mathbf{k}}; \mathbf{z}_{[n]}^{\ell}; \mathbf{z}_{[n]}^{\mathbf{c}}] \in \{0, 1\}^{3 \times 1}$ such that $p(\mathbf{z}_n = [1; 0; 0]) = \rho_{[n]}^{\mathbf{k}}$, $p(\mathbf{z}_n = [0; 1; 0]) = \rho_{[n]}^{\ell}$ and $p(\mathbf{z}_n = [0; 0; 1]) = \rho_{[n]}^{\mathbf{c}}$. The GM prior in (9) can be rewritten, in terms of the components of the vector \mathbf{z}_n as follows

$$P(\mathbf{X}_k^{[n,m]} | \boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}, \boldsymbol{\alpha}_{l[n]}^{\ell}, \boldsymbol{\alpha}_{[n]}^{\mathbf{c}}, \mathbf{z}_{[n]}^{\mathbf{k}}, \mathbf{z}_{[n]}^{\ell}, \mathbf{z}_{[n]}^{\mathbf{c}}) = \mathcal{N}_C(\mathbf{X}_k^{[n,m]} | 0, (\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}})^{-1})^{\mathbf{z}_{[n]}^{\mathbf{k}}} \mathcal{N}_C(\mathbf{X}_k^{[n,m]} | 0, (\boldsymbol{\alpha}_{l[n]}^{\ell})^{-1})^{\mathbf{z}_{[n]}^{\ell}} \mathcal{N}_C(\mathbf{X}_k^{[n,m]} | 0, (\boldsymbol{\alpha}_{[n]}^{\mathbf{c}})^{-1})^{\mathbf{z}_{[n]}^{\mathbf{c}}}. \quad (10)$$

We see from (10) that the precisions $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$, $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ and the multinoulli variables $\mathbf{z}_{[n]}^{\mathbf{k}}$, $\mathbf{z}_{[n]}^{\ell}$, $\mathbf{z}_{[n]}^{\mathbf{c}}$ control the entries of the n th row of \mathbf{X}_k i.e., $\mathbf{X}_k^{[n,:]}$. The elements of $\mathbf{X}_k^{[n,:]}$ become zero when $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$ and $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ approach infinity, regardless of the values $\mathbf{z}_{[n]}^{\mathbf{k}}$, $\mathbf{z}_{[n]}^{\ell}$ and $\mathbf{z}_{[n]}^{\mathbf{c}}$. If $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$, $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ are small, the elements of $\mathbf{X}_k^{[n,:]}$ are non-zero. The values of $\mathbf{z}_{[n]}^{\mathbf{k}}$, $\mathbf{z}_{[n]}^{\ell}$ and $\mathbf{z}_{[n]}^{\mathbf{c}}$ thus show whether the n th index satisfies i) $n \in \Omega_k^{\mathbf{k}}$ – user specific sparsity; ii) $n \in \Omega_l^{\ell}$ – cluster-specific sparsity or; iii) $n \in \Omega^{\mathbf{c}}$ – common sparsity, giving rise to the following cases:

- If $\mathbf{z}_n = [\mathbf{z}_{[n]}^{\mathbf{k}}; \mathbf{z}_{[n]}^{\ell}; \mathbf{z}_{[n]}^{\mathbf{c}}] = [1; 0; 0]$, then the elements of $\mathbf{X}_k^{[n,:]}$ are only determined by $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, which implies that $n \in \Omega_k^{\mathbf{k}}$, i.e., the n th index corresponds to the user-specific sparsity.
- If $\mathbf{z}_n = [0; 1; 0]$, the rows $\mathbf{X}_k^{[n,:]}$ and $\mathbf{X}_{k'}^{[n,:]}$, $\forall k', k \in \mathcal{C}_l$, are

coupled via $\boldsymbol{\alpha}_{l[n]}^{\ell}$, which implies that $n \in \Omega_l^{\ell}$, i.e., the n th index corresponds to the cluster-specific sparsity.

- If $\mathbf{z}_n = [0; 0; 1]$, the elements of $\mathbf{X}_k^{[n,:]}$ and $\mathbf{X}_{k'}^{[n,:]}$ have the same sparsity $\forall k, k'$. This implies that $n \in \Omega^{\mathbf{c}}$, i.e., the n th index corresponds to the common sparsity.

For the centralized Bayesian inference having the proposed GM prior, we have \mathbf{D} as observations, $\mathbf{Y} = [\mathbf{Y}_1, \dots, \mathbf{Y}_K]$, $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_K]$ and $\{\mathbf{z}^{\mathbf{k}}, \mathbf{z}^{\ell}, \mathbf{z}^{\mathbf{c}}\}$ as hidden variables and $\{\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}, \forall k, \boldsymbol{\alpha}_{l[n]}^{\ell}, \forall l, \boldsymbol{\alpha}_{[n]}^{\mathbf{c}}\}$ as hyperparameters. The likelihood of the received data \mathbf{D} is given as

$$p(\mathbf{D} | \mathbf{Y}) = \prod_{k=1}^K \prod_{t=1}^T \prod_{m=1}^M \mathbb{I}(\mathbf{Y}_k^{[t,m]} \in (y_l, y_u]) = \prod_{k=1}^K \prod_{m=1}^M \mathbb{I}(\mathbf{Y}_k^{[:,m]} \in (\mathbf{y}_l, \mathbf{y}_u]). \quad (11)$$

Here $\mathbb{I}(\cdot)$ is the indicator function. The notation $\mathbf{Y}_k^{[t,m]} \in (y_l, y_u]$ represents that its real part $\Re(\mathbf{Y}_k^{[t,m]})$ and imaginary part $\Im(\mathbf{Y}_k^{[t,m]})$ lie in the range $(\Im(y_l), \Im(y_u])$ and $(\Re(y_l), \Re(y_u])$, respectively. Furthermore, $y_l \in \{u_1, \dots, u_B\}$ and $y_u \in \{u_2, \dots, u_{B+1}\}$ represent the range in which the hidden variable $\mathbf{Y}_k^{[t,m]}$ lies, with u_b for $b = 1, \dots, B+1$ being the number of quantization levels of the quantizer $\mathcal{Q}(\cdot, d)$ and $B = 2^d$. For example, given the quantizer output $\mathbf{D}_k^{[t,m]} = v_b$, the hidden observation $\mathbf{Y}_k^{[t,m]}$ has a value between $y_l = u_b$ and $y_u = u_{b+1}$, with probability one. According to the Bayesian learning framework discussed earlier, the quantized observations \mathbf{D}_k are assumed to be generated from the hidden unquantized data \mathbf{Y}_k , which are generated from the hidden channel \mathbf{X}_k , which in turn are generated according to the hidden multinoulli variable \mathbf{z} and hyperparameters $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, $\boldsymbol{\alpha}_{l[n]}^{\ell}$ and $\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$. We recall that the E-step of the EM algorithm computes the posterior distribution of the hidden variables, which for the GM prior in (10) and Dirac-delta likelihood in (11) is intractable due to their non-conjugacy [30]. We use variational Bayesian learning (VBL) [30] which assumes that in the E-step, the joint posterior distribution of the hidden variable set $\mathcal{X} \triangleq \{\mathbf{Y}, \mathbf{X}, \mathbf{z}_n, \forall n\}$ can be written as the product of component-wise distributions, i.e., $q(\mathcal{X}) =$

$$\left(\prod_{k=1}^K \prod_{m=1}^M q(\mathbf{Y}_k^{[:,m]}) q(\mathbf{X}_k^{[:,m]}) \right) \left(\prod_{n=1}^N q(\mathbf{z}_{[n]}^{\mathbf{k}}) q(\mathbf{z}_{[n]}^{\ell}) q(\mathbf{z}_{[n]}^{\mathbf{c}}) \right). \quad (12)$$

Here $\mathbf{Y}_k^{[:,m]} \in \mathbb{C}^{T \times 1}$, $\mathbf{X}_k^{[:,m]} \in \mathbb{C}^{N \times 1}$ are the m th columns of \mathbf{Y}_k and \mathbf{X}_k , respectively. Here $q(\cdot)$ denotes the estimated posterior distribution of the argument. We now simplify E and M steps.

E-step: Updating posterior of hidden variables:

- 1) *Posterior of $\{\mathbf{Y}_k^{[:,m]}\}$:* The posterior $q(\mathbf{Y}_k^{[:,m]})$, of the hidden variables \mathbf{Y} , is given as

$$q(\mathbf{Y}_k^{[:,m]}) \stackrel{(a)}{=} \exp\left(\langle \ln p(\mathcal{Y}, \mathcal{X} | \boldsymbol{\theta}) \rangle_{q(\mathcal{X}) \setminus q(\mathbf{Y}_k^{[:,m]})} + \bar{\mathcal{C}}\right) \propto p(\mathbf{D}_k^{[:,m]} | \mathbf{Y}_k^{[:,m]}) \exp\left(\left\langle \ln p(\mathbf{Y}_k^{[:,m]} | \mathbf{X}_k^{[:,m]}, \sigma_k^2) \right\rangle_{q(\mathbf{X}_k^{[:,m]})}\right) \propto \mathbb{I}(\mathbf{Y}_k^{[:,m]} \in (\mathbf{y}_l, \mathbf{y}_u]) \exp\left(\|\mathbf{Y}_k^{[:,m]} - \Phi(\mathbf{X}_k^{[:,m]})\|^2 / \sigma_k^2\right). \quad (13)$$

Equality (a) is due to the variational approximation in (12), where the subscript $q(\mathcal{X}) \setminus q(\mathbf{Y}_k^{[:,m]})$ denotes the distribu-

tion $q(\mathcal{X})$ with $q(\mathbf{Y}_k^{[:,m]})$ marginalized out. Equality (b) is obtained by substituting $\mathcal{Y} = \{\mathbf{D}\}$, $\mathcal{X} = \{\mathbf{Y}, \mathbf{X}, \mathbf{z}\}$, $p(\mathbf{D}, \mathbf{Y}, \mathbf{X}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{D}|\mathbf{Y})p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{X}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})$, and retaining the terms dependent on $\mathbf{Y}_k^{[:,m]}$. In (c), we use (11) and the fact that the distribution $p(\mathbf{Y}_k|\mathbf{X}_k)$ is Gaussian, as the entries of the noise matrix \mathbf{E}_k are i.i.d., zero mean with variance σ_k^2 . We see that the estimated posterior distribution in (13) is a truncated Gaussian distribution between the lower and upper limits given by \mathbf{y}_l and \mathbf{y}_u , respectively. The posterior mean, which is the mean of a complex truncated Gaussian distribution is given as [18, Eq. (13.134)]

$$\langle \mathbf{Y}_k^{[t,m]} \rangle = \langle \Re\{\mathbf{Y}_k^{[t,m]}\} \rangle + \langle \Im\{\mathbf{Y}_k^{[t,m]}\} \rangle, \text{ where, } (14)$$

$$\langle \Re\{\mathbf{Y}_k^{[t,m]}\} \rangle = \Re\{\langle \Phi(\mathbf{X}_k^{[:,m]}) \rangle_t\} - \frac{\sigma_k}{\sqrt{2}} \frac{f(b_R) - f(a_R)}{F(b_R) - F(a_R)} (15)$$

$$\langle \Im\{\mathbf{Y}_k^{[t,m]}\} \rangle = \Im\{\langle \Phi(\mathbf{X}_k^{[:,m]}) \rangle_t\} - \frac{\sigma_k}{\sqrt{2}} \frac{f(b_I) - f(a_I)}{F(b_I) - F(a_I)}. (16)$$

Here we define $a_R = \sqrt{2}(\Re\{y_l\} - \Re\{\langle \Phi(\mathbf{X}_k^{[:,m]}) \rangle_t\})/\sigma_k$, $b_R = \sqrt{2}(\Re\{y_u\} - \Re\{\langle \Phi(\mathbf{X}_k^{[:,m]}) \rangle_t\})/\sigma_k$, $a_I = \sqrt{2}(\Im\{y_l\} - \Im\{\langle \Phi(\mathbf{X}_k^{[:,m]}) \rangle_t\})/\sigma_k$, $b_I = \sqrt{2}(\Im\{y_u\} - \Im\{\langle \Phi(\mathbf{X}_k^{[:,m]}) \rangle_t\})/\sigma_k$. The quantities $f(a)$ and $F(a)$ denote the probability and cumulative density function, respectively, of a standard normal Gaussian distribution at the point $a \in \mathbb{C}$. The notation $[\Phi(\mathbf{X}_k^{[:,m]})]_t$ represents the t th entry of the vector $\Phi(\mathbf{X}_k^{[:,m]}) \in \mathbb{C}^{T \times 1}$.

- 2) Posterior of $\{\mathbf{X}_k^{[:,m]}\}$: The posterior distribution of $\mathbf{X}_k^{[:,m]}$ is approximated as complex Gaussian as derived in Appendix A, and can be written as [30]:

$$p(\mathbf{X}_k^{[:,m]}|\boldsymbol{\alpha}_k^k, \boldsymbol{\alpha}_l^\ell, \boldsymbol{\alpha}^c, \langle \mathbf{z}^k \rangle, \langle \mathbf{z}^\ell \rangle, \langle \mathbf{z}^c \rangle, \langle \mathbf{Y}_k^{[:,m]} \rangle) = \mathcal{N}_{\mathbb{C}}(\mathbf{X}_k^{[:,m]}|\boldsymbol{\mu}_k^{[:,m]}, \boldsymbol{\Sigma}_k), \forall k \in \mathcal{C}_l, l, m, \text{ where} (17)$$

$$\boldsymbol{\mu}_k^{[:,m]} = \sigma_k^{-2} \boldsymbol{\Sigma}_k \Phi^H \langle \mathbf{Y}_k^{[:,m]} \rangle \in \mathbb{C}^{N \times 1}, (18)$$

$$\boldsymbol{\Sigma}_k = (\boldsymbol{\Lambda}_k + \sigma_k^{-2} \Phi^H \Phi)^{-1} \in \mathbb{C}^{N \times N}, (19)$$

$\boldsymbol{\Lambda}_k = \text{diag}(\langle \mathbf{z}^k \rangle \circ \boldsymbol{\alpha}_k^k + \langle \mathbf{z}^\ell \rangle \circ \boldsymbol{\alpha}_l^\ell + \langle \mathbf{z}^c \rangle \circ \boldsymbol{\alpha}^c) \in \mathbb{C}^{N \times N}$. The notations \circ and $\langle f(\mathbf{z}) \rangle$ represent the Hadamard product and the expectation of $f(\mathbf{z})$ with respect to the posterior distribution of \mathbf{z} , respectively.

