Compressed Training Based Massive MIMO

Baki Berkay Yilmaz, *Student Member* and, Alper T. Erdogan *Senior Member*

***Abstract*— Massive Multiple-Input-Multiple-Output (MIMO) scheme promises high spectral efficiency through the employment of large scale antenna arrays in base stations. In Time Division Duplexed (TDD) implementations, co-channel mobile terminals transmit training information such that base stations can estimate and exploit channel state information (CSI) to spatially multiplex these users. In the conventional approach, the optimal choice for training length was shown to be equal to the number of users, K. In this article, we propose a new semi-blind framework, named as "MIMO Compressed Training", which utilizes information symbols in addition to training symbols for adaptive spatial multiplexing. We show that this framework enables us to reduce (compress) the training length down to a value close to $\log_2(K)$, i.e., the logarithm of the number of users, without any sparsity assumptions on the channel matrix. We also derive a prescription for the required packet length for proper training. The framework is built upon some convex optimization settings which enable efficient and reliable algorithm implementations. The numerical experiments demonstrate the strong potential of the proposed approach in terms of increasing the number of users per cell and improving the link quality.**

***Index Terms*— Massive MIMO, Compressed Training**

I. Introduction

Steadily increasing demand and ever growing number of applications with flexible connectivity requirements have been the major driving forces for the evolution of wireless communication technologies. Massive MIMO is a recently introduced approach targeting high spectral efficiency empowered by the use of large scale antenna arrays at base stations [1]–[3]. The deployment of hundreds, even thousands of antennas in base stations is expected to yield near optimal multi-user data rates through simple linear transceiver processing algorithms [4].

In Time Division Duplexed (TDD) Massive MIMO, which is the focus of this article, uplink and downlink share the same frequency band, and their transmissions occur in non-overlapping time intervals. Fig. 1 outlines the generic protocol proposed for the TDD-Massive MIMO scheme [5]. According to this figure, uplink transmission and a short duration of base station processing are immediately followed by downlink transmission.

The basic benefit of the TDD scheme is the assumed reciprocity of the downlink and uplink channels, which is utilized by the base station to generate conjugate beamforming based on the channel state information (CSI) estimated during the uplink transmission phase. This eliminates the need for the complex CSI estimation at the mobile terminals. However, it is still a major task for the base station to estimate CSI for effective spatial multiplexing of mobile users. For this purpose, as illustrated in Fig. 1, a block of training symbols is embedded in the uplink transmission packet.

Fig. 1: TDD-Massive MIMO Protocol.

The selection of training length has been a fundamental problem in wireless communications research. In [6], a mutual information maximization based approach is used to address this question for point-to-point MIMO systems. As an important result, when training and data powers are allowed to be independently adjusted, the optimal training length is prescribed to be equal to the number of transmit antennas, N_T. For the multi-user massive MIMO protocol outlined in Fig. 1, where the base station estimates the channel using the uplink training data, the training length is required to be greater than or equal to the number of users [2], [7]. In [8], it is shown that this choice of training length is still optimal under pilot contamination caused by the users in other cells.

We should note that for both prescriptions on the optimal training length, namely *the number of transmit antennas* for point-to-point MIMO systems and *the number of user terminals* in Massive MIMO scheme, it is assumed that the channel estimate is based on only the training region. In this article, we show that by utilizing the uplink data section in addition to the uplink training, we can reduce training length to a value near *the logarithm of the number of user terminals*, $log_2(K)$ (or equivalently *the logarithm of the transmit antennas* in the point-to-point MIMO, $\log_2(N_T)$). For this purpose, we expand the recently introduced "Compressed Training" framework to cover MIMO communication systems.

Compressed Training was initially introduced for frequency selective Single-Input-Multiple-Output channels in [9], [10] as the semi-blind extension of the convex optimization based blind approaches in [11]–[14]. In this approach, to train the equalizer, an adaptive scheme based on a convex cost function measuring the least squares (LS) error in the training region and the infinity norm of the equalizer outputs for the whole packet was proposed. The former (LS) part of the cost measures training reconstruction performance, whereas the second part, i.e., the peak magnitude of equalizer output is shown to be a reflector for the sparseness of the overall channel (communication-equalizer channel combination) impulse response. It was shown that this approach can reduce the training length to a value close to the logarithm of equalizer length per receiver branch (or the logarithm of the channel spread).

In this article, we introduce the compressed training framework for MIMO flat fading channels. Instead of estimating

Baki Berkay Yilmaz is with the Electrical and Computer Engineering Department, Georgia Institute of Technology, Atlanta, GA, 30332 USA (e-mail: b.berkayyilmaz@gatech.edu).

Alper T. Erdogan is with the Electrical-Electronics Engineering Dept., Koc University, Istanbul, 34450, Turkey (e-mail: alperdogan@ku.edu.tr).

This work is supported in part by TUBITAK 112E057 project.

the channel matrix, we propose to obtain the linear equalizer/separator directly using both training and data regions. The corresponding algorithm exploits the special rectangular QAM constellation structure to utilize whole packet symbols in training the equalizer matrix. This is established by factoring the peak magnitude of the equalizer magnitude over the whole packet as an additional cost besides the training reconstruction cost used in conventional methods. The equalizer matrix obtained through the compressed training based adaptation can also be used as the conjugate beamformer for the downlink transmission. Again, the proposed "MIMO Compressed Training" approach does not make any assumption on the sparsity of the channel matrix. Therefore, it is applicable to both sparse and dense channel matrix scenarios. The initial results about this framework were presented in the conference article [15].

We should note that there exist other semi-blind MIMO adaptive approaches that exploit data region along with the training region mostly to estimate the channel matrix. As an example, [16] proposes a semi-blind MIMO channel estimation algorithm, where the channel matrix is modeled as the product of a whitening matrix and a unitary matrix. The whitening part is estimated blindly based on the whole data packet, whereas the unitary part is determined by the use of training symbols. As a more recent reference, [17] proposes another semi-blind channel estimation approach for Massive MIMO systems based on Gaussian priors on the unknown data symbols. As an example for the direct adaptation of the equalizer coefficients, in [18], the authors propose a semi-blind MIMO equalization approach based on the mixture of constant modulus and soft decision directed algorithms.

We can list main distinguishing features of the proposed approach, especially relative to the existing approaches, as follows:

- The proposed compressed training MIMO scheme is based on some convex optimization settings. This is an important feature with significant implications such as
  - Adaptive algorithms based on non-convex cost functions suffer from ill-convergence or mis-convergence problems caused by undesired local minima and slow convergence issues due to the existence of saddle points. Fortunately, having a convex cost function eliminates such concerns.
  - Convex cost functions enable efficient adaptive implementations. As an important connection, recent results for developing low complexity and parallel convex algorithms for "big-data" (see [19] for a recent review) are potentially applicable to develop real-time algorithms to be implemented in the base stations.
  - By employing random matrix theory in conjunction with the convex optimization settings [20], [21], it is possible to provide effective analysis of the proposed algorithms. In fact, we are able to show the existence of "phase transitions" for the choices of both training length and packet length parameters, and therefore, provide prescriptions for them. In particular, we can obtain the concrete result that the training length can be reduced from being proportional to "the number of user terminals" to its logarithm. This is an explicit result highlighting the potential gain of the proposed method especially in terms of significantly increasing the number of mobile users.
- The proposed approach has direct links with the compressed sensing concept due to the duality between ℓ_1 and ℓ_∞ norms. This link can be utilized to adapt rich algorithmic and analysis contributions in sparsity driven research to compressive training framework.

We note that the convex optimization based algorithms proposed in the article act as initial eye openers. Once decisions become reliable, these algorithms can be extended to non-convex settings to incorporate decisions for further performance improvement.

The article is organized as follows: Section II introduces the data setting for the Massive MIMO uplink connection. We introduce the proposed Compressed Training Massive MIMO approach in Section III. Various algorithm extensions to address different issues, such as the impact of noise/short packet lengths, and the acceleration of algorithm convergence, are introduced in Section IV. In Section V, numerical experiment results illustrating the potential of the proposed approach are provided. Finally, Section VI is the conclusion.

Notation: Following describes the notation of the article: Let $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, $\boldsymbol{x} \in \mathbb{C}^n$ and $\mathcal{S}$ is a set:

Notation	Meaning
$\boldsymbol{A}_{m,n}$	The element of $\boldsymbol{A}$ at the index (m,n)
$\boldsymbol{A}^H$	Conjugate-Transpose of $\boldsymbol{A}$
$\mathbb{P}(\bullet)$	Probability of a given event
$\boldsymbol{I}$	Identity matrix with proper size
$\|\boldsymbol{A}\|_F$	Frobenius norm of $\boldsymbol{A}$
$\mathrm{diag}(\boldsymbol{A})$	A vector containing the diagonal entries of $\boldsymbol{A}$
$\mathrm{idiag}(\boldsymbol{x})$	A diagonal matrix generated from vector $\boldsymbol{x}$
$\Re e\{\boldsymbol{A}\}$	Real part of matrix $\boldsymbol{A}$
$\mathcal{I}m\{\boldsymbol{A}\}$	Imaginary part of matrix $\boldsymbol{A}$
$\mathrm{sign}(\boldsymbol{A})$	Matrix obtained from $\boldsymbol{A}$ by replacing elements with $e^{j\theta_{mn}}$ where θ_{mn} is the phase of $\boldsymbol{A}_{m,n}$
$\boldsymbol{A}_{:,k}$ ($\boldsymbol{A}_{k,:}$)	k^{th} column (row) of $\boldsymbol{A}$
$\boldsymbol{e}_k$	Standard basis (column) vector with all zero elements except k^{th} element is equal to 1
$\boldsymbol{Co}\mathcal{S}$	Convex Hull of the set $\mathcal{S}$
$N_L(\boldsymbol{A})$	Left null space of $\boldsymbol{A}$

II. Massive MIMO Data Setting

We consider the uplink training scenario for the Massive MIMO systems. We assume flat fading channel model (which can be considered as one of the OFDM channels in the frequency selective case). Fig. 2 illustrates the baseband equivalent data model for the uplink transmission for a wireless multi user MIMO system. In this model:

- There are K user terminals.
- The uplink transmission packet consists of τ_D data symbols followed by τ_T training symbols. Therefore, the total package length is $\Gamma = \tau_D + \tau_T$. We define $\mathcal{I} = \{1, 2, \ldots, \Gamma\}$ as the index set of the uplink transmission package.
- $\{s_l(n) : n \in \mathcal{I}\}$ represents the uplink sequence transmitted by the l^{th} user's terminal, where $l \in \{1, 2, \ldots, K\}$. The uplink data sequence is the subset $\{s_l(n) : n \in$

$\{1, 2, \ldots, \tau_D\}\}$ and the uplink training sequence is the subset $\{s_l(n) : n \in \{\tau_D + 1, \tau_D + 2, \ldots, \Gamma\}\}$. We assume that the uplink sequence samples are taken from a constellation with unity average power.

- We define the source vector as $\boldsymbol{s}(n) = \begin{bmatrix} s_1(n) & \ldots & s_K(n) \end{bmatrix}^T$ for $n \in \mathcal{I}$.
- We define the uplink transmission sequence matrix for all user terminals as $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}(1) & \boldsymbol{s}(2) & \ldots & \boldsymbol{s}(\Gamma) \end{bmatrix}$. We define the uplink training sequence matrix for all user terminals as the submatrix $\boldsymbol{S}_T = \begin{bmatrix} \boldsymbol{s}(\tau_D + 1) & \ldots & \boldsymbol{s}(\Gamma) \end{bmatrix}$.
- M represents the number of base station antennas.
- $\{y_l(n) : n \in \mathcal{I}\}$ is the sequence received at the l^{th} base station antenna, where $l = 1, \ldots, M$. We define the uplink signal vector received at the base station as $\boldsymbol{y}(n) = \begin{bmatrix} y_1(n) & \ldots & y_M(n) \end{bmatrix}^T$, for $n \in \mathcal{I}$.
- The base station vector $\boldsymbol{y}(n)$ is related to source vector $\boldsymbol{s}(n)$ via $\boldsymbol{y}(n) = \boldsymbol{H}\boldsymbol{s}(n) + \boldsymbol{v}(n)$ where $\boldsymbol{H}$ is the $M \times K$ channel matrix, which is assumed to be full rank, and $\boldsymbol{v}(n) = \begin{bmatrix} v_1(n) & \ldots & v_M(n) \end{bmatrix}^T$ is the noise vector. The noise components are assumed to be zero mean i.i.d. random variables with variance σ_v^2.
- The matrix containing all received sequences at the base station is defined as $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(1) & \boldsymbol{y}(2) & \ldots & \boldsymbol{y}(\Gamma) \end{bmatrix}$. Its submatrix corresponding to the uplink training region is represented as $\boldsymbol{Y}_T = \begin{bmatrix} \boldsymbol{y}(\tau_D + 1) & \ldots & \boldsymbol{y}(\Gamma) \end{bmatrix}$.