- 3) Posterior of $\{\mathbf{z}^k, \mathbf{z}^\ell, \mathbf{z}^c\}$: The posterior of $\mathbf{z}_n = [\mathbf{z}_n^k, \mathbf{z}_n^\ell, \mathbf{z}_n^c]$, given by $p(\mathbf{z}_n|\boldsymbol{\eta}_{[n]}^k, \boldsymbol{\eta}_{[n]}^\ell, \boldsymbol{\eta}_{[n]}^c)$ and derived in Appendix A, is a multinoulli distribution with mean [30]

$$\langle \mathbf{z}_n^k \rangle = (1 + \exp(\boldsymbol{\eta}_{[n]}^c - \boldsymbol{\eta}_{[n]}^k) + \exp(\boldsymbol{\eta}_{[n]}^\ell - \boldsymbol{\eta}_{[n]}^k))^{-1}, (20)$$

$$\langle \mathbf{z}_n^\ell \rangle = (1 + \exp(\boldsymbol{\eta}_{[n]}^c - \boldsymbol{\eta}_{[n]}^\ell) + \exp(\boldsymbol{\eta}_{[n]}^k - \boldsymbol{\eta}_{[n]}^\ell))^{-1}, (21)$$

$$\langle \mathbf{z}_n^c \rangle = (1 + \exp(\boldsymbol{\eta}_{[n]}^k - \boldsymbol{\eta}_{[n]}^c) + \exp(\boldsymbol{\eta}_{[n]}^\ell - \boldsymbol{\eta}_{[n]}^c))^{-1}, (22)$$

$$\text{where, } \boldsymbol{\eta}_{[n]}^k = M \sum_{k=1}^K \ln \boldsymbol{\alpha}_{k[n]}^k, (23)$$

$$\boldsymbol{\eta}_{[n]}^\ell = M \sum_{l=1}^L |\mathcal{C}_l| \ln \boldsymbol{\alpha}_{l[n]}^\ell, \text{ and } \boldsymbol{\eta}_{[n]}^c = KM \ln \boldsymbol{\alpha}_{[n]}^c. (24)$$

We note that (20)-(22) satisfy $\langle \mathbf{z}_n^k \rangle + \langle \mathbf{z}_n^\ell \rangle + \langle \mathbf{z}_n^c \rangle = 1$.

M Step: Updating the hyperparameters: The updates of

$$\boldsymbol{\alpha}_{k[n]}^k, \boldsymbol{\alpha}_{l[n]}^\ell, \boldsymbol{\alpha}_{[n]}^c, \text{ derived in Appendix A, are given by [3]} (25)$$

$$\boldsymbol{\alpha}_{k[n]}^k = M / \sum_{l=1}^M \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle, \boldsymbol{\alpha}_{l[n]}^\ell = |\mathcal{C}_l| / \sum_{k \in \mathcal{C}_l} 1 / \boldsymbol{\alpha}_{k[n]}^k, (25)$$

$$\boldsymbol{\alpha}_{[n]}^c = K / \sum_{l=1}^L \sum_{k \in \mathcal{C}_l} 1 / \boldsymbol{\alpha}_{k[n]}^k, (26)$$

where $\langle |\mathbf{X}_k^{[n,m]}|^2 \rangle = |\boldsymbol{\mu}_k^{[n,m]}|^2 + \boldsymbol{\Sigma}_k^{[n,m]}$. The C-VBL design, summarized in Fig. 2, in the E-step evaluates the posterior distributions, and in the M-step updates the hyperparameters.

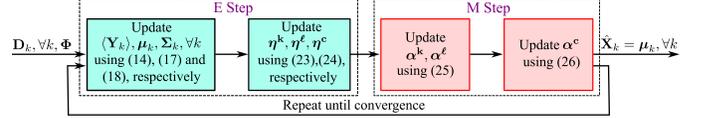


Fig. 2: Schematic of C-VBL algorithm at the CCU at the BS.

Remark 2. The centralized prior for C-VBL is designed in Eq. (10) to capture the sparsity of each cluster, and it is therefore cluster-specific. The C-VBL updates are, however, not cluster-specific. For example, the updates of the parameters $\boldsymbol{\eta}^k, \boldsymbol{\eta}^\ell$ and $\boldsymbol{\eta}^c$ (Eqs. (23)-(24)) in the E step, and; the hyperparameter $\boldsymbol{\alpha}^c$ (Eq. (26)) in the M step. These updates require summation over all $l = 1, \dots, L$ clusters, and thus are global updates. Their update, thus, requires joint processing of the observations $[\mathbf{D}_1, \dots, \mathbf{D}_K]$. *The C-VBL algorithm is thus applied jointly to all the clusters for exploiting the structured sparsity of the mMIMO channel.*

IV. PROPOSED DECENTRALIZED ALGORITHM

The BS in the C-VBL algorithm centrally processes $[\mathbf{D}_1, \dots, \mathbf{D}_K]$ to estimate channel \mathbf{X}_k of K users. We now propose our D-VBL algorithm for a decentralized BS having L PUs and a CCU. The PUs exploit user-specific, cluster-specific and common sparsities for improving the channel estimation accuracy by exchanging messages. We design novel messages and global variable updates assuming the centralized prior in (10), and exploit the aforementioned sparsities. The D-VBL algorithm limits the messages exchanged using a SNR-based criterion. The VEM algorithm of [3] also uses a centralized prior. The novel D-VBL messages, the updates and their convergence, which we design and discuss in this section to enable decentralized implementation, are the main contributions of this work over the VEM algorithm.

The BS, as in [26], groups the users having similar channel covariance matrices, which are assumed to be known [27]. It then assigns their observations to the PUs which use the D-VBL algorithm for decentrally estimating their channels. We note that the AG-SBL algorithm of [17] adaptively groups the users to exploit the channel's sparsity. The decentralized BS architecture, by contrast, requires apriori user grouping. This renders the extension of the AG-SBL algorithm to a decentralized architecture highly non-trivial. The existing literature, to the best of our knowledge, has not investigated decentralized algorithms based on adaptive-grouping.

The quantized observation vector $\mathbf{D}_{C_l} \in \mathbb{C}^{T \times M|\mathcal{C}_l|}$ at the l th PU is given by

$$\mathbf{D}_{C_l} = \mathcal{Q}(\mathbf{Y}_{C_l}, d) = \mathcal{Q}(\Phi \mathbf{X}_{C_l} + \mathbf{E}_{C_l}, d). (27)$$

Here the matrices $\mathbf{Y}_{C_l}, \mathbf{E}_{C_l} \in \mathbb{C}^{T \times M|\mathcal{C}_l|}$ and $\mathbf{X}_{C_l} \in \mathbb{C}^{N \times M|\mathcal{C}_l|}$ are obtained by column-wise stacking of $\mathbf{Y}_k, \mathbf{E}_k$ and \mathbf{X}_k

for all $k \in \mathcal{C}_l$ cluster, respectively. The matrix $\mathbf{X}_{\mathcal{C}_l}$ is the unknown channel of the user cluster \mathcal{C}_l which exhibits user-specific, cluster-specific and common sparsities, and is referred to as the *cluster channel matrix*. The matrix $\mathbf{E}_{\mathcal{C}_l}$ is the overall noise matrix of the l th cluster. The clustered quantized observations $\mathbf{D}_{\mathcal{C}_l}$ of all L clusters, are processed individually at the respective PU, which communicates via CCU to exploit the sparsity of the unknown channel. The l th PU updates its parameters using the messages received from other PUs. We call the parameters, whose updates in the C-VBL algorithm (do not) depend on the parameters of other PUs as the (*local*) *global* parameters. This implies that the parameter updates involving summation over all l PUs are global, and the rest are local. We next list them in Table II.

We note that the C-VBL algorithm in the i) E Step, updates the local parameters $\langle \mathbf{Y}_k^{[:,m]} \rangle$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $\forall k$ users employing (14), (17) and (18), respectively, and the global parameters $\boldsymbol{\eta}^k$, $\boldsymbol{\eta}^\ell$ and $\boldsymbol{\eta}^c$ using (23), (24); and ii) M Step, updates the local parameters $\boldsymbol{\alpha}_k^k$, $\forall k$ users, $\boldsymbol{\alpha}_l^\ell$, $\forall l$ clusters using (25), and global parameter $\boldsymbol{\alpha}^c$ using (26). In the proposed D-VBL algorithm, the l th PU, however, estimates the cluster channel matrix $\mathbf{X}_{\mathcal{C}_l}$ locally using the messages received from other PUs. It thus calculates local estimates of the global parameters $\boldsymbol{\eta}^k$, $\boldsymbol{\eta}^\ell$, $\boldsymbol{\eta}^c$ and $\boldsymbol{\alpha}^c$ denoted as $\tilde{\boldsymbol{\eta}}^{k,l}$, $\tilde{\boldsymbol{\eta}}^{\ell,l}$, $\tilde{\boldsymbol{\eta}}^{c,l}$ and $\tilde{\boldsymbol{\alpha}}^{c,l}$, respectively. To differentiate between local and global updates in both the E and M Steps of the D-VBL algorithm, we divide the E Step into E1 and E2 steps and M Step into M1 and M2 steps. The E1 and M1 steps correspond to the local E and M steps, which update the local parameters $\langle \mathbf{Y}_k^{[:,m]} \rangle$, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\alpha}_k^k$, $\boldsymbol{\alpha}_l^\ell$, respectively. The E2 and M2 steps correspond to the global parameter updates. We now provide a brief outline of the proposed D-VBL algorithm wherein the l th PU for all users $k \in \mathcal{C}_l$ cluster

- **E1:** estimates local parameters $\langle \mathbf{Y}_k \rangle$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ using (14), (17) and (18), respectively, for user $k \in \mathcal{C}_l$ cluster.
- **E2:** estimates, similar to (20)-(22), global parameters $\tilde{\boldsymbol{\eta}}^{k,l}$, $\tilde{\boldsymbol{\eta}}^{\ell,l}$, $\tilde{\boldsymbol{\eta}}^{c,l}$ and the posterior means $\langle \tilde{\mathbf{z}}^{k,l} \rangle$, $\langle \tilde{\mathbf{z}}^{\ell,l} \rangle$, $\langle \tilde{\mathbf{z}}^{c,l} \rangle$,³ using the information received from all other PUs, in the previous iteration.
- **M1:** updates local hyperparameters $\boldsymbol{\alpha}_k^k$, for all users $k \in \mathcal{C}_l$ cluster, and $\boldsymbol{\alpha}_l^\ell$ using (25).
- **C:** shares PU-specific local information with CCU, which broadcasts this to all other PUs.
- **M2:** updates the global hyperparameter $\tilde{\boldsymbol{\alpha}}^{c,l}$, as derived later in (34), using information received from all other PUs.

We see that the updates of local parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, $\boldsymbol{\alpha}_k^k$, $\forall k \in \mathcal{C}_l$ cluster, and $\boldsymbol{\alpha}_l^\ell$ in the **E1** and **M1** steps of the D-VBL algorithm are same as that of the C-VBL algorithm. The respective PUs, therefore, update them without communicating with other PUs. The l th PU, to update global parameters $\tilde{\boldsymbol{\eta}}^{k,l}$, $\tilde{\boldsymbol{\eta}}^{\ell,l}$, $\tilde{\boldsymbol{\eta}}^{c,l}$ and $\tilde{\boldsymbol{\alpha}}^{c,l}$ in the **E2** and **M2** steps, however, as seen from (23), (24) and (26), requires information from all other PUs. This information is given by the hyperparameters $\boldsymbol{\alpha}_{k[n]}^k$, for all $k \in \mathcal{C}_l$ cluster

³The local estimate of a global parameter at the l th PU for the D-VBL algorithm is denoted by a tilde and a superscript l over it. Notation for the set of local parameters $\{\langle \mathbf{Y}_k \rangle, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\alpha}_k^k, \boldsymbol{\alpha}_l^\ell\}$, however, remains same.

and $\forall l$ clusters. This is because all other parameters can be evaluated using $\boldsymbol{\alpha}_{k[n]}^k$. The D-VBL algorithm therefore uses a C step (communication step) to enable this exchange which, as shown later, will enhance the estimation accuracy. We next discuss how to exploit the sparsity of \mathbf{X}_k to limit the information exchange between the PUs. Based on this discussion, we will later derive the **E2** and **M2** steps.

A. Discussion of the sparsity structure of \mathbf{X}_k to limit the message exchange in C Step

We now consider four possible sparsity structures of $\mathbf{X}_k^{[n,:]}$, and show that the message exchange is required only for one particular case.

Case 1 (user-specific sparsity, $n \in \Omega_k^k$): For this case, the users are not coupled, which implies that the elements $\mathbf{X}_k^{[n,m]}$, $\forall m$ columns in (10), are generated from $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{k[n]}^k)^{-1})$ only, i.e., $\mathbf{z}_{[n]}^k = 1$, $\mathbf{z}_{[n]}^\ell = \mathbf{z}_{[n]}^c = 0$. The l th PU thus estimates $\mathbf{X}_k^{[n,:]}$, $\forall k \in \mathcal{C}_l$ cluster, by iterating (14), (17)-(18) and the first equation in (25). This does not require message exchange between PUs, as the updates are independent of the summation over the cluster index l .

Case 2 (cluster-specific sparsity, $n \in \Omega_l^\ell$): For this case, the user clusters are not coupled. The users belonging to the same cluster are, however, coupled due to their similar scattering environment at the BS side. We thus observe that $\mathbf{X}_k^{[n,m]}$, $\forall m$ columns, in (10) are generated from a mixture of $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{k[n]}^k)^{-1})$ and $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{l[n]}^\ell)^{-1})$, i.e., $\mathbf{z}_{[n]}^k + \mathbf{z}_{[n]}^\ell = 1$, $\mathbf{z}_{[n]}^c = 0$. The l th PU thus estimates $\mathbf{X}_k^{[n,:]}$, for all users $k \in \mathcal{C}_l$ cluster, by updating the

(i) local parameters $\langle \mathbf{Y}_k \rangle$, $\boldsymbol{\mu}_k^{[:,m]}$ and $\boldsymbol{\Sigma}_k$ in the E step using (14), (17) and (18), respectively, where the $[n, n]$ th entry of the prior precision $\boldsymbol{\Lambda}_k$ in (19) becomes: $\boldsymbol{\Lambda}_k^{[n,n]} = \langle \tilde{\mathbf{z}}_{[n]}^{k,l} \rangle \boldsymbol{\alpha}_{k[n]}^k + \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle \boldsymbol{\alpha}_{l[n]}^\ell$.

(ii) parameters $\langle \tilde{\mathbf{z}}^{k,l} \rangle$, $\langle \tilde{\mathbf{z}}^{\ell,l} \rangle$ in the E step as $\langle \tilde{\mathbf{z}}_{[n]}^{k,l} \rangle = (1 + \exp(\tilde{\boldsymbol{\eta}}_{[n]}^{\ell,l} - \tilde{\boldsymbol{\eta}}_{[n]}^{k,l}))^{-1}$, and $\langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle = (1 + \exp(\tilde{\boldsymbol{\eta}}_{[n]}^{k,l} - \tilde{\boldsymbol{\eta}}_{[n]}^{\ell,l}))^{-1}$, where, $\tilde{\boldsymbol{\eta}}_{[n]}^{k,l} = M \sum_{k \in \mathcal{C}_l} \ln \boldsymbol{\alpha}_{k[n]}^k$, $\tilde{\boldsymbol{\eta}}_{[n]}^{\ell,l} = M \sum_{k \in \mathcal{C}_l} \ln \boldsymbol{\alpha}_{l[n]}^\ell$.

(iii) local hyperparameters $\boldsymbol{\alpha}_{k[n]}^k$, $\boldsymbol{\alpha}_{l[n]}^\ell$ in the M step using (25). We note that in absence of coupling between different user clusters, to estimate $\mathbf{X}_k^{[n,:]}$, for all users $k \in \mathcal{C}_l$, the l th PU does not have to exchange messages with other PUs.

Case 3 (all zeros, $n \notin \Omega_k^k$): For this case, the elements $\mathbf{X}_k^{[n,m]}$ are not coupled with each other and $\boldsymbol{\alpha}_{k[n]}^k$, $\boldsymbol{\alpha}_{l[n]}^\ell$, $\boldsymbol{\alpha}_{[n]}^c \rightarrow \infty$ in (10). The elements $\mathbf{X}_k^{[n,m]}$ are thus generated from $\mathcal{N}_C(0, (\boldsymbol{\alpha}_{k[n]}^k)^{-1})$, where $\boldsymbol{\alpha}_{k[n]}^k$ takes very high values, for all $k \in \mathcal{C}_l$. The channel $\mathbf{X}_k^{[n,:]}$, $\forall k \in \mathcal{C}_l$ can be efficiently estimated at the l th PU, without any message exchange, similar to Case-1.

Case 4 (common sparsity, $n \in \Omega^c$): The elements of $\mathbf{X}_k^{[n,:]}$ for different users are coupled, for which the GM prior in (10) and the updates given by (14)-(26) are applicable. We see from (23), (24) and (26) that to update the global hyperparameters $\tilde{\boldsymbol{\eta}}^{k,l}$, $\tilde{\boldsymbol{\eta}}^{\ell,l}$, $\tilde{\boldsymbol{\eta}}^{c,l}$ and $\tilde{\boldsymbol{\alpha}}^{c,l}$, the l th PU requires message exchange between the PUs in this case, as they involve summation over the cluster index l .

Table II: Parameters classified as global and local for the proposed D-VBL algorithm.

Step	Parameter	Type	Step	Parameter	Type
E-Step	$\langle \mathbf{Y}_k^{[:,m]} \rangle$ in (14), $\boldsymbol{\mu}_k^{[:,m]}$ in (17), $\boldsymbol{\Sigma}_k$ in (18)	Local	M-step	$\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}, \boldsymbol{\alpha}_{l[n]}^{\mathbf{l}}$ in (25)	Local
	$\boldsymbol{\eta}_{[n]}^{\mathbf{k}}$ in (23), $\boldsymbol{\eta}_{[n]}^{\mathbf{l}}, \boldsymbol{\eta}_{[n]}^{\mathbf{c}}$ in (24)	Global		$\boldsymbol{\alpha}_{[n]}^{\mathbf{c}}$ in (26)	Global

We see that the message exchange is only required for Case-4, and to enable it, we have to determine whether the index n corresponds to Case-4 (the common support $\Omega^{\mathbf{c}}$). To this end, we introduce three binary-valued vectors related to the non-zero support of $\mathbf{X}_k, \forall k$ users:

1) **User-specific binary support vector**, denoted as $\mathbf{s}_k^{\mathbf{k}} \in \{0, 1\}^{N \times 1}$ for the k th user, whose n th entry $s_{k[n]}^{\mathbf{k}}$ represents the true support of the angular domain channel $\mathbf{X}_k^{[n,:]}$, i.e., $\mathbf{s}_k^{\mathbf{k}} = \mathbb{I}(n \in \Omega_k^{\mathbf{k}})$, where $\Omega_k^{\mathbf{k}}$ is given by (6). The l th PU estimates this binary support, which we denote as $\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}}$,⁴ $\forall n$ and $\forall k \in \mathcal{C}_l$ cluster, using the following thresholding rule:

$$\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}} = \mathbb{I}\left(\tilde{\gamma}_{k[n]}^{\mathbf{k},l} > \frac{c}{\text{SNR}_k}\right). \quad (28)$$

Here $\tilde{\gamma}_{k[n]}^{\mathbf{k},l} \triangleq 1/(\langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \boldsymbol{\alpha}_{k[n]}^{\mathbf{k}} + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{l},l} \rangle \boldsymbol{\alpha}_{l[n]}^{\mathbf{l}} + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle \tilde{\boldsymbol{\alpha}}_{[n]}^{\mathbf{c},l})$, $\forall k \in \mathcal{C}_l$ is the prior variance of the sparse row vector $\mathbf{X}_k^{[n,:]}$ from (19). The signal-to-noise ratio for the k th cluster $\text{SNR}_k \triangleq P/\sigma_k^2$, and $c > 1$ is a constant. The threshold chosen is shown to work well in our experiments.

2) **Cluster-specific binary support vector**, denoted as $\mathbf{s}_l^{\mathbf{l}} \in \{0, 1\}^{N \times 1}$ for the l th PU, whose n th entry $s_{l[n]}^{\mathbf{l}}$ indicates the true common support of the l th cluster, i.e., $\mathbf{s}_l^{\mathbf{l}} = \mathbb{I}(n \in \Omega_l^{\mathbf{l}} \cup \Omega^{\mathbf{c}})$. The l th PU estimates this cluster-specific binary variable, which is denoted as $\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}}, \forall n$. We note that for true user-specific binary variables $\mathbf{s}_k^{\mathbf{k}} = 1$, for all $k \in \mathcal{C}_l$, this implies that $s_{l[n]}^{\mathbf{l}} = 1$. This is because $\bigcap_{k \in \mathcal{C}_l} \Omega_k^{\mathbf{k}} = \Omega_l^{\mathbf{l}} \cup \Omega^{\mathbf{c}}$. The true cluster-specific binary variable $s_{l[n]}^{\mathbf{l}}$, thus depends on the user-specific binary variable $\mathbf{s}_k^{\mathbf{k}}$, for all $k \in \mathcal{C}_l$, as follows: $s_{l[n]}^{\mathbf{l}} = 1$, if $\sum_{k \in \mathcal{C}_l} \mathbf{s}_k^{\mathbf{k}} = |\mathcal{C}_l|$. To estimate cluster-specific support, the l th PU applies majority rule over $\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}}, \forall n, k \in \mathcal{C}_l$ as

$$\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}} = \mathbb{I}\left(\sum_{k \in \mathcal{C}_l} \hat{\mathbf{s}}_{k[n]}^{\mathbf{k}} \geq \left\lceil \frac{|\mathcal{C}_l|}{2} \right\rceil\right). \quad (29)$$

In the C-step of the D-VBL algorithm, the l th PU uses this cluster-specific support estimate to share the information with other PUs as follows – the l th PU shares the hyperparameters $\boldsymbol{\alpha}_{k[n]}^{\mathbf{k}}$, of the n th index, for all $k \in \mathcal{C}_l$ cluster, only if its cluster-specific support estimate is $\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}} = 1$. At each iteration, the l th PU thus shares a message $\mathbf{M}_l \in \mathbb{C}^{N \times |\mathcal{C}_l|}$, having the k th column

$$\mathbf{M}_l^{[:,k]} = \boldsymbol{\alpha}_k^{\mathbf{k}} \circ \hat{\mathbf{s}}_l^{\mathbf{l}}, \forall k \in \mathcal{C}_l. \quad (30)$$

Here $\boldsymbol{\alpha}_k^{\mathbf{k}} = [\boldsymbol{\alpha}_{k[1]}^{\mathbf{k}}; \dots; \boldsymbol{\alpha}_{k[N]}^{\mathbf{k}}]$, $\hat{\mathbf{s}}_l^{\mathbf{l}} = [\hat{\mathbf{s}}_{l[1]}^{\mathbf{l}}(1); \dots; \hat{\mathbf{s}}_{l[N]}^{\mathbf{l}}] \in \{0, 1\}^{N \times 1}$, and $\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}}$ is the cluster-specific binary support estimate. The n th entry of the message is thus $\mathbf{M}_l^{[n,k]} = \boldsymbol{\alpha}_{k[n]}^{\mathbf{k}} \hat{\mathbf{s}}_{l[n]}^{\mathbf{l}}$.

3) **Common binary support vector**, denoted as $\mathbf{s}^{\mathbf{c}} \in \{0, 1\}^{N \times 1}$, where each of its entry $s_{[n]}^{\mathbf{c}}$ indicates the true common row support of the channel matrix $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_K]$. Mathematically, $\mathbf{s}_{[n]}^{\mathbf{c}} = \mathbb{I}(n \in \Omega^{\mathbf{c}})$. (31)

The l th PU shares the message \mathbf{M}_l in (30), with the CCU, which broadcasts it to all other PUs. The l th PU, consequently, also receives the messages \mathbf{M}_j , from the CCU, sent by all other PUs $j \neq l$. The received non-zero row $\mathbf{M}_j^{[n,:]}$ of message indicates that the corresponding cluster-specific binary support is $\hat{\mathbf{s}}_{j[n]}^{\mathbf{l}} = 1$. Using this information, the l th PU estimates the common binary support $\mathbf{s}_{[n]}^{\mathbf{c}}$, which we denote as $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$. We know from (29) that the binary variable $\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}} = 1$ is the estimate that the index n either belongs to the cluster-specific support $\Omega_l^{\mathbf{l}}$ (Case-2) or to the common support $\Omega^{\mathbf{c}}$ (Case-4). The l th PU now has to calculate $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$, which becomes one only for common support, i.e., $n \in \Omega^{\mathbf{c}}$, but not for the cluster-specific support, i.e., $n \in \Omega_l^{\mathbf{l}}$. We note that the n th index belongs to the common support $\Omega^{\mathbf{c}}$ if $s_{l[n]}^{\mathbf{l}} = 1$ for all l clusters. Mathematically, $\mathbf{s}_{[n]}^{\mathbf{c}} = 1$ if $\sum_{l=1}^L s_{l[n]}^{\mathbf{l}} = L$. The l th PU, similar to [13], calculates the common support estimate $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$ by applying a majority rule to the cluster-specific support estimates $\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}}, \forall n$, as

$$\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = \mathbb{I}\left(|\hat{\mathcal{S}}_n^l| > \left\lceil \frac{L-1}{2} \right\rceil\right), \text{ where,} \quad (32)$$

$$\hat{\mathcal{S}}_n^l \triangleq \{1 \leq j \leq L: \mathbf{M}_j^{[n,k]} \neq 0, \forall k \in \mathcal{C}_j \text{ and } \hat{\mathbf{s}}_{l[n]}^{\mathbf{l}} = 1\}. \quad (33)$$

When the cluster-specific support estimate is $\hat{\mathbf{s}}_{l[n]}^{\mathbf{l}} = 1$, the set $\hat{\mathcal{S}}_n^l$ contains the following information – the l th index and the cluster indices j for the n th non-zero row of the message matrix \mathbf{M}_j , sent by the j th PU. We note that the set $\hat{\mathcal{S}}_n^l$ is indicative of the common support, and at the l th PU, gives the estimate of the set $\mathcal{S}_n = \begin{cases} \{1, 2, \dots, L\} & \text{if } n \in \Omega^{\mathbf{c}} \\ \phi & \text{otherwise,} \end{cases}$ where ϕ is the null set. Thus, the non-zero indices in $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$ give the estimate of the set corresponding to the Case-4 ($n \in \Omega^{\mathbf{c}}$).

Based on this binary common support estimate $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}, \forall n$, the l th PU computes the global hyperparameters $\tilde{\boldsymbol{\eta}}^{\mathbf{k},l}, \tilde{\boldsymbol{\eta}}^{\mathbf{l},l}, \tilde{\boldsymbol{\eta}}^{\mathbf{c},l}$ and $\tilde{\boldsymbol{\alpha}}^{\mathbf{c},l}$ in the **E2** and **M2** steps, respectively, which we discuss next. Before proceeding further, we summarize the variables and support sets in Table III.

B. Local update of global hyperparameters with binary support estimate

The l th PU now estimates the global hyperparameters $\tilde{\boldsymbol{\eta}}^{\mathbf{k},l}, \tilde{\boldsymbol{\eta}}^{\mathbf{l},l}, \tilde{\boldsymbol{\eta}}^{\mathbf{c},l}$ and $\tilde{\boldsymbol{\alpha}}^{\mathbf{c},l}$ by incorporating the information received from all other PUs, and the binary common support estimate $\hat{\mathbf{s}}^{\mathbf{c},l}$. The C-VBL updates of these hyperparameters are given by (23), (24) and (26), given the hyperparameters $\boldsymbol{\alpha}_k^{\mathbf{k}}, \forall k$. For the D-VBL algorithm, the l th PU, however, has access only to the censored copies of the hyperparameter set $\boldsymbol{\alpha}_k^{\mathbf{k}}, \forall k$, which is given by $\mathbf{M}_j^{[:,k]}, \forall k \in \mathcal{C}_j, j \neq l$, in (30) and $\boldsymbol{\alpha}_k^{\mathbf{k}}, \forall k \in \mathcal{C}_l$. We know from the discussion in Section IV-A that the l th PU for the first three cases can recover the cluster channel matrix $\mathbf{X}_{\mathcal{C}_l}$ efficiently without communicating with other PUs. The l th PU thus needs to incorporate the messages \mathbf{M}_j from other PUs only if $n \in \Omega^{\mathbf{c}}$ or equivalently $\mathbf{s}_{[n]}^{\mathbf{c}} = 1$. The l th PU consequently uses the common support

⁴The estimate of any true support vector is denoted by a hat over it.

Table III: List of D-VBL parameters.

Parameters	Description
$\bar{\mathbf{z}}_{[n]}^{\mathbf{k},l}, \bar{\mathbf{z}}_{[n]}^{\ell,l}$ and $\bar{\mathbf{z}}_{[n]}^{\mathbf{c},l}$	Local estimates of variables $\mathbf{z}_{[n]}^{\mathbf{k}}, \mathbf{z}_{[n]}^{\ell}$ and $\mathbf{z}_{[n]}^{\mathbf{c}}$, respectively, at the l th PU (See Eqs. (20)-(22)).
$\tilde{\eta}_{[n]}^{\mathbf{k},l}, \tilde{\eta}_{[n]}^{\ell,l}$ and $\tilde{\eta}_{[n]}^{\mathbf{c},l}$	Local estimates of hyperparameters $\eta_{[n]}^{\mathbf{k}}, \eta_{[n]}^{\ell}$ and $\eta_{[n]}^{\mathbf{c}}$ at the l th PU, respectively (See Eqs. (35)-(39)).
$\tilde{\alpha}_{[n]}^{\mathbf{c},l}$ and $\tilde{\gamma}_{[n]}^{\mathbf{c},l}$	Local proxy of the common precision $\alpha_{[n]}^{\mathbf{c}}$ at the l th PU, and $\tilde{\gamma}_{[n]}^{\mathbf{c},l} = 1/\tilde{\alpha}_{[n]}^{\mathbf{c},l}$, respectively (See Eq. (34) and before Eq. (41)).
$\mathbf{s}_{k[n]}^{\mathbf{k}}$ and $\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}}$	User-specific non-zero support of user k , for n th row and its estimate, respectively (See Eq. (28) and the discussion therein).
$\mathbf{s}_{l[n]}^{\ell}$ and $\hat{\mathbf{s}}_{l[n]}^{\ell}$	Cluster-specific non-zero support of cluster l , for n th row and its estimate, respectively (See Eq. (29) and the discussion therein).
$\mathbf{s}_{[n]}^{\mathbf{c}}$ and $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$	Common non-zero support of the n th row and its estimate at the l th PU, respectively (See Eqs. (31)-(32)).
$\tilde{\gamma}_{k[n]}^{\mathbf{t},l}$	Total prior variance hyperparameter of the entry $\mathbf{X}_k^{[n,m]}, \forall m$, defined as $\tilde{\gamma}_{k[n]}^{\mathbf{t},l} \triangleq 1/(\langle \bar{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \alpha_{k[n]}^{\mathbf{k}} + \langle \bar{\mathbf{z}}_{[n]}^{\ell,l} \rangle \alpha_{l[n]}^{\ell} + \langle \bar{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle \tilde{\alpha}_{[n]}^{\mathbf{c},l})$ (See after Eq. (28)).
\mathcal{S}_n and $\hat{\mathcal{S}}_n$	$\mathcal{S}_n = \begin{cases} \{1, 2, \dots, L\} & \text{if } n \in \Omega^{\mathbf{c}} \\ \emptyset & \text{otherwise,} \end{cases}$ and $\hat{\mathcal{S}}_n \triangleq \{1 \leq j \leq L : \mathbf{M}_j^{[n,k]} \neq 0, \forall k \in \mathcal{C}_j \text{ and } \hat{\mathbf{s}}_{l[n]}^{\ell} = 1\}$, respectively (See Eq. (33) and discussion therein).
$\hat{\mathcal{S}}_n^{-l}$	$\hat{\mathcal{S}}_n^{-l} = \{1, 2, \dots, L\} \setminus \hat{\mathcal{S}}_n$
$\mathbf{M}_j^{[n,k]}$	$[n, k]$ th entry of the message matrix \mathbf{M}_l sent from the l th PU (See Eq. (30)).
K_1 and K_2	$K_1 \triangleq \sum_{j' \in \hat{\mathcal{S}}_n^l} \mathcal{C}_{j'} $ and $K_2 \triangleq \sum_{j \in \hat{\mathcal{S}}_n^{-l}} \mathcal{C}_j $ (See before Lemma 1).

estimate $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$, calculated using (32), to update the global hyperparameters $\tilde{\eta}^{\mathbf{k},l}, \tilde{\eta}^{\ell,l}, \tilde{\eta}^{\mathbf{c},l}, \tilde{\alpha}^{\mathbf{c},l}$ as follows:

(a) For $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 1$, it uses the messages $\mathbf{M}_j^{[n,k]}, \forall k \in \mathcal{C}_j$, to calculate the local estimate of the global hyperparameters. The l th PU, using the limited information available, computes the local estimates in D-VBL such that they are unbiased approximation of their updates in C-VBL. The unbiased approximation, as shown later, turns out to be an effective estimate in practice.