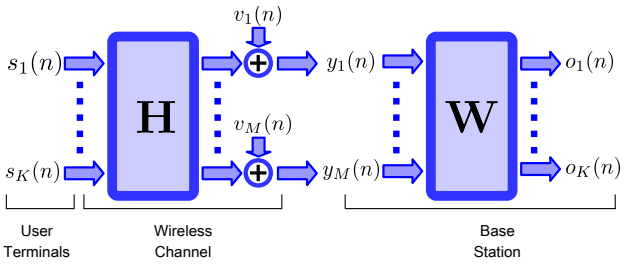


Fig. 2: Uplink Data Model.

- The base station employs a $K \times M$ equalizer matrix $\boldsymbol{W}$ to compensate the effects of the channel and separate individual user signals. The output of the equalizer is represented with $\boldsymbol{o}(n) = \boldsymbol{W}\boldsymbol{y}(n)$ where $\boldsymbol{o}(n) = \begin{bmatrix} o_1(n) & \ldots & o_K(n) \end{bmatrix}^T$.
- The matrix containing all output sequences at the base station is defined as $\boldsymbol{O} = \begin{bmatrix} \boldsymbol{o}(1) & \boldsymbol{o}(2) & \ldots & \boldsymbol{o}(\Gamma) \end{bmatrix}$.
- $\boldsymbol{G}$ is the cascade of the equalizer and the channel such that $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{H}$.

III. Compressed Training Based Massive MIMO

The major task of the base station is to obtain the equalizer matrix $\boldsymbol{W}$ using the observations at the antennas and the known training sequences transmitted by the users. In this section, we introduce Compressed Training based approach for obtaining $\boldsymbol{W}$. In Section III-A, we start by introducing the conventional adaptive algorithm. Then, we will motivate for the compressed training approach in Section III-B. The compressed training approach is introduced in Section III-C.

A. Conventional Adaptive Approach

The conventional method for adaptive processing at the base station receiver is to use the training region of the received samples $\boldsymbol{Y}_T$ and the training sequence samples $\boldsymbol{S}_T$. The conventional least squares estimate for $\boldsymbol{H}$ can be written as

$$\hat{\boldsymbol{H}} = \underset{\boldsymbol{H}}{arg\,min} \|\boldsymbol{Y}_T - \boldsymbol{H}\boldsymbol{S}_T\|_F.$$

Here, given that $\boldsymbol{S}_T$ is full rank and fat, which implies training length (τ_T) is greater than or equal to the number of users (K), the least squares estimate can be more explicitly written as

$$\hat{\boldsymbol{H}} = \boldsymbol{Y}_T {\boldsymbol{S}_T}^{\dagger},$$

where ${\boldsymbol{S}_T}^{\dagger} = {\boldsymbol{S}_T}^H(\boldsymbol{S}_T{\boldsymbol{S}_T}^H)^{-1}$. If the rows of $\boldsymbol{S}_T$, i.e., the training sequences of individual users, are orthonormal, i.e., $\boldsymbol{S}_T{\boldsymbol{S}_T}^H = \boldsymbol{I}$, then the least squares channel estimate simplifies to $\hat{\boldsymbol{H}} = \boldsymbol{Y}_T{\boldsymbol{S}_T}^H$ which amounts to taking inner products of the sequences received at base station antennas with the training sequences of the users. From the estimate $\hat{\boldsymbol{H}}$, it is possible to construct $\boldsymbol{W}$ through different approaches:

- *Match Filtering:* Choose $\boldsymbol{W} = \hat{\boldsymbol{H}}^H$, i.e., as the match filter to maximize Signal to Noise Ratio,
- *Zero Forcing (ZF):* Choose $\boldsymbol{W} = (\hat{\boldsymbol{H}}^H\hat{\boldsymbol{H}})^{-1}\hat{\boldsymbol{H}}^H$ to generate zero Inter User Interference (IUI) (in case of perfect channel estimate),
- *Minimum Mean Square Error (MMSE):* Choose $\boldsymbol{W} = (\sigma_v^2\boldsymbol{I} + \hat{\boldsymbol{H}}^H\hat{\boldsymbol{H}})^{-1}\hat{\boldsymbol{H}}^H$ to minimize the average energy of residual IUI and noise.

The alternative to this two step approach is to obtain $\boldsymbol{W}$ directly at one step, potentially through the least squares formulation:

$$\boldsymbol{W} = \underset{\tilde{\boldsymbol{W}}}{arg\,min} \|\tilde{\boldsymbol{W}}\boldsymbol{Y}_T - \boldsymbol{S}_T\|_F. \tag{1}$$

Note that the optimization in (1) may have infinitely many solutions. If we look at the noiseless case, we can rewrite the least squares cost as

$$\|\tilde{\boldsymbol{W}}\boldsymbol{Y}_T - \boldsymbol{S}_T\|_F = \|\tilde{\boldsymbol{W}}\boldsymbol{H}\boldsymbol{S}_T - \boldsymbol{S}_T\|_F = \|(\tilde{\boldsymbol{W}}\boldsymbol{H} - \boldsymbol{I})\boldsymbol{S}_T\|_F.$$

In such a case, if $\boldsymbol{S}_T$ is a full rank "fat" matrix, i.e.,

- $\tau_T \geq K$, i.e., the training length is greater than or equal to the number of users, and
- the rows of $\boldsymbol{S}_T$, are linearly independent,

then it is guaranteed that only the left inverses of $\boldsymbol{H}$ are the optimal solutions, for any full rank $\boldsymbol{H}$. In such a case, the cascade of the equalizer and the channel would be

$$\boldsymbol{G} = \boldsymbol{W}\boldsymbol{H} = \boldsymbol{I}. \tag{2}$$

B. Motivation for the Compressed Training Approach

In the compressed training approach, the goal is to reduce the training length τ_T below the required value K as discussed in the previous section. Of course, if we reduce τ_T below K, then the set of solutions for the linear system of equations

$$\boldsymbol{W}\boldsymbol{Y}_T = \boldsymbol{S}_T \tag{3}$$

is strictly larger than the set of left inverses of $\boldsymbol{H}$ satisfying (2). Therefore, in case $\tau_T < K$, we need some intelligent processing that would pick a $\boldsymbol{W}$ which is a left inverse of $\boldsymbol{H}$.

Since each row of the $\boldsymbol{G}$ matrix corresponding to the perfect equalization condition has all zeros except for one index location, we can restate the condition for the desired $\boldsymbol{W}$ as the one with the corresponding $\boldsymbol{G}$ has rows that are as sparse as possible. It is therefore sensible to choose a $\boldsymbol{W}$ which minimizes the ℓ_1-norm of each row of $\boldsymbol{G}$. Therefore, we can now cast the desired problem of obtaining a perfect equalizer (for the k^{th} user branch) as

$$\begin{aligned} \textit{Setting Ia}: \quad \underset{\boldsymbol{W}_{k,:}}{\text{minimize}} \quad & \|\Omega(\boldsymbol{G}_{k,:})\|_1 \\ \text{s.t.} \quad & \boldsymbol{W}_{k,:}\boldsymbol{Y}_T = \boldsymbol{S}_{Tk,:} \\ & \boldsymbol{G}_{k,:} = \boldsymbol{W}_{k,:}\boldsymbol{H}, \end{aligned}$$

for $k = 1, \ldots, K$. Here, the $\Omega(\cdot)$ over a complex vector $\boldsymbol{w}$ is defined as the isomorphic mapping

$$\Omega(\boldsymbol{w}^T) = \left[\begin{array}{cc} \Re e\{\boldsymbol{w}^T\} & -\mathcal{I}m\{\boldsymbol{w}^T\} \end{array} \right]$$

if its argument is a row vector, and otherwise

$$\Omega(\boldsymbol{w}) = \left[\begin{array}{cc} \Re e\{\boldsymbol{w}^T\} & \mathcal{I}m\{\boldsymbol{w}^T\} \end{array} \right]^T.$$

For the rest of the discussion, if isomorphic mapping is applied to a vector, the output will be denoted by its "˜" version, i.e. $\boldsymbol{w} \overset{\Omega(\bullet)}{\rightarrow} \tilde{\boldsymbol{w}}$. Since the optimization variable is $\boldsymbol{W}_{k,:}$, we can rewrite corresponding setting as

$$\begin{aligned} \textit{Setting Ib:} \quad \underset{\boldsymbol{W}_{k,:}}{\text{minimize}} \quad & \|\Omega(\boldsymbol{W}_{k,:}\boldsymbol{H})\|_1 \\ \text{s.t.} \quad & \boldsymbol{W}_{k,:}\boldsymbol{Y}_T = \boldsymbol{S}_{Tk,:}. \end{aligned}$$

The goal of this optimization process is to pick a $\boldsymbol{W}_{k,:}$ from the set of all possible $\boldsymbol{W}_{k,:}$'s that would satisfy the training reconstruction requirement in the constraint such that the resulting $\boldsymbol{G}_{k,:}$ is sparsest. However, it is clear that to solve *Setting Ia-b*, we need to know $\boldsymbol{H}$, which seems to be an irrational assumption, because if we knew what $\boldsymbol{H}$ was, the proposed adaptive approach would not be needed.

Now, we will describe a nice work around by replacing the cost function in *Setting Ia-b* with a practically implementable function that does not require the knowledge of $\boldsymbol{H}$. We start with the observation that real part of the k^{th} equalizer output can be written as

$$\begin{aligned} \Re e\{o_k(n)\} &= [\Re e\{\boldsymbol{G}_{k,:}\} \ -\mathcal{I}m\{\boldsymbol{G}_{k,:}\}] \left[\begin{array}{c} \Re e\{\boldsymbol{s}(n)\} \\ \mathcal{I}m\{\boldsymbol{s}(n)\} \end{array} \right] \\ &= \Omega\{\boldsymbol{G}_{k,:}\}\tilde{\boldsymbol{s}}(n). \end{aligned} \tag{4}$$

For the time being, lets assume that all user terminals use the complex β-QAM constellation

$$\begin{aligned} \mathcal{C}_{QAM}^{\beta} = \{a + ib : a, b \in \ & \{-\sqrt{\beta}+1, -\sqrt{\beta}+3, \ldots, \\ & \sqrt{\beta}-3, \sqrt{\beta}-1\}\}. \end{aligned} \tag{5}$$

where $\beta \in \mathcal{Z}_+$. This choice implies that $\|\tilde{\boldsymbol{s}}(n)\|_\infty \leq (\sqrt{\beta}-1)$

The following theorem establishes the link between the peak magnitude output component and the sparsity of the corresponding row of $\boldsymbol{G}$:

Theorem I: For a given $\boldsymbol{G}_{k,:}$, if there exists an $m \in \{1, \ldots, \tau\}$ such that

$$\tilde{\boldsymbol{s}}(m) = (\sqrt{\beta}-1)\text{sign}(\Omega(\boldsymbol{G}_{k,:}))^T, \tag{6}$$

then

$$\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty = (\sqrt{\beta}-1)\|\tilde{\boldsymbol{G}}_{k,:}\|_1. \tag{7}$$

Proof: Applying the Hölder Inequality on the inner product expression in (4) yields

$$\begin{aligned} |\Re e\{o_k(n)\}| &\leq \|\Omega\{\boldsymbol{G}_{k,:}\}\|_1 \left\| \begin{array}{c} \Re e\{\boldsymbol{s}(n)\} \\ \mathcal{I}m\{\boldsymbol{s}(n)\} \end{array} \right\|_\infty \\ &= \|\tilde{\boldsymbol{G}}_{k,:}\|_1 \|\tilde{\boldsymbol{s}}(n)\|_\infty. \end{aligned} \tag{8}$$

Therefore, the inequality in (8) takes the form $|\Re e\{o_k(n)\}| \leq (\sqrt{\beta}-1)\|\tilde{\boldsymbol{G}}_{k,:}\|_1$. Applying the same procedure for the imaginary part of the equalizer output, we obtain $|\mathcal{I}m\{o_k(n)\}| \leq (\sqrt{\beta}-1)\|\tilde{\boldsymbol{G}}_{k,:}\|_1$. This inequality further implies that

$$\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty \leq (\sqrt{\beta}-1)\|\tilde{\boldsymbol{G}}_{k,:}\|_1, \tag{9}$$

i.e., the peak absolute value for the imaginary and real components of k^{th} user's equalizer output sequence is bounded by the scaled version of the ℓ_1-norm of the corresponding row of $\boldsymbol{G}$. Now, we'll show that, through an assumption, we can convert this inequality into an equality so that we can replace the cost function $\|\tilde{\boldsymbol{G}}_{k,:}\|_1$ in *Setting Ia-b* with an expression which is a function of the equalizer outputs. For a given $\boldsymbol{G}$, suppose that there exists an index $m \in \mathcal{I}$, for which

$$\tilde{\boldsymbol{s}}(m) = (\sqrt{\beta}-1) \left[\begin{array}{c} \text{sign}(\Re e\{\boldsymbol{G}_{k,:}^H\}) \\ \text{sign}(\mathcal{I}m\{\boldsymbol{G}_{k,:}^H\}) \end{array} \right], \tag{10}$$

then the corresponding equalizer output would be equal to

$$\begin{aligned} \Re e\{o_k(m)\} &= \sum_{l=1}^{2K} (\sqrt{\beta}-1)\text{sign}(\tilde{G}_{k,l})\tilde{G}_{k,l} \\ &= (\sqrt{\beta}-1)\sum_{l=1}^{2K} |\tilde{G}_{k,l}| = (\sqrt{\beta}-1)\|\tilde{\boldsymbol{G}}_{k,:}\|_1. \end{aligned}$$

This implies that upper bound in (9) is indeed achieved under this assumption. The same conclusion would hold for the negative of the vector given in (10). Moreover, with a proper choice for $\tilde{\boldsymbol{s}}(m)$, we can achieve the upper bound in (9) with equality for $\mathcal{I}m\{o_k(m)\}$ as well. ■

As a result, we reach the following conclusion: if the uplink transmission sequence matrix $\tilde{\boldsymbol{S}}$ contains all possible (scaled) sign patterns in the form (10), then $\frac{1}{\sqrt{\beta}-1}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty$ would be an adaptively realizable replacement for the cost function $\|\tilde{\boldsymbol{G}}_{k,:}\|_1$ in *Setting Ia*. In other words, the peak absolute equalizer output is an observable measure of the sparsity of the $\tilde{\boldsymbol{G}}_{k,:}$ that can be utilized in the adaptive applications. Based on these observations, we can rewrite the *Setting Ia* as

$$\begin{aligned} \textit{Setting IIa:} \quad \underset{\boldsymbol{W}_{k,:}}{\text{minimize}} \quad & \frac{1}{\sqrt{\beta}-1}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty \\ \text{s.t.} \quad & \boldsymbol{W}_{k,:}\boldsymbol{Y}_T = \boldsymbol{S}_{Tk,:} \\ & \boldsymbol{O}_{k,:} = \boldsymbol{W}_{k,:}\boldsymbol{Y}, \end{aligned}$$

for $k = 1, \ldots, K$. We can write it in terms of $\boldsymbol{W}_{k,:}$ and known/observed parameters as

$$\textit{Setting IIb:} \quad \underset{\boldsymbol{W}_{k,:}}{\text{minimize}} \quad \frac{1}{\sqrt{\beta}-1}\|\Omega(\boldsymbol{W}_{k,:}\boldsymbol{Y})\|_\infty \quad \text{s.t.} \quad \boldsymbol{W}_{k,:}\boldsymbol{Y}_T = \boldsymbol{S}_{Tk,:}.$$

To conclude, if we analyze the *Setting IIb* carefully:

- The constraint part involves observations at the antennas during training ($\boldsymbol{Y}_T$) and the known uplink training sequence sent by user k ($\boldsymbol{S}_{Tk,:}$). This constraint imposes the reconstruction requirement for the training symbols at the output of the equalizer. We remind that since the training size is selected to be less than the number of users, i.e., $\tau_T < K$, the constraint set involves not only the perfect equalizers but also some other matrices.
- It is the task of the cost function minimization to eliminate the undesired matrices in the constraint set which do not lead to sparse $\tilde{\boldsymbol{G}}_{k,:}$. In fact, as it is shown above, the peak equalizer output $\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty$ is a reflector of the (non)sparsity of $\tilde{\boldsymbol{G}}_{k,:}$.

In summary, the compressed training approach exploits the special rectangular QAM structure of the digital communication signals, to utilize the data symbols for learning, and therefore, reducing the required training length. In other words, the special constellation structure of sources is the side information used as an unsupervised resource to supplement training based adaptation.

C. Compressed Training Approach

In this section, we will extend the optimization setting introduced in the previous section. *Setting IIa* is used to obtain the k^{th} row of the equalizer matrix $\boldsymbol{W}$. We can actually combine individual optimization settings for different rows of $\boldsymbol{W}$ into a more compact single optimization setting:

$$\textit{Setting IIIa:} \quad \underset{\boldsymbol{W}}{\text{minimize}} \quad \frac{1}{\sqrt{\beta}-1}\sum_{k=1}^{K}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty \quad \text{s.t.} \quad \begin{matrix} \boldsymbol{W}\boldsymbol{Y}_T & = & \boldsymbol{S}_T \\ \boldsymbol{O} & = & \boldsymbol{W}\boldsymbol{Y}. \end{matrix}$$

Setting IIIa can be solved using Linear Programming (LP).

1) Training Length Selection: Although the proposed approach allows the reduction in the training length below the number of users K, we can not arbitrarily decrease it. It is interesting to investigate what is the minimum training length that we need to use. For this purpose, we first formalize the assumption that we used in the previous section about the packet of uplink transmit sequences:

Let $\mathcal{H}^\beta = \{\boldsymbol{q} : \|\Omega(\boldsymbol{q})\|_\infty \leq \sqrt{\beta}-1\}$ be the minimum volume hyper-rectangle covering the source vector samples, where each source takes its values from the β-QAM constellation $\mathcal{C}^\beta_{QAM}$ defined in (5). To proceed further, for a complex matrix $\boldsymbol{C}$, let $\tilde{\boldsymbol{C}}$ be the output of $\Upsilon(\cdot)$ which is defined as

$$\Upsilon(\boldsymbol{C}) = \begin{bmatrix} \Re e\{\boldsymbol{C}\} & -\mathcal{I}m\{\boldsymbol{C}\} \\ \mathcal{I}m\{\boldsymbol{C}\} & \Re e\{\boldsymbol{C}\} \end{bmatrix}. \tag{11}$$

We also define

$$V(\mathcal{H}^\beta) = \{(\sqrt{\beta}-1)\boldsymbol{v} \mid \boldsymbol{v} \in \{-1,1\}^{2K}\}, \tag{12}$$

as the set of all vertex points of $\mathcal{H}^\beta$. The following is the main assumption in establishing the link between peak magnitude equalizer of the k^{th} output and the sparsity of $\boldsymbol{G}_{k,:}$:

Assumption ($\mathbf{A}_*$): The set of columns of the transmit sequence matrix $\Upsilon(\boldsymbol{S})$ contains all vertex points (corners) of $\mathcal{H}^\beta$, i.e., $V(\mathcal{H}^\beta)$ is a subset of the set of columns of $\Upsilon(\boldsymbol{S})$.

The assumption ($\mathbf{A}_*$) makes sure that $\frac{1}{\sqrt{\beta}-1}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty = \|\tilde{\boldsymbol{G}}_{k,:}\|_1$. Therefore, under the assumption ($\mathbf{A}_*$), and based on the full rank condition on $\boldsymbol{H}$, *Setting IIa* is equivalent to *Setting Ia*. Observing

$$\boldsymbol{W}_{k,:}\boldsymbol{Y}_T = \boldsymbol{W}_{k,:}\boldsymbol{H}\boldsymbol{S}_T = \boldsymbol{G}_{k,:}\boldsymbol{S}_T,$$

we can rewrite *Setting Ia*, completely in terms of $\boldsymbol{G}_{k,:}$ as

$$\textit{Setting Ic:} \quad \underset{\boldsymbol{G}_{k,:}}{\text{minimize}} \quad \|\tilde{\boldsymbol{G}}_{k,:}\|_1 \quad \text{s.t.} \quad \boldsymbol{G}_{k,:}\boldsymbol{S}_T = \boldsymbol{S}_{Tk,:}.$$

Similarly, *Setting IIIa* can be rewritten in terms of $\boldsymbol{G}$ as

$$\textit{Setting IIIc:} \quad \underset{\boldsymbol{G}}{\text{minimize}} \quad \sum_{k=1}^{K}\|\tilde{\boldsymbol{G}}_{k,:}\|_1 \quad \text{s.t.} \quad \boldsymbol{G}\boldsymbol{S}_T = \boldsymbol{S}_T.$$

Setting IIIc above covers all the rows of $\boldsymbol{G}$ and it can be decomposable into K optimizations in *Setting Ic* for individual rows.

We note that *Setting Ic* is in the form of the Sparse Reconstruction Problem in Compressed Sensing [22]. Therefore, we can adapt analysis approaches developed in the compressed sensing literature to make an assessment about the minimum training length. We first assume that the training sequences use the constellation $\mathcal{C} = \{c_r + ic_I : c_r, c_I \in \{-(\sqrt{\beta}-1), \sqrt{\beta}-1\}\}$, i.e., the training source vectors, which are rows of $\boldsymbol{S}_T$, are selected from the corners of $\mathcal{H}^\beta$. This is a usual practice for the training and also simplifies the notation in our analysis. Under these assumptions, the following corollary provides a recipe for the training length selection:

Corollary I:. Given $\boldsymbol{S}_T \in \mathbb{C}^{K\times\tau_T}$ is a matrix with i.i.d. elements chosen from the set $\mathcal{C}$. If $\tau_T > log_4(K(K-1))+0.5$, then the solution of the *Setting IIIc* is unique and equals to $\boldsymbol{I}$ with probability at least

$$1 - \frac{K(K-1)}{2\cdot 4^{\tau_T-1}}. \tag{13}$$

Proof: See Appendix I.

Therefore, the corollary above suggests that the training length should be chosen as

$$\boxed{\tau_T > log_2(K) + 0.5}.$$

Furthermore, the probability lower bound for perfect equalization in (13) has a phase transition around this lower bound for the training length. We illustrate this phenomenon through an example in the numerical examples section (Section V).

IV. Practical Considerations And Extensions

In this section, we address some practical aspects of the compressed training approach introduced in the previous section. The impact of packet length on the algorithm's performance is discussed in Section IV-A. Section IV-B addresses the algorithm modifications to mitigate the effects of receiver noise. The decision directed extension of the algorithm for performance improvement is introduced in Section IV-C. Finally, Section IV-D presents an enhancement targeting algorithm acceleration.