(b) For $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 0$, it updates global hyperparameters without using any information from other PUs. The global hyperparameters $\tilde{\alpha}^{\mathbf{c},l}, \tilde{\eta}^{\mathbf{k},l}, \tilde{\eta}^{\ell,l}$ and $\tilde{\eta}^{\mathbf{c},l}$, at the l th PU, are next derived.

1) *Updating $\tilde{\alpha}^{\mathbf{c},l}$* : The l th PU, using the messages $\mathbf{M}_j, \forall j \neq l$, defined in (30), updates the global hyperparameter in the **M2** Step, $\tilde{\alpha}_{[n]}^{\mathbf{c},l}$ as follows

$$\tilde{\alpha}_{[n]}^{\mathbf{c},l} = \begin{cases} \frac{\sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j|}{\sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} 1/\alpha_{k[n]}^{\mathbf{k}}} & \text{if } \hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 1 \\ \alpha_{l[n]}^{\ell} & \text{otherwise.} \end{cases} \quad (34)$$

When the common support estimate is $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 1$, (34) gives common hyperparameter $\tilde{\alpha}_{[n]}^{\mathbf{c},l}$ by taking the harmonic mean of the available user-specific hyperparameters $\alpha_{k[n]}^{\mathbf{k}}, \forall k \in \mathcal{C}_j, \forall j$, which acts as an unbiased approximation of $\alpha_{[n]}^{\mathbf{c}}$ in (26).

2) *Updating $\tilde{\eta}^{\mathbf{k},l}, \tilde{\eta}^{\ell,l}, \tilde{\eta}^{\mathbf{c},l}$* : The l th PU, using the messages $\mathbf{M}_j, \forall j \neq l$, from all other PUs updates the global hyperparameters $\tilde{\eta}_{[n]}^{\mathbf{k},l}, \tilde{\eta}_{[n]}^{\ell,l}$ and $\tilde{\eta}_{[n]}^{\mathbf{c},l}$ and thus $\langle \bar{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle, \langle \bar{\mathbf{z}}_{[n]}^{\ell,l} \rangle$ and $\langle \bar{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle$ in the **E2** Step. We note that when all the hyperparameters are available at all the PUs, which is equivalent to the C-VBL, we have $\tilde{\eta}_{[n]}^{\mathbf{k},l} = \eta_{[n]}^{\mathbf{k}}, \tilde{\eta}_{[n]}^{\ell,l} = \eta_{[n]}^{\ell}$ and $\tilde{\eta}_{[n]}^{\mathbf{c},l} = \eta_{[n]}^{\mathbf{c}}, \forall l, n$. The updates of $\tilde{\eta}_{[n]}^{\mathbf{k},l}, \tilde{\eta}_{[n]}^{\ell,l}, \tilde{\eta}_{[n]}^{\mathbf{c},l}$, are thus given by (23), (24). However, when only censored copies of the variable $\alpha_k^{\mathbf{k}}$ are available and the binary support estimate $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 1$, the l th PU computes $\tilde{\eta}_{[n]}^{\mathbf{k},l}, \tilde{\eta}_{[n]}^{\ell,l}, \tilde{\eta}_{[n]}^{\mathbf{c},l}, \forall n$ as follows

$$\tilde{\eta}_{[n]}^{\mathbf{k},l} = \frac{MK}{K_1} \sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} \ln \alpha_{k[n]}^{\mathbf{k}}, \quad (35)$$

$$\tilde{\eta}_{[n]}^{\ell,l} = \frac{MK}{K_1} \sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j| \ln \left(\frac{|\mathcal{C}_j|}{\sum_{k \in \mathcal{C}_j} 1/\alpha_{k[n]}^{\mathbf{k}}} \right), \quad (36)$$

$$\tilde{\eta}_{[n]}^{\mathbf{c},l} = \frac{MK}{K_1} \sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j| \ln \left(\frac{\sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j|}{\sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} 1/\alpha_{k[n]}^{\mathbf{k}}} \right). \quad (37)$$

When $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 0$, the l th PU updates $\tilde{\eta}_{[n]}^{\mathbf{k},l}, \tilde{\eta}_{[n]}^{\ell,l}, \tilde{\eta}_{[n]}^{\mathbf{c},l}$, using only $\alpha_{k[n]}^{\mathbf{k}}, \forall k \in \mathcal{C}_l$, as follows

$$\tilde{\eta}_{[n]}^{\mathbf{k},l} = ML \sum_{k \in \mathcal{C}_l} \ln \alpha_{k[n]}^{\mathbf{k}}, \quad (38)$$

$$\tilde{\eta}_{[n]}^{\ell,l} = \tilde{\eta}_{[n]}^{\mathbf{c},l} = ML |\mathcal{C}_l| \ln \left(\frac{|\mathcal{C}_l|}{\sum_{k \in \mathcal{C}_l} 1/\alpha_{k[n]}^{\mathbf{k}}} \right). \quad (39)$$

Here, $K_1 \triangleq \sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j|$. The l th PU then uses (20)-(22) to update the multinoulli variables $\langle \bar{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle, \langle \bar{\mathbf{z}}_{[n]}^{\ell,l} \rangle$ and $\langle \bar{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle$. We note that the derived updates, for $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 1$, can be seen as the unbiased estimate of a true variable, given its noisy and censored observations.

We now summarize the proposed D-VBL design in Algorithm 1, wherein the l th PU first updates the posterior parameters in Step 2 and Step 3. It updates the local hyperparameters $\alpha_{k[n]}^{\mathbf{k}}, \forall k \in \mathcal{C}_l$, and $\alpha_{l[n]}^{\ell}$ in Step 4, and evaluates the binary support estimates $\mathbf{s}_k^{\mathbf{k}}, \forall k \in \mathcal{C}_l$ and $\hat{\mathbf{s}}_l^{\ell}$ in Step 5.1. It then sends the messages \mathbf{M}_l to the CCU, which, in Step 5.2, broadcasts the message to all other PUs. The l th PU next updates the fused parameter $\hat{\mathbf{s}}_{[n]}^{\mathbf{c},l}$ in Step 5.3 and updates the global hyperparameter $\tilde{\alpha}^{\mathbf{c},l}$ in Step 6. The l th PU updates $\hat{\mathbf{X}}_k^{[n,m]} = \mu_k^{[n,m]}$ in Step 7.

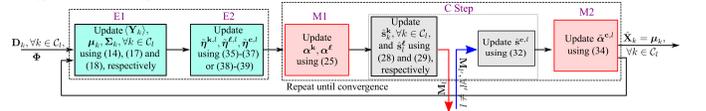


Fig. 3: Schematic of the D-VBL algorithm at the l th PU in the decentralized architecture at the BS. The red and blue arrows after the first C step show the outgoing and incoming message exchanges, respectively, with the other PUs via the CCU.

Remark 3. The D-VBL algorithm, unlike C-VBL, is designed to be independently applied to each cluster $l = 1, \dots, L$ by considering the structured sparsities. This is done by deriving local updates for the global variables in the C-VBL. Specifically, the l th PU in D-VBL algorithm estimates the global i) parameters $\tilde{\eta}^{\mathbf{k},l}, \tilde{\eta}^{\ell,l}, \tilde{\eta}^{\mathbf{c},l}$ according to (35)-(39) in the E2 step and; ii) the hyperparameter $\tilde{\alpha}^{\mathbf{c},l}$, according to (34), using the information received from all other PUs. Since all the PUs estimate the channel for a group of users in parallel, the complexity, as shown later, analytically in Section IV-C and numerically in Fig. 6a, reduces drastically.

Algorithm 1: Decentralized VBL Algorithm

Input: Set number of iterations I_t , error tolerance ϵ and initial values of $\alpha_k^k, \alpha_l^\ell, \tilde{\alpha}^{c,l}, \langle \tilde{\mathbf{z}}^{k,l} \rangle, \langle \tilde{\mathbf{z}}^{\ell,l} \rangle, \langle \tilde{\mathbf{z}}^{c,l} \rangle, \hat{s}_k^c, \forall k \in \mathcal{C}_l$.

1 **for** $i_t \leftarrow 1$ **to** I_t **do**

2 **E1 Step:** Update $\langle \mathbf{Y}_k \rangle, \boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k, \forall k \in \mathcal{C}_l, \forall l$, according to using (14), (17) and (18), respectively.

3 **E2 Step:** Compute $\tilde{\boldsymbol{\eta}}_{[n]}^{k,l}, \tilde{\boldsymbol{\eta}}_{[n]}^{\ell,l}$ and $\tilde{\boldsymbol{\eta}}_{[n]}^{c,l}$ using (35)-(37) or (38)-(39), to update $\langle \tilde{\mathbf{z}}_{[n]}^{k,l} \rangle, \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle$ and $\langle \tilde{\mathbf{z}}_{[n]}^{c,l} \rangle, \forall n, l$, according to (20), (21) and (22), respectively.

4 **M1 Step:** Update $\alpha_{k[n]}^k, \forall n, k \in \mathcal{C}_l, l, \alpha_{l[n]}^\ell, \forall n$, using (25).

5.1 **C Step:** Each PU l updates its user-specific support $\hat{s}_{k[n]}^k, \forall n, \forall k \in \mathcal{C}_l$, according to (28) and the cluster-specific non-zero support \hat{s}_l^ℓ according to (29).

5.2 **C Step:** Each PU l sends the message \mathbf{M}_l , in (30) to the CCU which broadcasts it to all other PUs.

5.3 **C Step:** Compute $\hat{s}_{[n]}^{c,l}$ using the majority rule (32).

6 **M2 Step:** Compute $\tilde{\alpha}_{[n]}^{c,l}, \forall n, l$, using (34).

7 Update $\hat{\mathbf{X}}_k^{[n,m]} = \boldsymbol{\mu}_k^{[n,m]}, \forall l, \forall k \in \mathcal{C}_l, \forall m, n$. Do until convergence
 if $\|\hat{\mathbf{X}}_k^{(i_t)} - \hat{\mathbf{X}}_k^{(i_t-1)}\| \leq \epsilon \|\hat{\mathbf{X}}_k^{[i_t-1]}\|, \forall k \in \mathcal{C}_l$ **and** $\forall l$ **then** break.

8 **return** $\hat{\mathbf{X}}_k, \forall k \in \mathcal{C}_l$ **and** $\forall l$ (BS calculates the channel using (3) as $\hat{\mathbf{H}}_k = \mathbf{A}_R \hat{\mathbf{X}}_k^H \mathbf{A}_T^H$).

C. Complexity analysis of C-VBL and D-VBL algorithms

We show in Table IVa and Table IVb at each iteration the step-wise complexity of the proposed C-VBL and D-VBL algorithm per PU, respectively. We use the Woodbury identity from [30] to reduce the computational complexity of inverting $N \times N$ covariance matrix $\boldsymbol{\Sigma}_k$ as follows

$$\boldsymbol{\Sigma}_k = \boldsymbol{\Lambda}_k^{-1} - \boldsymbol{\Lambda}_k^{-1} \boldsymbol{\Phi}^H (\sigma_k^2 \mathbf{I}_T + \boldsymbol{\Phi} \boldsymbol{\Lambda}_k^{-1} \boldsymbol{\Phi}^H)^{-1} \boldsymbol{\Phi} \boldsymbol{\Lambda}_k^{-1} \quad (40)$$

The complexity of computing $\boldsymbol{\Sigma}_k$ thus reduces from $\mathcal{O}(N^3)$ to $\mathcal{O}(T^3)$, which is included in the first rows of Table IVa and Table IVb. We note from these tables that the complexity of both the algorithms is dominated by the E-step), which calculates the posterior mean and variance of \mathbf{X}_k . The overall computational complexity of the C-VBL and D-VBL are thus $\mathcal{O}(T^3 + N^2T)MK$ and $\mathcal{O}((T^3 + N^2T)M|\mathcal{C}_l|)$, respectively. We see that the complexity of C-VBL increases linearly with the number of users, while that of the D-VBL does not increase if it keeps the number of users per PU ($|\mathcal{C}_l|$) constant. The D-VBL algorithm thus provides a novel scalable constant-complexity framework of channel estimation.

Hardware complexity of quantizer: We note that the quantizer plays a pivotal role in the C-VBL algorithm only at the E1 step, where the BS computes the posterior mean of the unquantized observation, i.e., $\langle \mathbf{Y}_k \rangle$ using (14). We observe that while updating $\langle \mathbf{Y}_k \rangle$, a higher number of quantization bits q increases the complexity of calculating the upper and lower limits y_l and y_u . We, however, note from the Table IVa, that the C-VBL algorithm's complexity is dominated by the calculation of the covariance matrix \mathbf{X}_k and $\boldsymbol{\Sigma}_k, \forall k$, which is on the order of $\mathcal{O}[(T^3 + N^2T)MK]$. The computation of $\langle \mathbf{Y}_k \rangle$ has very low complexity, which can be ignored. We thus conclude that the cost of quantization associated with different q values remains almost the same and it is negligible.

V. CONVERGENCE OF THE PROPOSED ALGORITHM

We show that D-VBL updates converge to their C-VBL counterparts. Depending on the sparsity type, proof is split in two parts. Part one proves that the D-VBL updates converge to their C-VBL counterparts by quantifying the maximum absolute error between the global C-VBL variables, and their local proxies in the D-VBL algorithm. These proofs require non-trivial step-wise analysis of the error between global variables and their local proxies. The second part shows that D-VBL updates reduce to a special case of C-VBL updates.

We note that the updates of local parameters $\{\langle \mathbf{Y}_k^{[:,m]} \rangle, \boldsymbol{\mu}_k^{[:,m]}, \boldsymbol{\Sigma}_k, \alpha_k^k\}, \forall k \in \mathcal{C}_l$ and α_l^ℓ are the same for both the C- and D-VBL algorithms. We thus show the convergence of the global hyperparameter updates $\tilde{\alpha}^{c,l}, \tilde{\boldsymbol{\eta}}^{k,l}, \tilde{\boldsymbol{\eta}}^{\ell,l}$ and $\tilde{\boldsymbol{\eta}}^{c,l}$. For the n th index, we divide the analysis into two cases

Case A: $\hat{s}_{[n]}^{c,l} = 1$, which corresponds to the common sparsity case, i.e., $n \in \Omega^c$.