A. Packet Length Considerations

One of the concerns about the compressed training approach is the potential impact of the packet length on the performance of the algorithm. The packet length plays a role in establishing **Assumption ($\mathbf{A}_*$)**, and therefore, the equivalence on the peak value of the equalizer output, i.e., $\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty$, and the non-sparseness measure of the corresponding row of the overall equalized channel, i.e., $\|\tilde{\boldsymbol{G}}_{k,:}\|_1$. As we noted earlier in Section III-B, the equivalence is achieved if the transmit packets contain all vertex points of the form given by (10). The probability of inclusion of these vertex points decreases with decreasing packet length. Despite this valid concern, the following theorem asserts a phase transition property for the choice of packet lengths, which implies that perfect equalizers can be obtained through *Setting IIIa* with overwhelming probability even for practical packet lengths significantly shorter than that is required to satisfy **Assumption ($\mathbf{A}_*$)**.

Theorem II: Let the solution of *Setting IIIc* is unique and *Corollary I* holds with probability almost one. Then, the probability of the equivalence of *Setting IIIa* and *Setting IIIc* is upper bounded by

$$4(1-\mathbb{P}_\chi(\Gamma, d_{N_L})) \tag{14}$$

and lower bounded by

$$1-K2^{\upsilon}\left(1-\frac{1}{\sqrt{\beta}}\frac{2^{\mathcal{X}}-1}{2^{\mathcal{X}}}\right)^{\Gamma} \tag{15}$$

where $d_{N_L}=2(K-\tau_T)$, $\chi=1-\log_2(\frac{m_o-1}{m_o-d_{N_L}})$ and $\upsilon=-(d_{N_L}-1)\log_2(d_{N_L}-1)-\log_2(m_o-1)+d_{N_L}\log_2(m_o-d_{N_L})+1$, $m_o\in(2d_{N_L},\Gamma)$ and,

$$\mathbb{P}_\chi(\Gamma, d_{N_L})=\frac{1}{2^{\Gamma-1}}\sum_{j=0}^{d_{N_L}-1}\binom{\Gamma-1}{j}. \tag{16}$$

Proof: Proof is in Appendix II.

These bounds explicitly show that there exists a phase transition for packet length concerning the equivalence of *Setting IIIa* and *Setting IIIc*. Defining $q=m_o/d_{N_L}$, we can write

$$\begin{aligned}\left(1-\frac{1}{\sqrt{\beta}}\frac{2^{\mathcal{X}}-1}{2^{\mathcal{X}}}\right) &= \left(1-\frac{1}{2\sqrt{\beta}}\frac{q-2}{q-1}\right)\\ \upsilon &\approx d_{N_L}\log_2(q-1).\end{aligned}$$

Therefore, based on (15), the phase transition value for the packet length can be written approximately as

$$\begin{aligned}\Gamma_{ph} &\approx \frac{-(\upsilon+\log_2(K))}{\log_2\left(1-\frac{1}{\sqrt{\beta}}\frac{2^{\mathcal{X}}-1}{2^{\mathcal{X}}}\right)}\\ &\approx \frac{-(d_{N_L}\log_2(q-1)+\log_2(K))}{\log_2(1-\frac{1}{2\sqrt{\beta}}\frac{q-2}{q-1})}\\ &\approx \frac{d_{N_L}\log_2(q-1)+\log_2(K)}{\frac{1}{2\sqrt{\beta}\log(2)}\frac{q-2}{q-1}}\\ &= 2\sqrt{\beta}(d_{NL}\log(q-1)+\log(K))\frac{q-1}{q-2}\end{aligned} \tag{17}$$

where (17) is due to the approximation $\log_2(1-x)\approx(1-x)/\log(2)$. For the specific choice of $q=2.6$, we can write

$$\boxed{\Gamma_{ph}\approx 5\sqrt{\beta}((K-\tau_T)+log(K)).} \tag{18}$$

As a result, the minimum packet length is linearly proportional to the sum of the gap between the number of users and the training length and the logarithm of the number of user terminals, as well as the square root of the QAM-constellation size. We should note that this prescription is based on the lower bound, and therefore, it provides an overestimate as illustrated by the examples in Section V.

We should note that Assumption $\mathbf{A}_*$ requires that the packet length is at least 2^{2K}. However, Theorem II provides a significant relief on this requirement, and shows that the compressed training framework can work with realistic packet lengths. Main enabler of this practical result is the fact that the probability of intersection of the random subspace with the positive orthant decays exponentially with packet length, as used in the proof of Theorem II.

B. Noise Considerations

So far, we assumed that there is no noise in the observations. Of course, this assumption is too optimistic for real applications. In this subsection, we take noise into consideration and develop corresponding compressed training algorithms.

We start with the following observations:

- In case $\boldsymbol{Y}_T$ contains noise, the goal of reconstructing training in the form $\boldsymbol{W}\boldsymbol{Y}_T=\boldsymbol{S}_T$ as in the constraint part of the optimization *Setting IIIa* is very ambitious/unrealistic.
- The presence of noise in $\boldsymbol{Y}$ implies that the equalizer output $\boldsymbol{O}$ is also noisy. The noise effects should be taken into consideration in determining the absolute peak estimates corresponding to the noiseless version of the equalizer outputs.

We first address the first issue, the noise effect on the constraint, and then extend our treatment to cover the second issue, the noise effect on the cost function.

1) The Noise in the Constraint: In the presence of noise, instead of trying to perfectly reconstruct the training symbols, we can aim to keep the equalizer output at the close vicinity of them. Assuming that the noise samples are Gaussian,

natural distance metric would be the Frobenius norm. We can, therefore, reflect this modification on *Setting IIIa* as follows:

Setting IVa:

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & \frac{1}{\sqrt{\beta}-1}\sum_{k=1}^{K}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty \\ \text{subject to} \quad & \|\boldsymbol{W}\boldsymbol{Y}_T-\boldsymbol{S}_T\|_F \leq \delta \\ & \boldsymbol{W}\boldsymbol{Y} = \boldsymbol{O}, \end{aligned}$$

where δ is the algorithm parameter reflecting the strength of the noise. As an alternative, this setting can be replaced with the "Lagrangian" form where the constraint is appended to the cost function:

Setting IVb:

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & \textstyle\sum_{k=1}^{K}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty + \lambda\|\boldsymbol{W}\boldsymbol{Y}_T-\boldsymbol{S}_T\|_F \\ \text{subject to} \quad & \boldsymbol{W}\boldsymbol{Y}=\boldsymbol{O} \end{aligned}$$

where we dropped $\frac{1}{\sqrt{\beta}-1}$ factor for notational convenience, as the regularization parameter λ can be used to adjust the relative weights of the components.

Iterative Algorithm: We can obtain the solution of *Setting IVb* through an iterative algorithm. We note that due to the first term in the cost function, this is a non-smooth convex optimization. There are various alternative approaches that can be followed, especially if we take recent advances in the field of low complexity and efficient algorithm development for non-smooth convex optimization problems into consideration [19]. However, we conceive subgradient iterations as a simple solution.

We first start with partitioning the cost function in *Setting IVb* into two components:

$$J_1(\boldsymbol{W})=\sum_{k=1}^{K}\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty, \qquad J_2(\boldsymbol{W})=\lambda\|\boldsymbol{W}\boldsymbol{Y}_H-\boldsymbol{S}_T\|_F.$$

Since $J_2(\boldsymbol{W})$ is a differentiable function, we can write the corresponding gradient as

$$\nabla_{\boldsymbol{W}}J_2(\boldsymbol{W})=\frac{\lambda\boldsymbol{Y}^H(\boldsymbol{W}\boldsymbol{Y}_T-\boldsymbol{S}_T)}{\|\boldsymbol{W}\boldsymbol{Y}_T-\boldsymbol{S}_T\|_F}.$$

We can decompose the non-smooth function $J_1(\boldsymbol{W})$ as the sum of K non-smooth functions, each corresponding to a different user,

$$J_1(\boldsymbol{W})=\sum_{k=1}^{K}J_{1,k}(\boldsymbol{W})$$

where $J_{1,k}(\boldsymbol{W})=\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty$ for $k=1,\ldots,K$. The subdifferential set corresponding to $J_{1,k}$ can be written as

$$\partial J_{1,k}(\boldsymbol{W})=\boldsymbol{Co}\{\boldsymbol{e}_k\Omega^{-1}(\text{sign}(\tilde{o}_k(l_k))\tilde{\boldsymbol{y}}(l_k)^T), l_k\in\mathcal{L}_k\},$$

where $\Omega^{-1}(\bullet)$ is the inverse of the isomorphic mapping and

$$\mathcal{L}_k=\{l:|\tilde{o}_k(l)|=\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty\}$$

is the set of indices at which the absolute peak is achieved for the k^{th} equalizer output. As a result, the subgradient based iterative algorithm can be written as

$$\boldsymbol{W}^{(t+1)}=\boldsymbol{W}^{(t)}+\mu^{(t)}\boldsymbol{U}^{(t)},$$

where $\mu^{(t)}$ is the step size at the t^{th} iteration, and the update matrix is given by

$$\begin{aligned} \boldsymbol{U}^{(t)} &= \sum_{k=1}^{K}\boldsymbol{e}_k{\boldsymbol{\xi}_k^{(t)}}^T\text{diag}(\text{sign}(\tilde{\boldsymbol{O}}_{k,:}^{(t)}))\begin{bmatrix}\boldsymbol{Y}^H\\ j\boldsymbol{Y}^H\end{bmatrix} \\ &\quad +\frac{\lambda\boldsymbol{Y}^H(\boldsymbol{W}^{(t)}\boldsymbol{Y}_T-\boldsymbol{S}_T)}{\|\boldsymbol{W}^{(t)}\boldsymbol{Y}_T-\boldsymbol{S}_T\|_F}. \end{aligned} \tag{19}$$

where we define $\boldsymbol{\xi}_k^{(t)}=\frac{\boldsymbol{\zeta}_k^{(t)}}{\|\boldsymbol{\zeta}_k^{(t)}\|_1}$ and

$$\boldsymbol{\zeta}_{k\ l,1}^{(t)}=\begin{cases}1 & |\tilde{\boldsymbol{O}}_{k,l}^{(t)}|\geq\alpha\|\tilde{\boldsymbol{O}}_{k,:}^{(t)}\|_\infty\\ 0 & \text{otherwise.}\end{cases}$$

Here, $\boldsymbol{\xi}_{k,l_k^{(t)}}$ are the convex combination coefficients which are non-negative and satisfy $\sum_{l_k^{(t)}\in\mathcal{L}_k^{(t)}}\boldsymbol{\xi}_{k,l_k^{(t)}}=1$.

2) Noise in the Cost Function: In the previous discussion, we just paid attention to the noise in the constraint part, and ignored the impact of noise in the objective part. In this section, we will improve the algorithm to mitigate the impact caused by the presence of noise at the equalizer output.

We start with writing the expression for the separator output in a form to reflect the noise component

$$\boldsymbol{O} = \boldsymbol{W}\boldsymbol{Y} = \boldsymbol{W}(\boldsymbol{H}\boldsymbol{S}+\boldsymbol{V}) = \underbrace{\boldsymbol{W}\boldsymbol{H}\boldsymbol{S}}_{\boldsymbol{Q}}+\underbrace{\boldsymbol{W}\boldsymbol{V}}_{\boldsymbol{\Delta}}.$$

The goal of the cost function is to enforce sparsification of $\boldsymbol{G}=\boldsymbol{W}\boldsymbol{H}$, and the non-sparsity of $\boldsymbol{G}$ is reflected in the absolute peak values for the rows of $\boldsymbol{Q}$. Due to the impact of noise term $\boldsymbol{\Delta}$, a non-absolute-peak of $\tilde{\boldsymbol{Q}}_{k,:}$ can appear as a peak location in $\tilde{\boldsymbol{O}}_{k,:}$ and vice versa where k corresponds any column.

Our actual objective is to determine the absolute peak locations in $\tilde{\boldsymbol{Q}}_{k,:}$, rather than the non-absolute-peak values, as that is what matters in the algorithm iterations in (19). Since the noise is random, we can only make a probabilistic inference about whether a given location is actually an absolute peak or not. Therefore, the following discussion introduces an approach *to estimate the true subgradient* corresponding to the cost function component based on the available packet data.