Case B: $\hat{s}_{[n]}^{c,l} = 0$, which corresponds to one of the following cases i) user-specific sparsity, i.e., $n \in \Omega_k^k$ and $n \notin \Omega_{k'}^k, \forall k' \neq k$ users; ii) cluster-specific sparsity, i.e., $n \in \Omega_l^\ell$ and $n \notin \Omega_{j'}^\ell, \forall j' \neq l$ clusters; and iii) all zeros, i.e., $n \notin \Omega_k, \forall k$ users,

For Case A, we derive upper bounds on the absolute error between the C- and D-VBL updates, and show that it converges to zero when $K_2 \triangleq \sum_{j \in \hat{\mathcal{S}}_n^{c,l}} |\mathcal{C}_j| \rightarrow 0$. For Case B, we show that the D-VBL updates reduce to a special case of C-VBL updates.

Case A: The following lemmas assume, for the sake of brevity, that the number of PUs is equal to the number of users $L = K$ and $\text{SNR}_k = \text{SNR}$ for all k . The proofs can be readily extended for $L \neq K$ and unequal SNR, but will complicate the notation. We also define $K_1 \triangleq \sum_{j' \in \hat{\mathcal{S}}_n^k} |\mathcal{C}_{j'}|$ and $K_2 \triangleq \sum_{j \in \hat{\mathcal{S}}_n^{c,l}} |\mathcal{C}_j|$. We start with $\tilde{\alpha}_{[n]}^{c,l}$ and state the following lemma.

Lemma 1. The absolute error of common variance hyperparameter of the D-VBL for the l th PU, defined as $\Delta_{\gamma_{[n]}^c}^l \triangleq |\gamma_{[n]}^c - \tilde{\gamma}_{[n]}^{c,l}|$, with $\tilde{\gamma}_{[n]}^{c,l} \triangleq \frac{1}{\tilde{\alpha}_{[n]}^{c,l}}$ and $\gamma_{[n]}^c \triangleq \frac{1}{\alpha_{[n]}^c}$, can be upper bounded as follows

$$\Delta_{\gamma_{[n]}^c}^l \leq \frac{K_2}{K} \max \left\{ \max_{k \in \mathcal{C}_j, j \in \hat{\mathcal{S}}_n^{c,l}} \left(\gamma_{k[n]}^k \right), \frac{c}{\text{SNR}} \right\}. \quad (41)$$

Proof. See Appendix B. \square

We prove the convergence of global hyperparameters $\tilde{\boldsymbol{\eta}}_{[n]}^{c,l}, \tilde{\boldsymbol{\eta}}_{[n]}^{\ell,l}, \tilde{\boldsymbol{\eta}}_{[n]}^{k,l}$ in the following lemma.

Lemma 2. The absolute error of the hyperparameter

(a) $\tilde{\boldsymbol{\eta}}_{[n]}^{c,l}$, defined as $\Delta_{\boldsymbol{\eta}_{[n]}^c}^l \triangleq |\boldsymbol{\eta}_{[n]}^c - \tilde{\boldsymbol{\eta}}_{[n]}^{c,l}|$ can be upper bounded as follows

$$\Delta_{\boldsymbol{\eta}_{[n]}^c}^l \leq MK \max \left\{ -\ln \frac{K_1}{K}, \ln \left(1 + \frac{K_2}{\sum_{j' \in \hat{\mathcal{S}}_n^k} \sum_{k' \in \mathcal{C}_{j'}} \langle \tilde{\mathbf{z}}_{[n]}^{k,j'} \rangle} \right) + \ln \frac{K_1}{K} \right\}; \quad (42)$$

Table IV: (a) Per iteration complexity of C-VBL. (b) Per cluster per iteration complexity of D-VBL.

Variable updates	proposed C-VBL (Complexity)
$\langle \mathbf{Y}_k^{[:m]} \rangle, \forall m, k$	$\mathcal{O}(TMK)$
$\mu_k^{[:m]}, \Sigma_k, \forall m, k$	$\mathcal{O}((T^3 + N^2T)MK)$
$\langle \mathbf{z}^k \rangle, \langle \mathbf{z}^\ell \rangle, \langle \mathbf{z}^c \rangle,$ $\eta^k, \eta^\ell, \eta^c$	$\mathcal{O}(N)$
$\alpha_k, \forall k, \alpha_l^\ell, \forall l, \alpha^c$	$\mathcal{O}(NK)$

(a)

Variable updates	proposed D-VBL (Complexity per PU)
$\langle \mathbf{Y}_k^{[:m]} \rangle, \forall m, k$	$\mathcal{O}(TMK)$
$\mu_k^{[:m]}, \Sigma_k, \forall m, k \in \mathcal{C}_l$	$\mathcal{O}((T^3 + N^2T)M \mathcal{C}_l)$
$\langle \tilde{\mathbf{z}}^{k,l} \rangle, \langle \tilde{\mathbf{z}}^{\ell,l} \rangle, \langle \tilde{\mathbf{z}}^{c,l} \rangle, \tilde{\eta}^{k,l}, \tilde{\eta}^{\ell,l}, \tilde{\eta}^{c,l}$ $\alpha_k^k, \hat{\mathbf{s}}_k^k, \forall k \in \mathcal{C}_l, \alpha_l^\ell$ $\hat{\mathbf{s}}_l^\ell, \hat{\mathbf{s}}_l^c, \tilde{\alpha}^{c,l}$	$\mathcal{O}(N)$ $\mathcal{O}(N \mathcal{C}_l + N)$ $\mathcal{O}(N)$

(b)

(b) $\tilde{\eta}_{[n]}^{\ell,l}$ defined as $\Delta_{\eta_{[n]}^\ell}^l \triangleq |\eta_{[n]}^\ell - \tilde{\eta}_{[n]}^{\ell,l}|$ or $\tilde{\eta}_{[n]}^{k,l}$, defined as

$\Delta_{\eta_{[n]}^k}^l \triangleq |\eta_{[n]}^k - \tilde{\eta}_{[n]}^{k,l}|$ can be bounded as

$$\Delta_{\eta_{[n]}^k}^l \leq MK_2 \max \left\{ \ln \left(\frac{\min_{k' \in \mathcal{C}_{j'}, j' \in \hat{\mathcal{S}}_n^l} (\gamma_{k[n]}^k)}{\min_{k \in \mathcal{C}_j, j \in \hat{\mathcal{S}}_n^{-l}} (\gamma_{k[n]}^k)} \right), \right. \\ \left. - \frac{1}{K_1} \sum_{j' \in \hat{\mathcal{S}}_n^l} \sum_{k' \in \mathcal{C}_{j'}} \ln \langle \tilde{\mathbf{z}}_{[n]}^{k,j'} \rangle \right\}. \quad (43)$$

Proof. See Appendix B. \square

The scalar K_1 depicts the number of users $k \in \mathcal{C}_l, \forall l$, for which the estimated common support $\hat{\mathbf{s}}_{[n]}^{c,l}$ is one. The scalar K_2 , similarly, depicts the number of users $k \in \mathcal{C}_l, \forall l$, for which the estimated common support $\hat{\mathbf{s}}_{[n]}^{c,l}$ is zero. When the estimated common and true supports are equal $\hat{\mathbf{s}}_{[n]}^{c,l} = \mathbf{s}_{[n]}^c$, the set obeys $\hat{\mathcal{S}}_n^l = \mathcal{S}_n$. For $n \in \Omega^c$, the set $\hat{\mathcal{S}}_n^l = \{1, 2, \dots, L\}$ and $\hat{\mathcal{S}}_n^{-l} = \emptyset$, making the constants $K_1 = K$ and $K_2 = 0$ and the upper bound (RHS) in (41), (42) and (43) equal to zero. *The non-zero upper bound on the absolute error $\Delta_{\gamma_{[n]}^c}$ thus quantifies the error caused due to mismatch in the estimated and the true common support.*

Case B: We will show that the proposed updates reduce to a special case of the C-VBL algorithm. We commence by stating the following lemma, which is proved in Appendix B.

Lemma 3. The updates of the proposed D-VBL, for the case $\hat{\mathbf{s}}_{[n]}^{c,l} = 0$, are equivalent to applying C-VBL at the l th PU in a standalone manner with a prior on $\mathbf{X}_k^{[n,m]}, \forall m, \forall k \in \mathcal{C}_l$, as follows

$$P(\mathbf{X}_k^{[n,m]} | \alpha_{k[n]}^k, \alpha_{l[n]}^\ell, \tilde{\mathbf{z}}_{[n]}^{k,l}, \tilde{\mathbf{z}}_{[n]}^{\ell,l}) \\ = \mathcal{N}_C \left(0, \left(\alpha_{k[n]}^k \right)^{-1} \right) \tilde{\mathbf{z}}_{[n]}^{k,l} \mathcal{N}_C \left(0, \left(\alpha_{l[n]}^\ell \right)^{-1} \right) \tilde{\mathbf{z}}_{[n]}^{\ell,l} \quad (44)$$

We know from Section IV-A that case B includes the cases of all zeros, user-specific and cluster-specific sparsities. Since there is no common sparsity across the clusters in these cases, we ideally want the prior to be independent across the clusters. The D-VBL prior in (44) is indeed independent across clusters as it contains all the hyperparameters, which are independent of other clusters $j \neq l$. Since the prior in (44) is a special case of the C-VBL updates, the D-VBL updates in this case also converge to that of the C-VBL.

VI. SIMULATION RESULTS

We numerically demonstrate the performance improvement achieved by the proposed D-VBL algorithm upon comparing it to the following decentralized counterparts: 1) *FB-DSBL* [13]: fusion based decentralized SBL (FB-DSBL) which decentrally estimates the sparse channel by assuming only the common

sparsity; 2) *M-SBL* [24]: decentrally estimates the channel by using multiple SBL (M-SBL) schemes, where each PU estimates the matrix $\mathbf{X}_{\mathcal{C}_l}$, individually and in parallel. It only assumes common cluster-specific sparsity; 3) *D-VBL*: proposed D-VBL algorithm with random user clustering; and 4) *Oracle LS*: assumes that the true non-zero support of the overall channel \mathbf{X} is known and calculates the least-squares solution for this over-determined problem. Since the Oracle LS algorithm uses extra information about the non-zero support, it lower-bounds the NMSE and BER to all the algorithms which estimate the overall channel matrix \mathbf{X} without this knowledge.

We consider a cellular system relying on an $N = 256$ -antenna BS and $K = 20$ users, each having $M = 2$ antennas. We use $T = 52$ pilots, since for compressive-sensing-based estimation, the pilot length has to obey $T \geq 2|\Omega^l|$ [29]. Similar to [3], [4], we choose Φ , according to the Rademacher distribution, with each of its entry drawn from the set $\{\sqrt{P/N}, -\sqrt{P/N}\}$ uniformly. For the sake of simplicity, we assume the same sparsity levels for all the users and all clusters i.e., $(|\Omega_k^k| = |\Omega^k|, \forall k, |\Omega_l^\ell| = |\Omega^\ell|, \forall l)$ and a known noise variance of $\sigma^2 = \sigma_k^2, \forall k$. The SNR is therefore $P/\sigma^2, \forall k$, and we set $|\Omega^k| = 4, |\Omega^\ell| = 6, |\Omega^c| = 14$. The non-zero angular domain channel gains obey the i.i.d Rayleigh distribution, so that $\mathbb{E}(\|\mathbf{H}_k\|_2^2) = MN$. We also assume that the number of users per PU is the same, i.e., $|\mathcal{C}_l| = K/L, \forall l$. We consider $L = 4$ PUs and each PU is thus assigned 5 users. The thresholding constant in (28) is set to $c = 8$. The EM algorithm is executed for a maximum of $I_t = 150$ iterations with the following initial parameters [3]: $\langle \tilde{\mathbf{z}}_{[n]}^{k,l} \rangle = \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle = \langle \tilde{\mathbf{z}}_{[n]}^{c,l} \rangle = 1/3, \alpha_{k[n]}^k = \alpha_{l[n]}^\ell = \tilde{\alpha}^{c,l} = 100, \forall n, k, l$. The error tolerance is $\epsilon = 10^{-3}$. These settings remain the same unless stated otherwise.

NMSE performance: In Fig. 4a, we first compare the NMSE of the proposed D-VBL algorithm for different number of quantization bits q , to the unquantized D-VBL (uD-VBL) algorithm, which is derived from D-VBL by setting $q = 16$. We observe that the 7-bit D-VBL has the same NMSE as that of the uD-VBL. *The novel D-VBL algorithm thus drastically reduces the overhead of sending the observations from CCU to PUs and that too without affecting the NMSE.* We also note that the NMSE of D-VBL for $q < 4$ bits does not reduce upon increasing the SNR. We note that the quantizer adds noise with the variance $\sigma_q^2 \propto \Delta^2/12$, to its input signal. Here Δ is the quantizer step size. For $q < 4$ quantization bits, the step size Δ increases considerably. This leads to a higher quantization noise variance σ_q^2 than the thermal noise variance σ^2 . The overall $\text{SNR}_o = P/(\sigma^2 + \sigma_q^2)$ is thus dominated by σ_q^2 . Even though the $\text{SNR} = P/\sigma^2$ is increased in Fig. 4a, the overall

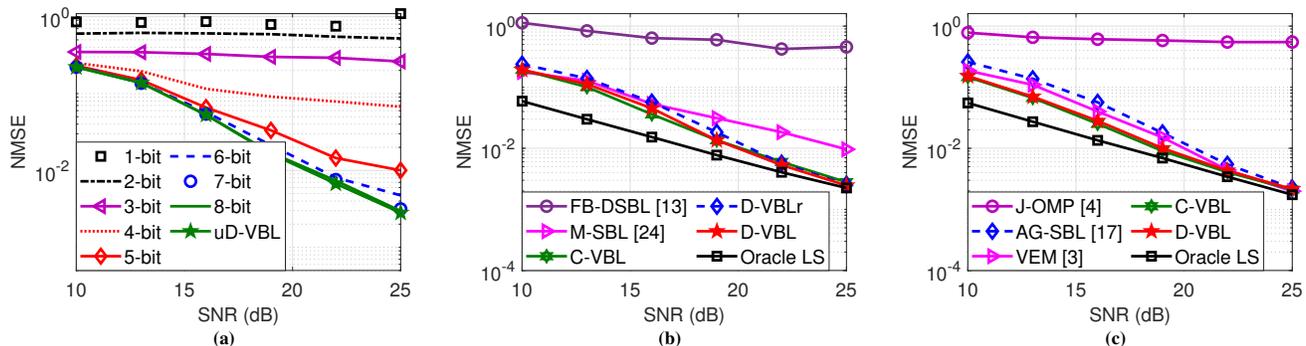


Fig. 4: NMSE comparison of D-VBL algorithm with (a) different number of quantization bits; (b) existing state-of-the-art decentralized algorithms for $T = 52$ pilots, $K = 20$ users, and; (c) with existing state-of-the-art centralized algorithms for $T = 52$ pilots, $K = 8$ users.