Let $\{q_k(n)|n\in\{1,\cdots,2\Gamma\}\}$ represent the noiseless equalizer output sequence for the k^{th} user, which corresponds to $\tilde{\boldsymbol{Q}}_{k,:}$. We can write

$$\tilde{o}_k(n)=q_k(n)+\Delta_k(n), \tag{20}$$

where $\Delta_k(n)=\tilde{\boldsymbol{\Delta}}_{k,n}$ is the noise component. Our goal is to find the probability mass function, $\{p_k(n);n\in\{1,\cdots,2\Gamma\}\}$ where $p_k(n)$ represents the probability of index n is an absolute peak location for the k^{th} user's noiseless output sequence $\{q_k\}$. More explicitly, we can write

$$p_k(n)=\mathbb{P}(|q_k(n)|=\|\tilde{\boldsymbol{Q}}_{k,:}\|_\infty).$$

If the index m is known to be the peak location for the sequence $|q_k|$, then the corresponding subgradient would be

$$\boldsymbol{e}_k\Omega^{-1}(\text{sign}(\tilde{o}_k(m))\tilde{\boldsymbol{y}}(m)^T),$$

where we assumed that $\text{sign}(q_k(n))=\text{sign}(\tilde{o}_k(m))$ for the potential absolute peak locations. Since each index l has an

assigned probability of $p_k(l)$, we can calculate the expected subgradient as the probability weighted sum of the corresponding subgradients:

$$\sum_{l=1}^{2\Gamma} p_k(l)\boldsymbol{e}_k\Omega^{-1}(\text{sign}(\tilde{o}_k(l))\tilde{\boldsymbol{y}}(l)^T).$$

This very much resembles the form of the subgradient based term in (19) for the noiseless case. However, in that case the summation is over the set of indices $\mathcal{L}_k$ for which the same peak value is achieved. In the noisy scenario, this summation is extended to the whole index set $\{1,\cdots,2\Gamma\}$. The convex combination coefficients for contributions of different index sets are replaced by $p_k(n)$. So, each index point contributes to the expected subgradient proportional to its probability of containing the absolute peak. As a result, for the noisy case, we can write the algorithm update term in iteration t as,

$$\begin{aligned}\boldsymbol{U}^{(t)} &= \sum_{k=1}^{K}\sum_{l_k=1}^{2\Gamma} p_k^{(t)}(l_k)\boldsymbol{e}_k\Omega^{-1}(\text{sign}(\tilde{o}_k^{(t)}(l_k))\tilde{\boldsymbol{y}}(l_k)^T) \\ &\quad + \frac{\lambda \boldsymbol{Y}^H(\boldsymbol{W}^{(t)}\boldsymbol{Y}_T - \boldsymbol{S}_T)}{\|\boldsymbol{W}^{(t)}\boldsymbol{Y}_T - \boldsymbol{S}_T\|_F}. \end{aligned} \quad (21)$$

The main question to ask at this point is how to obtain the pmf $p_k^{(t)}(n)$ for each iteration and for all user sequences, using the observations $\{\tilde{o}_k^{(t)}(n), n\in\mathcal{I}\}$ satisfying (20) and assumed noise variance σ_{Δ_k}. Naturally, the ordering of the probabilities $p_k^{(t)}(n)$ should follow the ordering of the absolute noisy output values $\{\tilde{o}_k^{(t)}(n)\}$. However, it turns out the exact calculation of the pmf p_k is really a cumbersome process. Especially for the Gaussian noise assumption, it is almost impossible to obtain a closed form formula for these probabilities. In order to ease this task, we introduce the following low-complexity approach to come up with a reasonable p_k: at iteration t, we determine the indices for which the absolute output values are at the vicinity of the absolute output peak value, i.e.,

$$\mathcal{I}_k^{(t)} = \{l : |\tilde{o}_k^{(t)}(l)| \geq \alpha\|\tilde{\boldsymbol{O}}_{k,:}^{(t)}\|_\infty, l\in\{1,\cdots,2\Gamma\}\},$$

where $\alpha\leq 1$ is an algorithm parameter to be adjusted. Then the proposed pmf can be written as

$$p_k^{(t)}(n) = \begin{cases} \frac{1}{|\mathcal{I}_k^{(t)}|} & n\in\mathcal{I}_k^{(t)}, \\ 0 & \text{otherwise.}\end{cases}$$

This pmf assigns uniform probability to all indices determined by $\mathcal{I}_k^{(t)}$. The update rule based on this pmf can be written as

$$\begin{aligned}\boldsymbol{U}^{(t)} &= \sum_{k=1}^{K}\sum_{l_k\in\mathcal{I}_k^{(t)}} \frac{1}{|\mathcal{I}_k^{(t)}|}\boldsymbol{e}_k\Omega^{-1}(\text{sign}(\tilde{o}_k^{(t)}(l_k))\tilde{\boldsymbol{y}}(l_k)^T) \\ &\quad + \frac{\lambda \boldsymbol{Y}^H(\boldsymbol{W}^{(t)}\boldsymbol{Y}_T - \boldsymbol{S}_T)}{\|\boldsymbol{W}^{(t)}\boldsymbol{Y}_T - \boldsymbol{S}_T\|_F}. \end{aligned} \quad (22)$$

We'll refer to the corresponding algorithm as the **MIMO Co**mpressed **T**raining **A**lgorithm with acronym MIMO-CoTA.

Investigating the convergence properties of the proposed algorithm is required to understand its weaknesses and strengths. In order to illustrate the convergence behaviour of the algorithm, we consider the following numerical experiment: For a setting with $M=500$ base station antennas and $K=20$ users (with $\boldsymbol{H}$ is selected as zero mean i.i.d. Gaussian entries), we compare the the ideal vertex point $\boldsymbol{\nu}_k^{(t)}\in V(\mathcal{H}^\beta)$ with its estimate obtained by projecting k^{th} row of subgradient of $\Upsilon(\boldsymbol{U})^{(t)}$ in (22) which is equal to

$$\hat{\boldsymbol{\psi}}_k^{(t)^T} = \sum_{l_k\in\mathcal{I}_k^{(t)}} \frac{1}{|\mathcal{I}_k^{(t)}|}\text{sign}(\tilde{o}_k^{(t)}(l_k))\tilde{\boldsymbol{y}}(l_k)^T.$$

through

$$\hat{\boldsymbol{\nu}}_k^{(t)} = \boldsymbol{H}^\dagger\hat{\boldsymbol{\psi}}_k^{(t)},$$

where $\boldsymbol{H}^\dagger$ is the Moore-Penrose pseudo-inverse of the channel matrix $\Upsilon(\boldsymbol{H})$.

The comparison is performed through computing the angle between the vertex point and its estimate as

$$\theta = \text{acos}\left(\frac{\langle\boldsymbol{\nu}_k^{(t)},\hat{\boldsymbol{\nu}}_k^{(t)}\rangle_{\boldsymbol{\Pi}_k^{(t)}}}{\|\boldsymbol{\nu}_k^{(t)}\|_{\boldsymbol{\Pi}_k^{(t)}}\|\hat{\boldsymbol{\nu}}_k^{(t)}\|_{\boldsymbol{\Pi}_k^{(t)}}}\right),$$

where $\langle\boldsymbol{a},\boldsymbol{b}\rangle_{\boldsymbol{\Pi}_k^{(t)}} \triangleq \boldsymbol{b}^T\boldsymbol{\Pi}_k^{(t)}\boldsymbol{a}$ is a weighted inner product and $\|\boldsymbol{a}\|_{\boldsymbol{\Pi}_k^{(t)}} \triangleq \sqrt{\langle\boldsymbol{a},\boldsymbol{a}\rangle_{\boldsymbol{\Pi}_k^{(t)}}}$ is the corresponding induced norm. Here the weighting matrix $\boldsymbol{\Pi^{(t)}}$ is selected as the diagonal matrix whose i^{th} diagonal entry is chosen to be equal to $|\Upsilon(\boldsymbol{G})_{k,i}^{(t)}|^2$. The reason for this weighting can be explained as follows: the impact of the source vector component in determining peak value is proportional to the magnitude of the corresponding element of $\Upsilon(\boldsymbol{G})_{k,:}^{(t)}$. Therefore, in comparing the vertex $\boldsymbol{\nu}_k^{(t)}$ and its estimate $\hat{\boldsymbol{\nu}}_k^{(t)}$, we use an inner product which weighs components proportional to its impact on determining the output value. Fig. 3 shows the angle between the source vertex and its sample based estimate as a function of iterations, for different Signal-to-Noise-Power-Ratio (SNR) values and packet lengths ($\boldsymbol{\Gamma}$). These plots were obtained for 8000 independent channel, transmit data and noise realizations, and the solid line represents the mean behaviour while the shaded area corresponds to 95 percent confidence region. It can be confirmed by these results that the vertex based true subgradient and its estimate forms an acute angle whose value decreases significantly towards zero as iterations increase. Furthermore, increasing SNR and/or packet size significantly decreases the variance.

As illustrated by the numerical experiments in Section V, the algorithm based on this update has very reasonable performance.

3) A Complete Noise Consideration: The algorithms given in the previous section assume a noise free output in the optimization setting, and then cast the impact of noise by modifying the algorithm updates based on the estimation of the subgradient in the noise-free output case. In this subsection, the optimization formulation is modified to address the impact of noise in the equalizer outputs. For this purpose, we convert the constraint on the equality of the noise free outputs $\boldsymbol{O}$ and $\boldsymbol{WY}$ to an inequality. In this respect, assuming we have a good estimate of noise variance σ^2, we can propose the

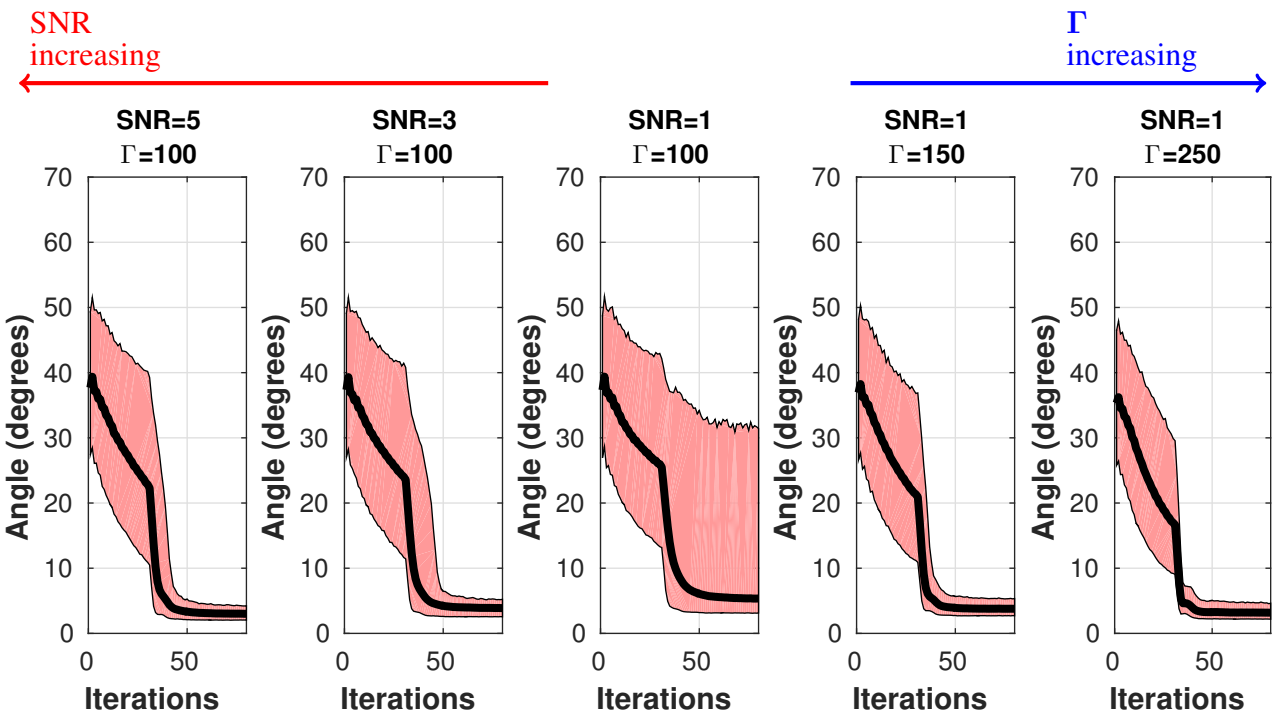


Fig. 3: The effective angle between the vertex $\boldsymbol{v}_k^{(t)}$ and its estimate obtained by reflecting the corresponding subgradient component into source domain, for different SNR's and packet lengths ($\boldsymbol{\Gamma}$). Mean behavior is the solid line, and 95 confidence region is the shaded area.

following setting:

$$\begin{aligned} &\textit{Setting V:}\\ &\underset{\boldsymbol{W},\boldsymbol{O}}{\text{minimize}} \quad \sum_{k=1}^{K} \frac{\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty^2}{(\sqrt{\beta}-1)^2} + \sigma^2\|\boldsymbol{W}_{k,:}\|_2^2\\ &\text{s.t.} \quad \|\boldsymbol{W}_{k,:}\boldsymbol{Y}_T - \boldsymbol{S}_{T_{k,:}}\|_2 \le \sigma\sqrt{L_T}\|\boldsymbol{W}_{k,:}\|_2\\ &\qquad \|\boldsymbol{W}_{k,:}\boldsymbol{Y} - \boldsymbol{O}_{k,:}\|_2 \le \sigma\sqrt{L_D}\|\boldsymbol{W}_{k,:}\|_2\\ &\qquad \text{for } \forall k \in \{1,\cdots,K\}, \end{aligned}$$

and the Lagrangian of the *Setting V* can be written as

$$\begin{aligned} &\textit{Setting VI:}\\ &\underset{\boldsymbol{W},\boldsymbol{O}}{\text{minimize}} \sum_{k=1}^{K}\Bigg[\frac{\|\tilde{\boldsymbol{O}}_{k,:}\|_\infty^2}{(\sqrt{\beta}-1)^2} + \sigma^2\|\boldsymbol{W}_{k,:}\|_2^2\\ &+\lambda_1\left(\|\boldsymbol{W}_{k,:}\boldsymbol{Y}_T - \boldsymbol{S}_{T_{k,:}}\|_2 - \sigma\sqrt{L_T}\|\boldsymbol{W}_{k,:}\|_2\right)\\ &+\lambda_2\left(\|\boldsymbol{W}_{k,:}\boldsymbol{Y} - \boldsymbol{O}_{k,:}\|_2 - \sigma\sqrt{L_D}\|\boldsymbol{W}_{k,:}\|_2\right)\Bigg]. \end{aligned}$$

Following the same procedure as in Section IV-B, we can also propose an iterative algorithm to solve this optimization problem. However, this time we have two parameters to update. Exploiting the convexity of the problem, we can force the algorithm to iterate over one variable and updating the other one with some period. Hence, the update based on $\boldsymbol{W}$ can be written as

$$\boldsymbol{W}^{(t+1)} = (\lambda_1\boldsymbol{Y}_T\boldsymbol{Y}_T^H + \lambda_2\boldsymbol{Y}\boldsymbol{Y}^H + \tilde{\lambda}\boldsymbol{I})^{-1}(\lambda_1\boldsymbol{S}_T\boldsymbol{Y}_T^H + \lambda_2\boldsymbol{O}^{(t)}\boldsymbol{Y}^H), \tag{23}$$

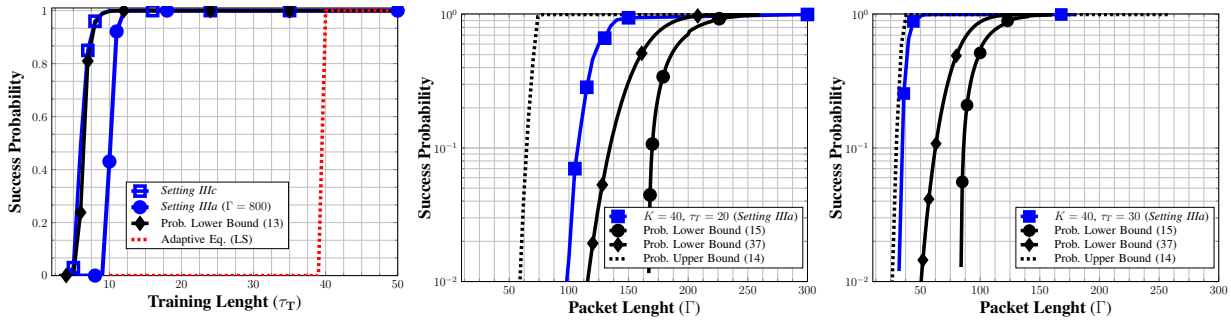
and $\boldsymbol{O}$ is iteratively updated as $\boldsymbol{O}^{t+1} = \boldsymbol{O}^t + \mu^{(t)}\boldsymbol{U}^{(t)}$ where

$$\begin{aligned} \boldsymbol{U}^{(t)} = \|\tilde{\boldsymbol{O}}_{k,:}\|_\infty \sum_{k=1}^{K}\sum_{l_k\in\mathcal{I}_k^{(t)}} \frac{1}{|\mathcal{I}_k^{(t)}|}\boldsymbol{e}_k\Omega^{-1}(\text{sign}(\tilde{o}_k(l_k))\boldsymbol{e}_{l_k}^T)\\ + \sum_{k=1}^{K}\frac{\lambda_2\boldsymbol{e}_k(\boldsymbol{W}_{k,:}^{(t)}\boldsymbol{Y} - \boldsymbol{S}_{k,:})\boldsymbol{Y}^H}{\|\boldsymbol{W}_{k,:}^{(t)}\boldsymbol{Y} - \boldsymbol{S}_{k,:}\|_2}. \end{aligned} \tag{24}$$

Note here, $\tilde{\lambda}$ is the coefficient that includes both Lagrangian coefficients of the constraints and the objective function related to the parameter $\boldsymbol{W}_{k,:}$, which can be written as

$$\tilde{\lambda} = 2\sigma^2 - \frac{\lambda_1\sigma\sqrt{L_T}}{2\|\boldsymbol{W}^{(t)}\|_F} - \frac{\lambda_2\sigma\sqrt{L_D}}{2\|\boldsymbol{W}^{(t)}\|_F}. \tag{25}$$

We refer to the corresponding algorithm as MIMO-CoTA-2. We need to note that the objective of *Setting V* aims to mimic the MMSE solution of the problem by considering the noise both in the constraint and the cost function. For that, it utilizes the knowledge of noise power. Moreover, the number of unknown is also much higher than *Setting IVa-b* which increases the number of iteration for the corresponding iterative algorithm to converge. Therefore, we still exploit *Setting V* as a sanity check algorithm to demonstrate that even with no knowledge about the power of the noise, we can still obtain satisfying results with the iterative algorithm corresponding *Setting IVa-b*.



(a) (b) (c)

Fig. 4: a) Probability of successful training for noiseless compressed training setting *Setting IIIa* is compared against: the idealized case *Setting IIIc*, Adaptive Least Squares Approach and The Probability Bound in (13). b-c) The equivalence probability of *Setting IIIa* and *Setting IIIc* for varying packet length.

C. Decision Directed Compressed Training Approach

The proposed approach makes use of the special boundedness feature of PAM and QAM constellations. It is actually desirable to exploit more detailed knowledge of these constellations for better training. In fact, the decision directed approaches try to exploit more detailed information about the location of the constellation points. However, there are two fundamental issues typically attributed to such approaches:

- The decision directed algorithms need reliable decisions to start with so that they can converge to a useful point. This is usually not a reasonable assumption in real applications, therefore, another "acquisition" algorithm is needed to open the initially closed eye.
- Another major difficulty is caused by the fact that decision directed approaches lead to non-convex objectives, in general, that have the issues about slow or mis-convergence.

In this section, we introduce an algorithm modification to incorporate decision directed approach to the proposed compressed training framework. The main idea is to extend the training region used in the algorithms with the reliable decisions. Therefore, we can consider this new approach as an algorithm consisting of two main phases:

Ph.I: We employ the compressed training algorithm introduced in the previous section for some initial period. After the completion of this interval, we can perform reliable decisions on some symbols,

Ph.II: We continue compressed training algorithm iterations with continuously extending training region: at the beginning of each iteration, the training region is appended with the reliable decisions of the previous iteration. Therefore, as the iterations progress, the training region will expand to an extent to cover the whole packet.

D. Nesterov's Acceleration Approach

The speed of convergence for the adaptive algorithms presented in the previous sections are critical for real time implementations. In order to increase the convergence speed of the proposed algorithm, we can adapt Nesterov's fast acceleration approach [23]. This approach is actually proposed for smooth and strongly convex functions and its interpretation/justification is still an area of research (see for example [24]). Despite the fact that our setting is piecewise smooth, adaptation of this scheme to our update term provides a significant improvement over the standard subgradient search.

The basic modification in the algorithm can be described as follows: let $\boldsymbol{U}^{(t)}$ be the update term computed for the standard form of the compressed training approach as outlined in Section IV-C. We replace the algorithm iteration in the form

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} + \mu^{(t)}\boldsymbol{U}^{(t)}$$

with a two step procedure by introducing an intermediate variable $\boldsymbol{X}^{(t)}$:

$$\begin{aligned} \boldsymbol{X}^{(t+1)} &= \boldsymbol{W}^{(t)} - \upsilon^{(t)}\boldsymbol{U}^{(t)} \\ \boldsymbol{W}^{(t+1)} &= \boldsymbol{X}^{(t+1)} + \kappa^{(t)}(\boldsymbol{X}^{(t+1)} - \boldsymbol{X}^{(t)}), \end{aligned}$$

where $\upsilon^{(t)}$ and $\kappa^{(t)}$ are algorithm parameters. In our implementations, we fixed $\upsilon^{(t)} = \frac{0.0003}{\sqrt{\log(t+1)}}$ and $\kappa^{(t)} = \frac{t-1}{t+2}$.

V. Numerical Experiments

A. Noiseless Case Example

The purpose of the first numerical experiment is to illustrate the phase transition phenomenon for the probability of perfect equalization as a function of training length τ_T and packet length Γ selections as predicted by the analytical results in Section III-C.1 and Section IV-A. For this purpose, a multiuser Massive MIMO setup with

- $K = 40$ users and $M = 150$ base station antennas,
- 4-QAM constellation,
- Channel matrix with i.i.d. Gaussian coefficients,

is assumed. In these experiments,

- We use *Setting IIIc*, which is in terms of overall mapping $\boldsymbol{G}$ to evaluate the empirical probability of success as a function of selected training length τ_T. Note that this corresponds to the case that assumption $(\mathbf{A}_*)$ is perfectly valid, and therefore, forms a benchmark to evaluate the effect of finite packet length. We refer to the solution of the optimization setting $(\boldsymbol{G}_*)$ as successful if $\|\boldsymbol{G}_* - \boldsymbol{I}_K\|_F \leq 10^{-5}$(similar to [20]).

- We also evaluate the empirical probability of success for *Setting IIIa*, which is in terms of equalizer parameters $\boldsymbol{W}$, where the packet length is selected as $\Gamma = 800$ symbols.

Fig. 4a shows the aforementioned empirical success probabilities relative to the probability bound in (13) and success probability of the conventional adaptive least squares (LS) based equalization approach. The least squares based direct adaptation approach clearly requires at least $K = 40$ samples. For this experiment scenario, the compressed training length lower bound $\log_2(K)+0.5$ is equal to 5.82. In fact, we can see a phase transition for three of the curves related to compressed sensing starting with this training length value. The probability lower bound appears to be very accurate in the sense that it closely follows the success probability for *Setting IIIc*. The performance of *Setting IIIa* is slightly degraded relative to *Setting IIIc*, which is caused by the fact that a finite-length package of size $\Gamma = 800$ is being used in *Setting IIIa*, and it is likely that assumption $(\mathbf{A}_*)$ is violated in some realizations.