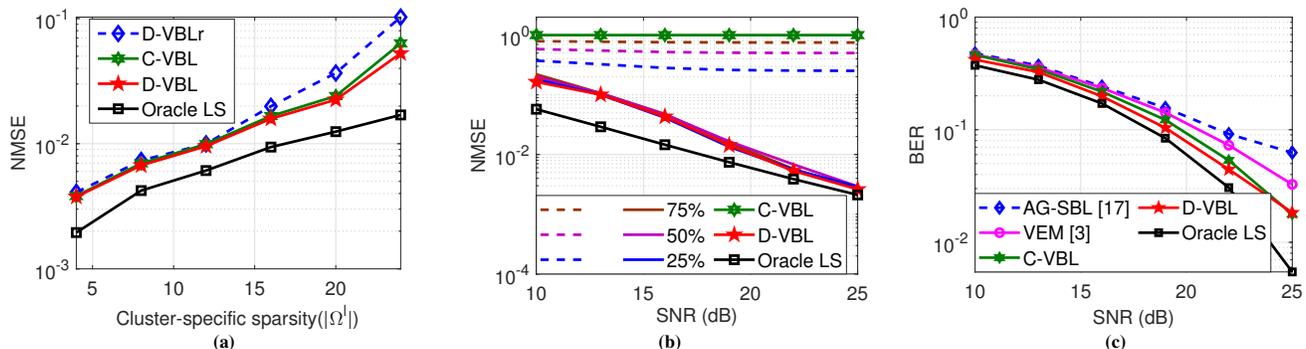


Fig. 5: (a) NMSE of D-VBL and C-VBL algorithms versus cluster-specific sparsity level ($|\Omega^\ell|$) for $K = 20$ users, $|\Omega^c| = |\Omega^k| = 0$, and; (b) NMSE for two different PUs failure set-ups; and (c) BER comparison of D-VBL, C-VBL with the existing state-of-the-art algorithms.

SNR_o does not increase, and the BER, therefore exhibits a residual floor. Since the 7-bit D-VBL has the same NMSE as that of uD-VBL, we now compare, unless otherwise stated, 7-bit C- and D-VBL to the existing unquantized centralized and decentralized algorithms. This is because their quantized versions are yet to be investigated.

We assume a practical channel, where the insignificant taps are not strictly zero. These close-to-zero tap-gains are generated randomly and uniformly between $[0, 10^{-2}]$. We see from Fig. 4b that the NMSE of the proposed D-VBL algorithm is close to the oracle LS, and it is much lower than that of other decentralized algorithms. This is because the proposed GM prior aptly captures the channel sparsity, and the variational inference correctly estimates all the parameters. The FB-DSBL algorithm assumes that the channel has only common sparsity, and it thus models the user- and cluster-specific sparsities as common sparsity, which increases its NMSE. The M-SBL algorithm estimates the channel individually at each PU, without communicating with other PUs. It thus i) partially exploits the common sparsity among all the users; and ii) wrongly models the user-specific sparsity as the *common* cluster-specific sparsity. The D-VBLr algorithm, which randomly clusters users, also has similar NMSE as that of the D-VBL algorithm. We will investigate this behavior in detail later in Fig. 5a. We observe that the D-VBL performance also matches that of its centralized C-VBL counterpart. This is because, as shown in the Section V, the D-VBL updates reduce to that of C-VBL.

We next compare in Fig. 4c the NMSE of the C-VBL and D-VBL algorithms to the following centralized algorithms:

1) *J-OMP* [4], which estimates the overall channel matrix \mathbf{X} by exploiting the joint and individual sparsities; 2) *VEM* [3], which centrally estimates \mathbf{X} and exploits joint and individual sparsities; and 3) *AG-SBL* [17], which centrally estimates \mathbf{X} and exploits user-specific, cluster-specific, and common sparsities. For this study, we fix the number of users to $K = 8$ and the sparsity levels to $|\Omega^c| = 12, |\Omega^k| = 0, |\Omega^\ell| = 12$. We see that both the D-VBL and C-VBL outperform the *J-OMP* and *VEM* algorithms. This is because these algorithms do not consider the cluster-specific sparsity and wrongly model it as either user-specific or common sparsity. We also note that the *AG-SBL* algorithm has slightly inferior NMSE than the proposed algorithms. This is due to its hierarchical modeling, which uses a lot of hyperparameters and is thus sensitive to their initialization, in contrast to the proposed algorithms. The above results show that due to the Gaussian mixture prior, the proposed algorithms outperform others centralized/decentralized ones for realistic non-sparse channels.

Earlier we observed in Fig. 4b that D-VBLr has similar NMSE as that of C- and D-VBL. We investigate this behavior further in Fig. 5a by comparing their NMSE upon varying the cluster-specific sparsity $|\Omega^\ell|$, and by fixing $|\Omega^k| = |\Omega^c| = 0$. We see that for low $|\Omega^\ell|$ values, C-VBL, D-VBL and D-VBLr have similar NMSE, but the gap increases upon increasing $|\Omega^\ell|$. This is because the random clustering of the D-VBLr algorithm does not let it exploit the cluster-specific sparsity. We also note that the D-VBLr algorithm models the performance of D-VBL, when it experiences a practical scenario of users having non-exclusive clusters. This is because it randomly clusters the users, which might result in overlapping clusters.

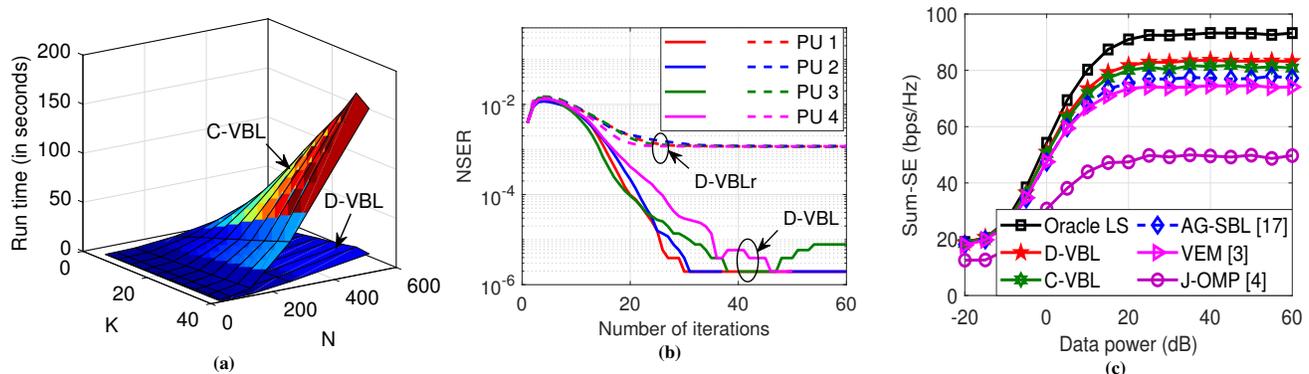


Fig. 6: (a) Run-time comparison of C-VBL and D-VBL algorithms by varying the number of users K and BS antennas N ; (b) NSER per PU versus number of iterations, and; (c) Sum-SE comparison of the proposed C- and D-VBL, and existing centralized algorithms.

We thus infer that the D-VBLr algorithm used for such a scenario has only slightly inferior NMSE to D-VBL. Further, the slope of all the algorithms changes at $|\Omega^\ell| = 24$. This is because now the number of observations reaches the well known limit from the compressed sensing theory to recover a sparse vector with high probability i.e., $T \geq 2|\Omega_k^t| = 48$ [31].

Robustness against failure: We now show in Fig. 5b for two different setups, that the D-VBL is robust to PU failures. In case-i, a fraction of PUs fail and the D-VBL algorithm does not estimate the channel of users assigned to the failed PUs. We calculate their NMSE by assuming $\hat{\mathbf{X}}_k = \mathbf{0}_{N \times M}$. In case-ii, users assigned to the failed PUs are reassigned to the operational ones. The failure percentage (FP) of 25% in this figure implies a failure of 1 PU out of $L = 4$. We observe for different FPs that the D-VBL algorithm in case-i has much lower NMSE than C-VBL, since C-VBL cannot estimate the channel for such failures due to its centralized nature. We next see that the D-VBL NMSE in case-ii, for a 25% to 75% FP, is only slightly inferior to 0% FP. This is because the reallocation in case-ii leads to different users clusters, having different cluster-specific sparsities being processed together, which slightly increases the NMSE.

BER performance: In Fig. 5c, we now compare the BER of the proposed C- and D-VBL algorithms to that of the existing centralized ones. The BS now uses estimated channels to design a zero-forcing precoder while transmitting BPSK data. For this study, we consider $N = 128$ BS antennas, $M = 2$ user antennas, $K = 8$ users, $L = 4$ PUs, $|\Omega^c| = 0$ common sparsity level, $|\Omega^\ell| = 12$ cluster-specific sparsity level, $|\Omega^k| = 0$ user-specific sparsity level, and $T = 26$ pilots. We see that both of them have much lower BER than the existing algorithms.

Run-time complexity: We plot in Fig. 6a the run-time (in seconds) of both the C- and D-VBL algorithms by simultaneously varying the number of users K and the BS antennas N . We consider $L = 4$ PUs for the D-VBL algorithm. We observe from Fig. 6a that C-VBL has much higher run time than D-VBL, particularly for large values of N . This is because it processes the whole $T \times K$ -sized observation matrix \mathbf{Y} to estimate the channels of all the K users jointly, where T is the number of pilots. By contrast, the D-VBL algorithm assigns it to multiple decentralized PUs, which simultaneously estimate the channel of $k \in \mathcal{C}_l$ users, using $T \times |\mathcal{C}_l|$ -sized observation

matrix $\mathbf{Y}_{\mathcal{C}_l}$.

SE and EE comparison: The BS precodes the users data $\mathbf{s} \in \mathbb{C}^{MK \times 1}$ by zero-forcing (ZF) precoder $\mathbf{W} = \hat{\mathbf{H}}^H (\hat{\mathbf{H}} \hat{\mathbf{H}}^H)^{-1} \in \mathbb{C}^{N \times MK}$. Here $\hat{\mathbf{H}} = [\hat{\mathbf{H}}_1, \dots, \hat{\mathbf{H}}_K]^T \in \mathbb{C}^{MK \times N}$ is the estimated channel. The received signal of the k th user, denoted as $\mathbf{y}_k \in \mathbb{C}^{M \times 1}$, is given by

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{W} \mathbf{s} + \mathbf{n}_k = \mathbf{H}_k \mathbf{W}_k \mathbf{s}_k + \mathbf{H}_k \sum_{i \neq k} \mathbf{W}_i \mathbf{s}_i + \mathbf{n}_k. \quad (45)$$

The matrix $\mathbf{W}_k \in \mathbb{C}^{N \times M}$ is the k th block column of the matrix \mathbf{W} , i.e., $\mathbf{W} = [\mathbf{W}_1, \dots, \mathbf{W}_K]$. The vector $\mathbf{s}_k \in \mathbb{C}^{M \times 1}$ is the k th block of the vector \mathbf{s} , i.e., $\mathbf{s} = [\mathbf{s}_1, \dots, \mathbf{s}_K]^T$, and $\mathbb{E}[\mathbf{s}_k \mathbf{s}_k^H] = P_k \mathbf{I}_M$. The second term in (45) is the multi-user interference. The noise vector \mathbf{n}_k is distributed as $\mathbf{n}_k \sim \mathcal{N}_c(\mathbf{0}, \sigma_k^2 \mathbf{I}_M)$. The sum-SE (in bps/Hz) for the K users is defined as follows [32]

$$\text{SE} = \left(1 - \frac{T_p}{T_c}\right) \sum_{k=1}^K \mathbb{E} [\log_2 |\mathbf{I}_M + \mathbf{S}_s \mathbf{N}_{in}^{-1}|]. \quad (46)$$

Here T_p and T_c denote the length of pilots and of the coherence interval, respectively. The matrices $\mathbf{S}_s = P_k \mathbf{H}_k \mathbf{W}_k \mathbf{W}_k^H \mathbf{H}_k^H$ and $\mathbf{N}_{in} = P_k \mathbf{H}_k \sum_{i \neq k} \mathbf{W}_i \sum_{j \neq k} \mathbf{W}_j^H \mathbf{H}_k^H + \sigma_k^2$ are the signal and interference plus noise matrices, respectively. The EE (in bits/J) is defined as the ratio of the sum-SE and of the total power consumed by the system i.e.,

$$\text{EE} = \frac{\text{SE}}{\sum_{k=1}^K \mu_k^{-1} P_k + \mu_{BS}^{-1} P_{BS} + \mathcal{P}_C}, \quad (47)$$

where B is the system bandwidth, and the scalars $\mu_k, \mu_{BS} \in (0, 1]$ denote the power amplifier efficiency at the k th user and the BS, respectively. The term \mathcal{P}_C is the total circuit power consumed by the system, which is modeled as follows [1]

$$\mathcal{P}_C = P_{\text{FIX}} + P_{\text{TC}} + P_{\text{SP}} + P_{\text{CE}}. \quad (48)$$

Here the values P_{FIX} and P_{TC} are taken from the parameter set 2 [1, Table 5.3], and P_{SP} is taken from [1, Table 5.2]. The term P_{CE} is the power consumed by the channel estimator, and is given by $P_{\text{CE}} = \frac{4B}{T_c L_{\text{BS}}} \times C$, where B is the bandwidth, T_c is the coherence interval, L_{BS} denotes the BS's computational efficiency in flops/W, and C is the complexity of the estimator. We assumed $B = 20$ MHz and $T_c = 1000$. The value of L_{BS} is taken from [1, Table 5.3] and the complexities of C- and D-VBL, from Table IVa and Table IVb, respectively, are

$$\text{for C-VBL: } C = (T^3 + N^2 T) M K I_t$$

$$\text{for D-VBL: } C = (T^3 + N^2 T) M |\mathcal{C}_l| I_t.$$

In Fig. 6c, we first compare that the sum-SE of the proposed

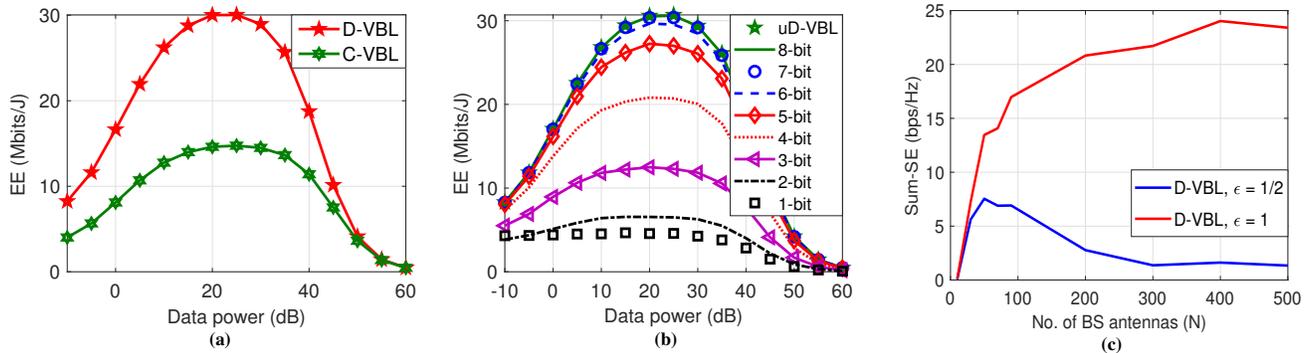


Fig. 7: (a) EE comparison of the proposed C- and D-VBL algorithms; (b) EE of D-VBL algorithms with different levels of quantization, with varying downlink data power, and; (c) number of BS antennas N . Here the data and pilot powers are varying with N as $P_p = P_d = P_0/N^\epsilon$, with $P_0 = 20$ dB.

algorithms to the existing ones. We observe that the Oracle-LS has the highest SE. This is because it knows the non-zero indices of the sparse channel. We also observe that the proposed C- and D-VBL algorithms outperform the J-OMP [3], VEM [6] and AG-SBL [14] algorithm. This is because, as shown in Fig. 4c, the proposed algorithm has a lower NMSE value. Given the better channel estimates, the ZF precoder is able to suppress multi-user interference better. We also observe that the SE of all the algorithms increases first and then saturates with increase the data power. This is due to the high multi-user interference at higher SNR values.

We next plot in Fig. 7a, the EE of the C-VBL and D-VBL algorithms. We observe that the C-VBL algorithm has lower EE than D-VBL. This is due to its higher computational complexity. Further, the EE initially increases with the data power P_d , reaches a maximum value and then reduces to zero. This is because for low P_d values, the increase in SE is commensurate with the energy dissipated to achieve it, which increases the EE. At high P_d values, the SE saturates, but the system keep on consuming power, which reduces the EE. In Fig. 7b, we compare the EE of the proposed D-VBL algorithm for different number of quantization bits q to that of the unquantized D-VBL (uD-VBL) algorithm. We see that for $q \geq 6$ -bits the EE of qD-VBL and uD-VBL are the same.

Power scaling: For this study, we consider $T = N/5$ pilots, $K = 5$ users, $L = 5$ clusters, and sparsity of $|\Omega^k| = 0$, $|\Omega^\ell| = 0$, $|\Omega^c| = N/10$. We vary the data and pilot powers as a function of number of antennas, as $P_d = P_p = P_0/N^\epsilon$, for $\epsilon = 1/2$ and $\epsilon = 1$. We observe from the figure that when P_d and P_p are reduced as $1/N^\epsilon$ with $\epsilon = 1/2$, the sum SE approaches its asymptotic limit. When P_d and P_p are reduced as $1/N^\epsilon$ with $\epsilon = 1$, we see that the sum SE goes to zero with the increase of N . The proposed design thus follows $\mathcal{O}(\sqrt{N})$ reduction in the transmission power in mMIMO systems [1, Lemma 5.1, Fig. 5.3].

Convergence: We next show in Fig. 6b that the common binary support $\hat{\mathbf{s}}_n^{c,l}$, calculated at the l th PU, converges to the true binary support of the angular domain channel \mathbf{X} given by $\mathbf{s}_n^c, \forall n$. We use the normalized support error rate (NSER) metric at the l th PU, which is defined as $\text{NSER}_l = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(\hat{\mathbf{s}}_n^{c,l} \neq \mathbf{s}_n^c)$ [12]. Here $\mathbb{I}(\cdot)$ is the indicator function. The metric NSER_l gives the normalized error in the estimated common support $\hat{\mathbf{s}}_n^{c,l}, \forall n$ at the l th PU. We see from Fig. 6b that NSER reduces upon iterating which shows that

$\hat{\mathbf{s}}_n^{c,l}$ converges to $\mathbf{s}_n^c, \forall n$.

VII. CONCLUSION AND FUTURE WORKS

We proposed the C-VBL and the low-complexity D-VBL algorithms for the BS to estimate the DL channels of FDD mMIMO systems. We analytically showed that the upper bound on the absolute error between the updates of D-VBL and its centralized counterpart C-VBL tends to zero, when the non-zero support is estimated accurately. We investigated the NMSE, BER, SE and EE of the C- and D-VBL algorithms, and showed that it outperforms multiple other state-of-the-art centralized and decentralized algorithms.

The current work considered digital transceiver architectures, where each antenna is connected to a RF chain. It can also be readily extended to a hybrid mMIMO transceiver architecture, where multiple antennas are connected to a single RF chain. This architecture will further compress the existing observations, which are already compressed due to the reduced number of pilots. The current algorithm can be designed to alleviate the performance loss imposed by the hybrid architecture.

APPENDIX A

Posterior of $\mathbf{X}_k^{[:,m]}$: We derive the optimal distribution using (12) as follows.

$$\begin{aligned} \ln q(\mathbf{X}_k^{[:,m]}) &\stackrel{(a)}{=} \langle \ln p(\mathcal{Y}, \mathcal{X} | \boldsymbol{\theta}) \rangle_{q(\mathcal{X}) \setminus q(\mathbf{X}_k^{[:,m]})} + \bar{C} \\ &\stackrel{(b)}{=} \langle \ln p(\mathbf{Y}_k^{[:,m]} | \mathbf{X}_k^{[:,m]}, \sigma_k^2) \rangle \\ &\quad + \ln p(\mathbf{X}_k^{[:,m]} | \boldsymbol{\alpha}_k^k, \boldsymbol{\alpha}_l^\ell, \boldsymbol{\alpha}^c, \mathbf{z}_n, \forall n)_{q(\mathbf{Y}), q(\mathbf{z})} + \bar{C} \\ &\stackrel{(c)}{=} 2\sigma_k^{-2} \langle \mathbf{Y}_k^{[:,m]} \rangle^H \boldsymbol{\Phi} \mathbf{X}_k^{[:,m]} - \left(\mathbf{X}_k^{[:,m]} \right)^H \left(\sigma_k^{-2} \boldsymbol{\Phi}^H \boldsymbol{\Phi} + \boldsymbol{\Lambda}_k \right) \mathbf{X}_k^{[:,m]} \\ &\quad + \bar{C} = \ln \mathcal{N}_C \left(\boldsymbol{\mu}_k^{[:,m]}, \boldsymbol{\Sigma}_k \right). \end{aligned} \quad (49)$$

Here $q(\mathcal{X}) \setminus q(\mathbf{X}_k^{[:,m]})$ is the distribution $q(\mathcal{X})$ with $q(\mathbf{X}_k^{[:,m]})$ marginalized out, where $\boldsymbol{\mu}_k^{[:,m]}, \boldsymbol{\Sigma}_k$ and $\boldsymbol{\Lambda}_k$ are given by (17), (18) and (19), respectively. Equality (b) is obtained by considering terms dependent on $\mathbf{X}_k^{[:,m]}$. The expectation term $\langle \mathbf{Y}_k^{[:,m]} \rangle$ in (c) is due to variational approximation [26], wherein we take expectation over the hidden variable $\mathbf{Y}_k^{[:,m]}$.

Posterior of $\mathbf{z}_n^k, \mathbf{z}_n^\ell, \mathbf{z}_n^c$: Using the factorization in (12):

$$\begin{aligned} \ln q(\mathbf{z}_n^k, \mathbf{z}_n^\ell, \mathbf{z}_n^c) &\stackrel{(a)}{=} \langle \ln p(\mathcal{Y}, \mathcal{X} | \boldsymbol{\theta}) \rangle_{q(\mathcal{X}) \setminus q(\mathbf{z}_n)} + \bar{C} \stackrel{(b)}{=} \\ &\langle \sum_{k=1}^K \sum_{m=1}^M \ln [p(\mathbf{X}_k^{[n,m]} | \boldsymbol{\alpha}_{k[n]}^k, \boldsymbol{\alpha}_{l[n]}^\ell, \boldsymbol{\alpha}_{[n]}^c, \mathbf{z}_n) p(\mathbf{z}_n)] \rangle_{q(\mathbf{X}_k^{[n,m]})} + \bar{C} \end{aligned}$$

$$\begin{aligned}
&\stackrel{(c)}{=} \sum_{k=1}^K \sum_{m=1}^M \mathbf{z}_{[n]}^{\mathbf{k}} \{ \ln \rho_{[n]}^{\mathbf{k}} + (\ln \alpha_{k[n]}^{\mathbf{k}} - \alpha_{k[n]}^{\mathbf{k}} \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle) \} \\
&\quad + \mathbf{z}_{[n]}^{\ell} \{ \ln \rho_{[n]}^{\ell} + (\ln \alpha_{l[n]}^{\ell} - \alpha_{l[n]}^{\ell} \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle) \} \\
&\quad + \mathbf{z}_{[n]}^{\mathbf{c}} \{ \ln \rho_{[n]}^{\mathbf{c}} + (\ln \alpha_{[n]}^{\mathbf{c}} - \alpha_{[n]}^{\mathbf{c}} \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle) \} + \bar{C} \\
&\stackrel{(d)}{=} \mathbf{z}_{[n]}^{\mathbf{k}} \{ M \sum_{k=1}^K \ln \alpha_{k[n]}^{\mathbf{k}} + \ln \rho_{[n]}^{\mathbf{k}} \} + \mathbf{z}_{[n]}^{\ell} \{ M \sum_{k=1}^K \ln \alpha_{l[n]}^{\ell} \\
&\quad + \ln \rho_{[n]}^{\ell} \} + \mathbf{z}_{[n]}^{\mathbf{c}} \{ M \sum_{k=1}^K \ln \alpha_{[n]}^{\mathbf{c}} + \ln \rho_{[n]}^{\mathbf{c}} \} + \bar{C}.
\end{aligned}$$

In equality (a), $q(\mathcal{X}) \setminus q(\mathbf{z}_n)$ denotes $q(\mathcal{X})$ with $q(\mathbf{z}_n)$ removed. Equality (b) is derived by writing the terms dependent on \mathbf{z}_n . Equality (c) is obtained by dropping the terms independent of \mathbf{z}_n , by taking the expectation inside and by rearranging the terms. We get equality (d) by using i)

$$\begin{aligned}
&\sum_{k=1}^K \alpha_{k[n]}^{\mathbf{k}} \sum_{m=1}^M \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle = \sum_{l=1}^L \alpha_{l[n]}^{\ell} \sum_{k \in \mathcal{C}_l} \sum_{m=1}^M \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle \\
&= \alpha_{[n]}^{\mathbf{c}} \sum_{k=1}^K \sum_{m=1}^M \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle = KM, \tag{50}
\end{aligned}$$

which is obtained from the updates of $\alpha_{k[n]}^{\mathbf{k}}$, $\alpha_{l[n]}^{\ell}$, $\alpha_{[n]}^{\mathbf{c}}$ in Sec. ‘‘Hyperparameter updates’’ below; and ii) $\mathbf{z}_{[n]}^{\mathbf{k}} + \mathbf{z}_{[n]}^{\ell} + \mathbf{z}_{[n]}^{\mathbf{c}} = 1$. Using a uniform prior on \mathbf{z}_n , i.e., $\rho_{[n]}^{\mathbf{k}} = \rho_{[n]}^{\ell} = \rho_{[n]}^{\mathbf{c}} = 1/3$:

$$\begin{aligned}
\ln q(\mathbf{z}_n) &= \mathbf{z}_{[n]}^{\mathbf{k}} \sum_{k=1}^K \sum_{m=1}^M \ln \alpha_{k[n]}^{\mathbf{k}} + \mathbf{z}_{[n]}^{\ell} \sum_{k=1}^K \sum_{m=1}^M \ln \alpha_{l[n]}^{\ell} \\
&\quad + \mathbf{z}_{[n]}^{\mathbf{c}} \sum_{k=1}^K \sum_{m=1}^M \ln \alpha_{[n]}^{\mathbf{c}} + \bar{C}. \tag{51}
\end{aligned}$$

Thus, $q(\mathbf{z}_n)$ is a multinoulli distribution with posterior expectations $\langle \mathbf{z}_{[n]}^{\mathbf{k}} \rangle$, $\langle \mathbf{z}_{[n]}^{\ell} \rangle$, $\langle \mathbf{z}_{[n]}^{\mathbf{c}} \rangle$ given by (20)- (22), and $\eta_{[n]}^{\mathbf{k}}$, $\eta_{[n]}^{\ell}$, $\eta_{[n]}^{\mathbf{c}}$, given by (23), (24).

Hyperparameter updates: The M-step maximizes the expected complete log-likelihood with respect to hyperparameters, $\theta = \{\alpha_k^{\mathbf{k}}, \forall k \in \mathcal{C}_l, \alpha_l^{\ell}, \forall l, \alpha_{[n]}^{\mathbf{c}}\}$ i.e., $\hat{\theta} = \arg \max_{\theta} \langle \ln p(\mathcal{Y}, \mathcal{X} | \theta) \rangle_{q(\mathcal{X} | \theta^{\text{old}})}$. The hyperparameter $\alpha_{k[n]}^{\mathbf{k}} \stackrel{(a)}{=} \arg \max_{\alpha_{k[n]}^{\mathbf{k}}} \sum_{k=1}^K \sum_{m=1}^M \left[\langle \mathbf{z}_{[n]}^{\mathbf{k}} \rangle (\ln \alpha_{k[n]}^{\mathbf{k}} - \alpha_{k[n]}^{\mathbf{k}} \langle |\mathbf{X}_k^{[n,m]}|^2 \rangle) \right]$. Equality (a) is obtained by expanding $p(\mathcal{Y}, \mathcal{X} | \theta)$ and by ignoring the terms independent of $\alpha_{k[n]}^{\mathbf{k}}$. Using first order optimality, we get the first equation in (25). We similarly obtain $\alpha_{l[n]}^{\ell}, \forall l$ in the second equation in (25), and $\alpha_{[n]}^{\mathbf{c}}, \forall n$, in (26). Observe from (25), (26), that we get the relationship between $\alpha_{k[n]}^{\mathbf{k}}$, $\alpha_{l[n]}^{\ell}$ and $\alpha_{[n]}^{\mathbf{c}}$ given in (50), which is used to update $q(\mathbf{z}_n)$ in (51).

APPENDIX B

We commence by proving a proposition which will be used in the sequel.

Proposition 1. If the common support of the n th index at the l th PU is estimated as

$$\text{(a) } \hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 0, \text{ the sum } \sum_{j \in \hat{\mathcal{S}}_n^{-l}} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}} \text{ can be bounded as } \\
0 \leq \sum_{j \in \hat{\mathcal{S}}_n^{-l}} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}} \leq K_2 \frac{c}{\text{SNR}}, \text{ and; } \tag{52}$$

$$\text{(b) } \hat{\mathbf{s}}_{[n]}^{\mathbf{c},l} = 1, \text{ the user-specific variance } \gamma_{k[n]}^{\mathbf{k}} \text{ can be bounded as } \\
\frac{c}{\text{SNR}} \sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},j} \rangle \leq \sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}} < K_1 \max_{k \in \mathcal{C}_j, j \in \hat{\mathcal{S}}_n^l} (\gamma_{k[n]}^{\mathbf{k}}).$$

Proof of (a). We bound the total variance $\tilde{\gamma}_{k[n]}^{\mathbf{k},l} = 1 / (\langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \alpha_{k[n]}^{\mathbf{k}} + \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle \alpha_{l[n]}^{\ell} + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle \tilde{\alpha}_{[n]}^{\mathbf{c},l})$ as follows.

$$\begin{aligned}
&\langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \alpha_{k[n]}^{\mathbf{k}} + \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle \alpha_{l[n]}^{\ell} + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle \tilde{\alpha}_{[n]}^{\mathbf{c},l} \stackrel{(a)}{\leq} \\
&\max\{\alpha_{k[n]}^{\mathbf{k}}, \alpha_{l[n]}^{\ell}, \tilde{\alpha}_{[n]}^{\mathbf{c},l}\} \stackrel{(b)}{=} \alpha_{k[n]}^{\mathbf{k}} \implies \tilde{\gamma}_{k[n]}^{\mathbf{k},l} \geq \gamma_{k[n]}^{\mathbf{k}}. \tag{53}
\end{aligned}$$

Inequality (a) is because $\langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle + \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle = 1$. Inequality (b) uses $\tilde{\alpha}_{[n]}^{\mathbf{c},l} = \alpha_{l[n]}^{\ell} = \alpha_{k[n]}^{\mathbf{k}}$, which is obtained from (34) for $\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}} = 0$ and $L = K$. By combining (53) with $\tilde{\gamma}_{k[n]}^{\mathbf{k},l} \leq \frac{c}{\text{SNR}}$, obtained from (28) and $\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}} = 0$, we get $\gamma_{k[n]}^{\mathbf{k}} \leq \frac{c}{\text{SNR}}$. By summing over $j \in \hat{\mathcal{S}}_n^{-l}$ and $k \in \mathcal{C}_j$, we get the upper bound in (52). The lower limit in (52) is the trivial minimum, the variance $\gamma_{k[n]}^{\mathbf{k}}$ can take. \square