Moreover, in Fig. 4b, we plot the probability bounds for the equivalence of *Setting IIIa* and *Setting IIIc* along with the empirical probability of success for varying packet lengths. For the experiments, we use $K = 40$ and $\tau_T = 20$, which corresponds to 20 discrepancy between the number of user terminals and the number of training symbols. Based on these plots, we can claim that the derived bounds capture the evidenced phase transitions with some reasonable accuracy. The same experiments are repeated for $K = 40$ and $\tau_T = 30$ in Fig. 4c, which corresponds to half the discrepancy compared to the previous case, and therefore, the phase transition also occurs earlier as predicted by the bound derivations.

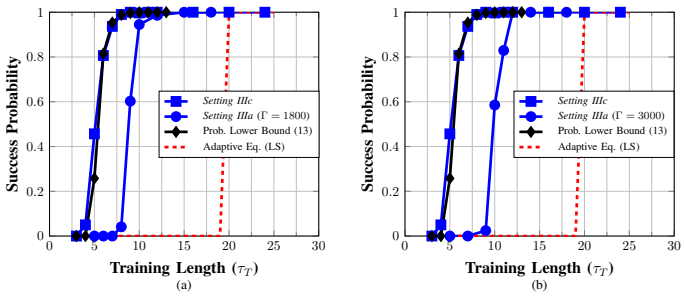


Fig. 5: Probability of successful training for noiseless compressed training setting *Setting IIIa* is compared against: the idealized case *Setting IIIc*, Adaptive Least Squares Approach and The Probability Bound in (13) for a) 64-QAM constellation, b) 256-QAM constellation.

In order to illustrate the success probability bounds for higher order constellations, we performed multiple experiments for 64-QAM and 256-QAM by varying the training size. We choose $M = 100$, $K = 20$, and $\Gamma = 1800$ and $\Gamma = 3000$ for 64-QAM and 256-QAM respectively. The results provided in Fig. 5 illustrates that, similar to Fig. 4a, there exists a phase transition in perfect equalization probability around $\log 2(K) + 0.5 \approx 5$. This confirms the expectations for the higher constellation case.

More critical issue is the impact of the constellation size on the required packet length. For this purpose, we devised an

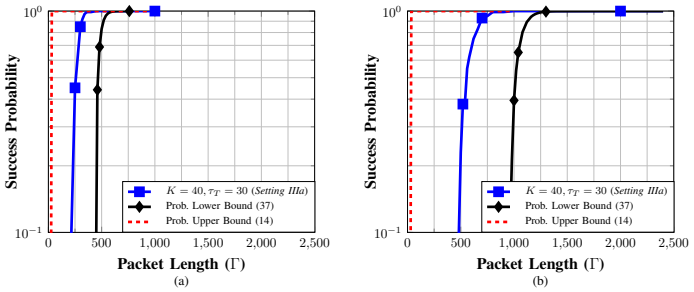


Fig. 6: The equivalence probability of *Setting IIIa* and *Setting IIIc* for varying packet length while the constellation schemes used in the simulations are a) 64-QAM, b) 256-QAM.

experiment for both 64-QAM and 256-QAM constellations, with parameters $M = 100$, $K = 40$ and $\tau_T = 30$. Fig. 6 provides empirical results as well as bounds for the perfect equalization probability as a function of packet length Γ for 64-QAM and 256-QAM constellations. As we expected, as the higher constellation schemes are considered, the gap between the lower and upper bounds also increases. Based on the theoretical prediction in (18), the required packet length is expected to be proportional to the square root of the constellation size. In fact, when the constellation size changes from 64 to 256, which is a 4 fold increase, the required packet length only doubles confirming the theoretical predictions. This is indeed a significant relief considering the exponential increase requirement implied by the worst case condition of including all corner points.

B. Noisy Case Example I

In this example, we consider a Massive MIMO setting with $K = 40$ users. Number of antennas is increased to $M = 1000$. The complex 4-QAM constellation is used (for both data and training sequences). To see the effect of the packet length on the performance of the proposed algorithm, we construct an experiment with the uplink transmission packet lengths $\Gamma = 100$ and $\Gamma = 300$. The channel matrix $\boldsymbol{H}$ is generated through the realizations of i.i.d. complex Gaussian random variables. We also consider a receiver noise with power corresponding to varying SNR levels.

In this experiment, we calculate the empirical (uncoded) average Symbol Error Rate (SER) as a function of receiver SNR (per receiver branch) for some selected training levels and SNR is defined as

$$\text{SNR} = K \cdot (\sigma_S^2/\sigma_N^2)$$

where σ_S^2 is the average source power and σ_N^2 is the power of additive noise. We compare the proposed algorithm's performance to the semi-blind equalization algorithm based on the combination of Constant-Modulus-Algorithm and Decision Directed scheme (CMA-LS) [18], semi-blind channel estimation algorithm in [17] followed by the Maximum Likelihood (ML) decoder, and adaptive least squares equalization. All these algorithms use $\tau_T = 50$ training symbols. For the CMA-LS scheme, we considered packet lengths $\Gamma = 100$ and $\Gamma = 300$. For the semi-blind channel estimation based ML

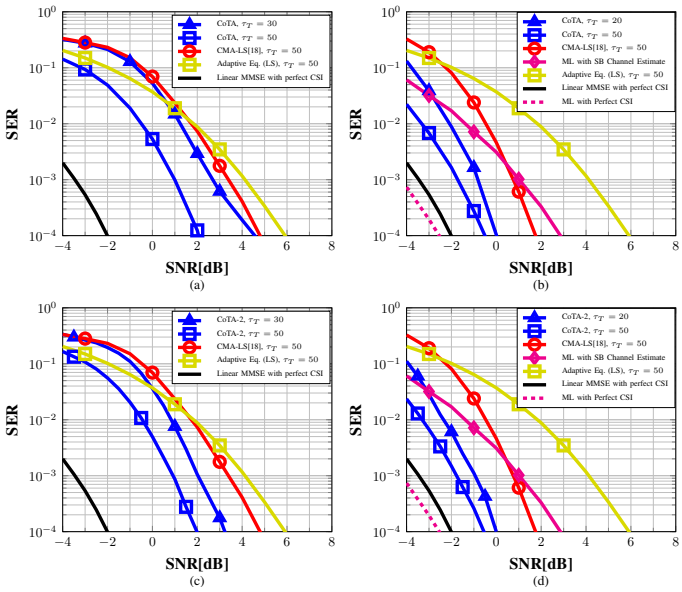
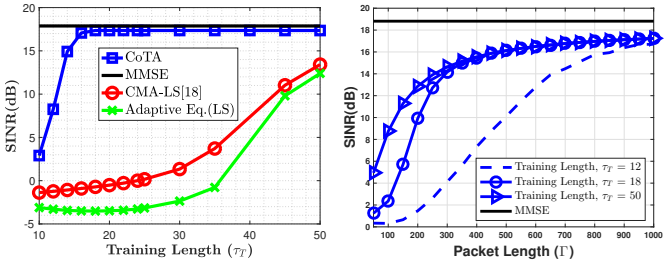


Fig. 7: Symbol Error Rate vs. SNR results for the Noisy Massive MIMO Experiment with 1000 antennas for the algorithms MIMO CoTA ((a),(b)) and MIMO CoTA-2 ((c),(d)), and for some selected training lengths indicated with different curves. The packet lengths considered are $\Gamma = 100$ ((a),(c)) and $\Gamma = 300$ ((b),(d)).

approach, we only considered $\Gamma = 300$, as the $\Gamma = 100$ case lead to poor channel estimates causing long running times for the sphere decoder. As benchmarks, we include SER performances for both Minimum Mean Square Error (MMSE) equalizer and ML sphere decoder assuming perfect Channel State Information (CSI). The algorithm we implement is the decision directed compressed training approach using Nesterov's accelerated iterations. As the algorithm parameters, we used: the regularization constant $\lambda = 0.15$ if the number of training symbols is less than number of users, and $\lambda = 0.05$ if training symbol length exceeds the number of users, the vicinity threshold of a vector to a vertex point α is set to 0.85, the accelerated method parameters $\upsilon^{(t)} = 3 \times 10^{-4}$ and $\kappa^{(t)} = \frac{t-1}{t+2}$. These parameters are fixed for all experiments given in this section. Here, we need to note that the parameter selection for the measurements is done empirically. We first select one SNR level, a packet length and a training length randomly. The only constraint on the parameter selection is to choose a training length corresponding to an underdetermined case. Our goal is to tune hyper-parameters in such a way that the performance gap between the proposed setup and genie aided MMSE is as small as possible. We need to remind that the hyper-parameters, we empirically obtain, might not lead to the optimum performance results of the proposed iterative algorithms. Despite the suboptimal choice of the hyper-parameters, the proposed approach provides a significant performance gain as illustrated by the examples in this section. Furthermore, after obtaining these parameters, we use the same hyper-parameter values for varying packet length, training length and SNR values. We only update our parameters if the considered modulation scheme or optimization setting change. We follow such a direction to show the power of the proposed algorithm in terms of parameter selection. Although our results do not reflect the best performance of our algorithms for a given setup, it outperforms the existing algorithms.

For the experiments of MIMO CoTA-2, we used: the regularization constants $\lambda_1 = 2.5$, $\lambda_2 = 15$, the vicinity threshold α is set to 0.95, the accelerated method parameter $\upsilon^{(t)} = 1$ and $\kappa^{(t)} = \frac{t-1}{t+2}$.

The results of these experiments are shown in Fig. 7. We first note that MMSE equalizer with perfect CSI has very close performance to ML with perfect CSI, as expected for high number of base station antennas. The proposed algorithms obtain a relatively close performance to the benchmark linear (MMSE) equalizer's performance for as low as $\tau_T = 20$ training symbols as the packet length increases. The adaptive least squares approach's performance is far from this even for the training length of 50. Both CMA-LS [18] and semi-blind channel estimate based ML [17] approaches provide improvements over the adaptive least squares, however, even with $\tau_T = 50$ training samples, they underperform the proposed algorithm with $\tau_T = 30$ and $\tau_T = 20$ training symbols for the communication scenarios with packet lengths $\Gamma = 100$ (Fig. 7.a,c) and $\Gamma = 300$ (Fig. 7.b,d) respectively. We also provide the performances when the number of embedded training symbols into each transmission packet is 50. As it is expected, increasing the number of training symbols enhances the performance of the proposed algorithm. Moreover, although the performance of MIMO CoTA-2 is better than MIMO CoTA when $L_D = 100$ and $L_T = 30$ due to consideration of noise on the magnitude of the equalizer, the performances of the proposed algorithms are similar. Therefore, utilizing MIMO CoTA can be advantageous in terms of complexity when the packet length or training length is enough.



(a) (b)

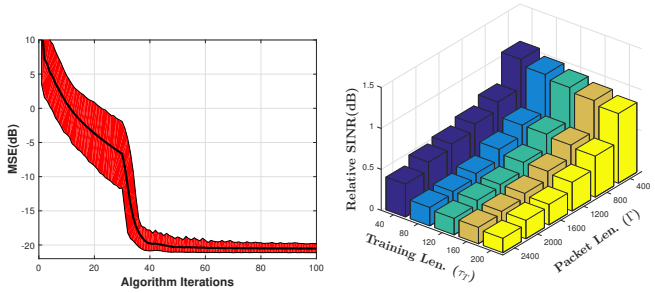
Fig. 8: a) Equalizer output SINR as a function of Training Length (τ_T) for the Noisy Massive MIMO with 1000 antennas. SNR per receiver branch is fixed as 5dB and the packet length is $\Gamma = 300$ b) The impact of the packet length on the MIMO CoTA algorithm's performance for the Noisy Massive MIMO with 1000 antennas. SNR per receiver branch is fixed as 2dB for two different training lengths.