Proof of (b). We bound the total variance $\tilde{\gamma}_{k[n]}^{\mathbf{k},l} = 1 / (\langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \alpha_{k[n]}^{\mathbf{k}} + \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle \alpha_{l[n]}^{\ell} + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle \tilde{\alpha}_{[n]}^{\mathbf{c},l})$ as follows.

$$\begin{aligned}
&\langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \alpha_{k[n]}^{\mathbf{k}} + \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle \alpha_{l[n]}^{\ell} + \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{c},l} \rangle \tilde{\alpha}_{[n]}^{\mathbf{c},l} \stackrel{(a)}{\geq} \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle \alpha_{k[n]}^{\mathbf{k}} \\
&\implies \tilde{\gamma}_{k[n]}^{\mathbf{k},l} \leq \gamma_{k[n]}^{\mathbf{k}} / \langle \tilde{\mathbf{z}}_{[n]}^{\mathbf{k},l} \rangle. \tag{54}
\end{aligned}$$

Inequality (a) is obtained since LHS is a summation of three positive terms. By combining (54) and $\tilde{\gamma}_{k[n]}^{\mathbf{k},l} > \frac{c}{\text{SNR}}$, obtained from (28) and $\hat{\mathbf{s}}_{k[n]}^{\mathbf{k}} = 1$, and by summing over all the PUs $j \in \hat{\mathcal{S}}_n^l$ and users $k \in \mathcal{C}_j$, we get the lower bound in Proposition 1(b). We will also use a weaker and trivial lower bound i.e., $\sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}} \geq 0$, wherever needed. The upper bound in Proposition 1(b) is obtained by noting that $\gamma_{k[n]}^{\mathbf{k}} \leq \max_{k \in \mathcal{C}_j, j \in \hat{\mathcal{S}}_n^l} (\gamma_{k[n]}^{\mathbf{k}})$, and by summing all the PUs $j \in \hat{\mathcal{S}}_n^l$ and $k \in \mathcal{C}_j$. \square

Proof of Lemma 1: We begin by simplifying the expression of $\gamma_{[n]}^{\mathbf{c}}$ using (26):

$$\begin{aligned}
\gamma_{[n]}^{\mathbf{c}} &\stackrel{(a)}{=} \frac{1}{K} \left(\tilde{\gamma}_{[n]}^{\mathbf{c},l} \sum_{j' \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_{j'}| + \sum_{j \in \hat{\mathcal{S}}_n^{-l}} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}} \right) \\
&\stackrel{(b)}{=} \tilde{\gamma}_{[n]}^{\mathbf{c},l} - \frac{\tilde{\gamma}_{[n]}^{\mathbf{c},l}}{K} \sum_{j \in \hat{\mathcal{S}}_n^{-l}} |\mathcal{C}_j| + \frac{1}{K} \sum_{j \in \hat{\mathcal{S}}_n^{-l}} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}}. \tag{55}
\end{aligned}$$

Equality in (a) uses $\{1, 2, \dots, L\} = \hat{\mathcal{S}}_n^l \cup \hat{\mathcal{S}}_n^{-l}$, where $\hat{\mathcal{S}}_n^{-l} \triangleq \{1, 2, \dots, L\} \setminus \hat{\mathcal{S}}_n^l$ and (34). Equality in (b) is because $\sum_{j' \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_{j'}| + \sum_{j \in \hat{\mathcal{S}}_n^{-l}} |\mathcal{C}_j| = K$. We now use (55) to compute $\Delta^l_{\gamma_{[n]}^{\mathbf{c}}} = |\gamma_{[n]}^{\mathbf{c}} - \tilde{\gamma}_{[n]}^{\mathbf{c},l}|$:

$$\stackrel{(a)}{=} \frac{1}{K K_1} \left| K_1 \sum_{j \in \hat{\mathcal{S}}_n^{-l}} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^{\mathbf{k}} - K_2 \sum_{j' \in \hat{\mathcal{S}}_n^l} \sum_{k' \in \mathcal{C}_{j'}} \gamma_{k'[n]}^{\mathbf{k}'} \right|. \tag{56}$$

In equality (a), we use first equation in (34) to replace $\tilde{\gamma}_{[n]}^{\mathbf{c},l}$ and $K_1 = \sum_{j' \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_{j'}|$ and $K_2 = \sum_{j \in \hat{\mathcal{S}}_n^{-l}} |\mathcal{C}_j|$. To bound both terms in (56), we next use Proposition 1(a) and 1(b), to get (41).

Proof of Lemma 2(a): We prove this lemma for $\tilde{\eta}_{[n]}^{\mathbf{c},l}$, whose update from (37) is given as

$$\begin{aligned}
\tilde{\eta}_{[n]}^{\mathbf{c},l} &= \frac{MK}{K_1} \sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j| \ln \left(\frac{\sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j|}{\sum_{j \in \hat{\mathcal{S}}_n^l} \sum_{k \in \mathcal{C}_j} 1 / \alpha_{k[n]}^{\mathbf{k}}} \right) \\
&\stackrel{(a)}{=} MK \ln \tilde{\alpha}_{[n]}^{\mathbf{c},l}. \tag{57}
\end{aligned}$$

Equality in (a) is due to (34) and $\sum_{j \in \hat{\mathcal{S}}_n^l} |\mathcal{C}_j| = K_1$. We now calculate $\Delta^l_{\eta_{[n]}^{\mathbf{c}}} = |\eta_{[n]}^{\mathbf{c}} - \tilde{\eta}_{[n]}^{\mathbf{c},l}|$:

$$\begin{aligned} & \Delta_{\eta_{[n]}^c}^l \stackrel{(a)}{=} \left| MK \ln \alpha_{[n]}^c - MK \ln \tilde{\alpha}_{[n]}^{c,l} \right| \\ & \stackrel{(b)}{=} \left| MK \ln \left(1 + \frac{\sum_{j \in \hat{\mathcal{S}}_n^{l-1}} \sum_{k \in \mathcal{C}_j} \gamma_{k[n]}^k}{\sum_{j' \in \hat{\mathcal{S}}_n^l} \sum_{k' \in \mathcal{C}_{j'}} \gamma_{k'[n]}^k} \right) + MK \ln \frac{K_1}{K} \right|. \end{aligned} \quad (58)$$

Equality (a) uses second equation in (24), and (57). Equality (b) is obtained by using (26) and (34). We further use Proposition 1(a) and 1(b), to get (42).

Proof of Lemma 2(b): The absolute error for $\eta_{[n]}^k$, $\Delta_{\eta_{[n]}^k}^l = |\eta_{[n]}^k - \tilde{\eta}_{[n]}^{k,l}|$ can be bounded as $\Delta_{\eta_{[n]}^k}^l \stackrel{(a)}{=} MK_2 \left| \frac{1}{K_1} \sum_{j' \in \hat{\mathcal{S}}_n^l} \sum_{k' \in \mathcal{C}_{j'}} \ln \gamma_{k'[n]}^k - \frac{1}{K_2} \sum_{j \in \hat{\mathcal{S}}_n^{l-1}} \sum_{k \in \mathcal{C}_j} \ln \gamma_{k[n]}^k \right|$. (59)

Equality (a) is obtained by using (23) and (35) and $\gamma_{k[n]}^k = 1/\alpha_{k[n]}^k$. We next use

$$\ln \left(\min_{k \in \mathcal{C}_j, j \in \hat{\mathcal{S}}_n^{l-1}} \left(\gamma_{k[n]}^k \right) \right) \leq \ln \gamma_{k[n]}^k \leq \ln \frac{c}{\text{SNR}}, \quad \forall k \in \mathcal{C}_j, j \in \hat{\mathcal{S}}_n^{l-1}, \text{ and} \quad (60)$$

$$\ln \frac{c \langle \tilde{\mathbf{z}}_{[n]}^{k,j'} \rangle}{\text{SNR}} \leq \ln \gamma_{k'[n]}^k \leq \ln \left(\max_{k' \in \mathcal{C}_{j'}, j' \in \hat{\mathcal{S}}_n^l} \left(\gamma_{k'[n]}^k \right) \right), \quad \forall k' \in \mathcal{C}_{j'}, j' \in \hat{\mathcal{S}}_n^l, \quad (61)$$

which are derived from Proposition 1(a) and 1(b), to get (43).

Proof of Lemma 3: When $\hat{\mathbf{s}}_{[n]}^{c,l} = 0$, the l th PU updates $\tilde{\alpha}_{[n]}^{c,l}$, $\tilde{\eta}_{[n]}^{\ell,l}$ and $\langle \tilde{\mathbf{z}}_{[n]}^{c,l} \rangle$ as follows:

$$\tilde{\alpha}_{[n]}^{c,l} \stackrel{(a)}{=} \alpha_{[n]}^{\ell}; \quad \tilde{\eta}_{[n]}^{\ell,l} \stackrel{(b)}{=} \tilde{\eta}_{[n]}^{c,l}; \quad \langle \tilde{\mathbf{z}}_{[n]}^{c,l} \rangle \stackrel{(c)}{=} \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle; \quad (62)$$

$$\tilde{\Lambda}_{[n]}^{[n,n]} \stackrel{(d)}{=} \langle \tilde{\mathbf{z}}_{[n]}^{k,l} \rangle \alpha_{k[n]}^k + 2 \langle \tilde{\mathbf{z}}_{[n]}^{\ell,l} \rangle \alpha_{[n]}^{\ell}. \quad (63)$$

Equality (a) is due to (34), equality (b) is due to (39), and equality (c) is obtained by substituting equality in (b) above, in (22). Equality (d) is obtained by using equalities (a), (b) and (c) of (19) to calculate the $[n, n]$ th entry of prior precision parameter. We notice from (62) that the common global hyperparameters $\tilde{\alpha}_{[n]}^{c,l}$, $\tilde{\eta}_{[n]}^{c,l}$ and $\langle \tilde{\mathbf{z}}_{[n]}^{c,l} \rangle$ becomes redundant. They, thus, get removed from the algorithm for the n th index. The prior precision in (63) thus corresponds to the GM prior given in (44) over $\mathbf{X}_k^{[n,m]}$, $\forall m$ and for $k \in \mathcal{C}_l$.

REFERENCES

- [1] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.
- [2] W. Chang, H.-W. Chan, and Y.-K. Hua, "Weighted graph coloring based softer pilot reuse for TDD massive MIMO systems," *IEEE Trans. Veh. Commun.*, vol. 67, no. 7, pp. 6272–6285, 2018.
- [3] X. Cheng, J. Sun, and S. Li, "Channel estimation for FDD multi-user massive MIMO: A variational Bayesian inference-based approach," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7590–7602, Nov 2017.
- [4] X. Rao and V. K. N. 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.
- [5] C. C. Tseng, J. Y. Wu, and T. S. Lee, "Enhanced compressive downlink CSI recovery for FDD massive MIMO systems using weighted block ℓ_1 -minimization," *IEEE Trans. Commun.*, vol. 64, no. 3, pp. 1055–1067, Mar 2016.
- [6] J. Dai, A. Liu, and V. K. N. Lau, "Joint channel estimation and user grouping for massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 67, no. 3, pp. 622–637, Feb 2019.
- [7] P. Liang, J. Fan, W. Shen, Z. Qin, and G. Y. Li, "Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9217–9222, 2020.
- [8] P. Zhao, K. Ma, Z. Wang, and S. Chen, "Virtual angular-domain channel estimation for FDD based massive MIMO systems with partial orthogonal pilot design," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5164–5178, 2020.
- [9] Y. Han, Q. Liu, C. Wen, S. Jin, and K. Wong, "FDD massive MIMO based on efficient downlink channel reconstruction," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4020–4034, 2019.
- [10] Q. He, T. Q. S. Quek, Z. Chen, Q. Zhang, and S. Li, "Compressive channel estimation and multi-user detection in C-RAN with low-complexity methods," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3931–3944, 2018.
- [11] M. Ke, Z. Gao, Y. Wu, X. Gao, and K. Wong, "Massive access in cell-free massive MIMO-based internet of things: Cloud computing and edge computing paradigms," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2020.
- [12] S. Khanna and C. R. Murthy, "Decentralized joint-sparse signal recovery: A sparse Bayesian learning approach," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 29–45, Mar 2017.
- [13] S. Khanna and C. R. Murthy, "Communication-efficient decentralized sparse Bayesian learning of joint sparse signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 617–630, Sep 2017.
- [14] Y.-S. Jeon, H. Do, S.-N. Hong, and N. Lee, "Soft-output detection methods for sparse millimeter-wave mimo systems with low-precision adcs," *IEEE Trans. Commun.*, vol. 67, no. 4, pp. 2822–2836, 2019.
- [15] Y. Jeon, N. Lee, and H. V. Poor, "Robust data detection for MIMO systems with one-bit adcs: A reinforcement learning approach," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 3, pp. 1663–1676, 2020.
- [16] J. Mo, P. Schniter, and R. W. H. Jr., "Channel estimation in broadband millimeter wave MIMO systems with few-bit adcs," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1141–1154, 2018.
- [17] X. Cheng, K. Xu, J. Sun, and S. Li, "Adaptive grouping sparse bayesian learning for channel estimation in non-stationary uplink massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4184–4198, Aug 2019.
- [18] Y. Ding, S. Chiu, and B. D. Rao, "Bayesian channel estimation algorithms for massive MIMO systems with hybrid analog-digital processing and low-resolution ADCs," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 3, pp. 499–513, 2018.
- [19] K. Li, R. R. Sharan, Y. Chen, T. Goldstein, J. R. Cavallaro, and C. Studer, "Decentralized baseband processing for massive MU-MIMO systems," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 4, pp. 491–507, Dec 2017.
- [20] E. Bertilsson, O. Gustafsson, and E. G. Larsson, "A scalable architecture for massive MIMO base stations using distributed processing," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, Nov 2016, pp. 864–868.
- [21] J. Rodríguez Sánchez, F. Rusek, O. Edfors, M. Sarajlić, and L. Liu, "Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms," *IEEE Trans. Signal Process.*, vol. 68, pp. 687–700, 2020.
- [22] L. Van der Perre, L. Liu, and E. G. Larsson, "Efficient DSP and circuit architectures for massive MIMO: State of the art and future directions," *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4717–4736, 2018.
- [23] L. Rugini and P. Banelli, "On the equivalence of maximum snr and mmse estimation: Applications to additive non-gaussian channels and quantized observations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6190–6199, 2016.
- [24] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, Jul 2007.
- [25] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. NY, USA: Cambridge University Press, 2005.
- [26] A. Adhikary, E. Al Safadi, M. K. Samimi, R. Wang, G. Caire, T. S. Rappaport, and A. F. Molisch, "Joint spatial division and multiplexing for mm-wave channels," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1239–1255, June 2014.
- [27] X. Mestre, "Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5113–5129, Nov 2008.
- [28] A. M. Sayeed, "Deconstructing multiantenna fading channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2563–2579, 2002.
- [29] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, June 2010.
- [30] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

- [31] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [32] Z. Zhang, Z. Chen, M. Shen, B. Xia, W. Xie, and Y. Zhao, "Performance analysis for training-based multipair two-way full-duplex relaying with massive antennas," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6130–6145, 2017.