We also look at Signal-to-Interference plus Noise Ratios (SINRs) at the outputs of the equalizer as a function of training length τ_T where SINR is defined as

$$\frac{\|\mathrm{diag}\,(\boldsymbol{W}_*\boldsymbol{H})\,\|_2^2}{\|\mathrm{idiag}\,(\mathrm{diag}\,(\boldsymbol{W}_*\boldsymbol{H})) - \boldsymbol{W}_*\boldsymbol{H}\|_F^2 + \|\boldsymbol{W}_*\|_F^2 \cdot \sigma_N^2}$$

where $\boldsymbol{W}_*$ is the optimal equalizer coefficient matrix of any algorithms. Fig. 8a displays the results for a fixed SNR level

of 5dB (per antenna branch). For this experiment, we kept λ fixed and equal to 0.1. According to this figure, the proposed algorithm captures SINR performance very close to the MMSE equalizer for a training length of $\tau_T = 15$, whereas the adaptive LS equalizer and CMA-LS algorithms still fall short of this benchmark performance even for $\tau_T = 50$.



(a) (b)

Fig. 9: a) The mean square error convergence of a user's equalizer output as a function of iterations (black solid line) with 95% confidence region (shaded area) for the Noisy Massive MIMO Experiment with 1000 antennas. SNR per receiver branch is fixed as 5dB and $\tau_T = 20$. Ph.II (Decision Directed Phase) starts at iteration 30. b) Relative SNR to achieve 10^{-4} SER level w.r.t. MMSE.

It is interesting to investigate the impact of the packet length Γ on the performance of the algorithm. Fig. 8b displays the SINR performance for an equalizer output as a function of the packet length Γ for three different training lengths, namely $\tau_T = 12$, $\tau_T = 18$, and $\tau_T = 50$, and for the receiver branch SNR=2dB. We fixed $\lambda = 0.1$ for all packet length scenarios. Based on this figure, we can make the following observations: as the training length decreases the algorithm demands larger packet lengths to achieve the same performance level.

Finally, Fig. 9a illustrates the convergence behavior of the MIMO CoTA. In this figure, we plot the mean square error which is defined as

$$\text{MSE[dB]} = 10\log_{10}\left(\|\boldsymbol{I} - \boldsymbol{W}_*\boldsymbol{H}\|_F^2 + \|\boldsymbol{W}_*\|_F^2 \cdot \sigma_N^2\right)$$

and its 95% confidence interval for equalizer outputs as a function of algorithm iterations. The decision directed phase (Ph.II) starts after iteration 30. The figure illustrates that the algorithm iterations are typically smooth and the confidence interval gets narrower as more iterations are progressed which indicates the convergence reliability of the proposed algorithm.

In Fig. 9b, we exploit the same experimental parameters as in Fig. 7. We plot the SNR gap between MIMO CoTA and MMSE to obtain 10^{-4} SER level for different packet lengths and numbers of training symbols. We observe that as the number of training symbol and packet length increases, the gap between MIMO CoTA and MMSE decreases. For example, when $\tau_T = 200$ and $\Gamma = 2400$, the gap decreases up to 0.17dB which indicates the proposed algorithm takes advantage of both training and information symbols.

C. Noisy Case Example II

In order to investigate how the proposed method works for varying constellation sizes and varying number of base station (BS) antennas in the noisy scenario, we performed several experiments whose results are provided in Fig. 10. The SER vs. SNR behaviour in Fig. 10a demonstrates the performance of the algorithm when the number of BS antennas is set to 200. We observe that as the number of BS antennas decreases, required SNR to achieve the same performance level increases as expected due to the decrease in the array gain. The impact of increasing constellation size is demonstrated by the simulation results for 16-QAM, 64-QAM and 256-QAM which are given in Fig. 10b, Fig. 10c and Fig. 10d, respectively. For these experiments, the number of BS antennas is set to 1000. As the constellation size increases, the required SNR to achieve the same SER performance level also increases, which is as expected due to the reduced minimum distance between the constellation points. These simulations also demonstrate that for reasonable packet lengths, the performance of the proposed algorithm is very close to the MMSE benchmark.

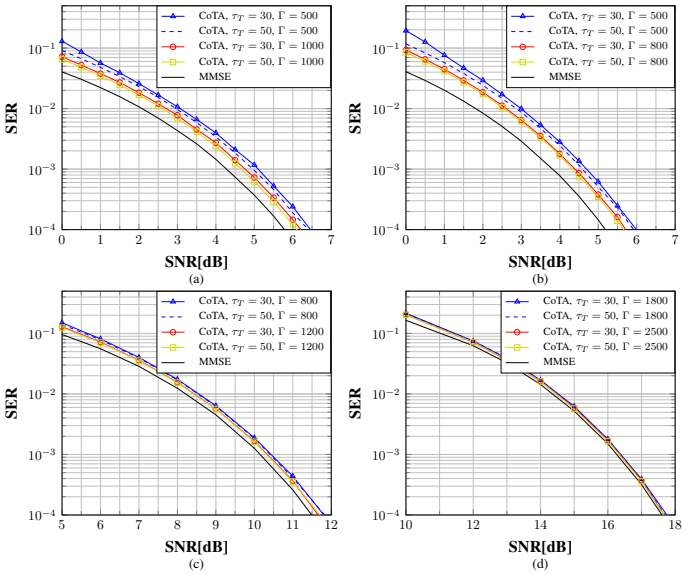


Fig. 10: Symbol Error Rate vs. SNR results for the Noisy Massive MIMO Experiment for the algorithms MIMO CoTA when a) # of BS antennas is 200, $K = 40$, 4-QAM, b) # of BS antennas is 1000, $K = 40$, 16-QAM, c) # of BS antennas is 1000, $K = 40$, 64-QAM, d) # of BS antennas is 1000, $K = 40$, 256-QAM for some selected training and packet lengths indicated with different curves.

VI. Conclusion

We introduce a new adaptive framework for training linear transceivers in the base stations of TDD Massive MIMO systems. Through this new *Compressed Training MIMO* framework, we can reduce the required training length from the number of users K to a value around $\log_2(K)$. This provides an important gain in terms of increasing system capacity and/or throughput. Furthermore, the proposed approach can be utilized to counteract against the pilot contamination problem, or to enable larger capacity cell free Massive MIMO systems [25] where the logarithmic gain would be even more emphasized.

We also show that the proposed semi-blind scheme works for practically reasonable packet lengths, which is proportional to number of users and square root of the constellation size. Together with reduced training size, this would enable communication in high mobility where the coherence times and allowed packet lengths are relatively small.

The new framework is built upon some convex optimization settings, where the main idea is to use infinity norm of the equalizer outputs to enforce sparsity on the transfer function of the overall link. The convexity is an attractive feature enabling efficient and reliable algorithms, as well as their analysis. In fact, we were able to provide analysis results for phase transitions and the related prescriptions corresponding to both training length and packet length choices, utilizing convexity and random matrix theory. Furthermore, the link established with the compressed sensing, or sparsity driven research is also very valuable. Recently, there has been a great surge of activity for developing very low complexity algorithms for high dimensional data with the emphasis on non-smooth optimization settings. These results can be utilized to address the problem of designing low complexity adaptive transceivers for large-scale MIMO receivers.

We conclude by noting that the proposed scheme can also be used for the point-to-point MIMO equalizer adaptation, where the training size can be reduced to a value close the logarithm of the number of transmit antennas.

Appendix I
Proof of Corollary I

We will first decompose *Setting IIIc* as the collection of K optimizations in *Setting Ic* for individual rows. We look at the success probability of *Setting Ic* as a function of training length, and then generalize it to the success probability of *Setting IIIc*. Clearly, the standard basis vector $\boldsymbol{e}_k$ is definitely a solution of *Setting Ic*. However, we would like it to be the unique solution. For this purpose, we will make use of the following lemma:

Lemma I: If no row of $\boldsymbol{S}_T$ other than $\boldsymbol{S}_{Tk,:}$ is chosen from the set $\mathfrak{S} = \{e^{j\theta}\boldsymbol{S}_{Tk,:} | \theta \in \mathfrak{A}\}$ where $\mathfrak{A}$ is the set $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ for β-QAM, then $\boldsymbol{e}_k$ is the unique solution of *Setting Ic*. Furthermore, when $\tau_T > log_4(K-1)+1$, this condition holds with probability at least

$$1 - (K-1)/(4^{\tau_T - 1}).$$

Proof: We adapt the result of Corollary 4.2 in [10].

Therefore, Lemma I outlines the success probability of *Setting Ic* for a given row of $\boldsymbol{G}$. The success probability of *Setting IIIc* is equal to the probability of the simultaneous success of settings corresponding to all K rows of $\boldsymbol{G}$. This is equivalent to the condition that no row of $\boldsymbol{S}_T$ should be equal to any other row or its rotated versions with the angle $\theta \in \mathfrak{A}$. This implies that if we pick any distinct two rows of $\boldsymbol{S}_T$, their inner products should satisfy

$$|\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H| \neq \|\boldsymbol{S}_{Tk,:}\|_2 \|\boldsymbol{S}_{Tl,:}\|_2$$

for $k \neq l$, as they are not aligned in the same or its rotated directions. We note that as the $\boldsymbol{S}_T$ is constructed from the corners of the constellation, we have $\|\boldsymbol{S}_{Tk,:}\|_2 = \mathcal{M}\sqrt{\tau_T}$ for all $k = 1, \dots K$ where $\mathcal{M}$ is $(\sqrt{\beta}-1)\sqrt{2}$ for β-QAM. So, the above condition can be rewritten as

$$|\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H| \neq \mathcal{M}^2\tau_T.$$

Therefore, for the probability of simultaneous success of K Setting Ic optimizations, we can write

$$\begin{aligned}
&\mathbb{P}\left(\bigcap_{k=1}^{K-1}\bigcap_{l=k+1}^{K}\{|\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H| \neq \mathcal{M}^2\tau_T\}\right) \qquad (26)\\
&= 1 - \mathbb{P}\left(\bigcup_{k=1}^{K-1}\bigcup_{l=k+1}^{K}\{|\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H| = \mathcal{M}^2\tau_T\}\right)\\
&\overset{(a)}{\geq} 1 - \sum_{k=1}^{K-1}\sum_{l=k+1}^{K}\mathbb{P}\left(\{|\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H| = \mathcal{M}^2\tau_T\}\right)\\
&= 1 - \frac{K^2-K}{2}\frac{1}{4^{\tau_T-1}} \qquad (27)\\
&= 1 - \frac{K(K-1)}{2\cdot 4^{\tau_T-1}}, \qquad (28)
\end{aligned}$$

where,

- (a) is obtained by applying union bound,
- $\frac{K^2-K}{2}$ in (27) is the number of distinct (k,l) pairs in the double summation in the previous step,
- The probability term in the argument of double summation can be written as

$$\begin{aligned}
&\mathbb{P}\left(\{|\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H| = \mathcal{M}^2\tau_T\}\right)\\
&= \sum_{\theta\in\mathfrak{A}}\mathbb{P}\left(\{\boldsymbol{S}_{Tk,:}\boldsymbol{S}_{Tl,:}^H = e^{j\theta}\mathcal{M}^2\tau_T\}\right)\\
&= \sum_{\theta\in\mathfrak{A}}\frac{1}{4^{\tau_T}} = \frac{1}{4^{\tau_T-1}}.
\end{aligned}$$

For the bound obtained in (28) to produce meaningful, positive, result, we require that

$$\tau_T > \log_4(K(K-1)) + 0.5. \blacksquare \qquad (29)$$

Appendix II
Proof of Theorem II

In this appendix, we provide the proof for Theorem 1, which is based on the geometric problem of the intersection of a random subspace with the non-negative orthant. We first start by rewriting *Setting IIb* as

$$\begin{aligned}
\textit{Setting IIc:}\quad &\underset{\boldsymbol{G}_{k,:}}{\text{minimize}} \quad J(\tilde{\boldsymbol{G}}_{k,:}) = \frac{1}{\sqrt{\beta}-1}\|\tilde{\boldsymbol{G}}_{k,:}\tilde{\boldsymbol{S}}\|_\infty\\
&\text{s.t.} \qquad \tilde{\boldsymbol{G}}_{k,:}\tilde{\boldsymbol{S}}_T = \tilde{\boldsymbol{s}}_T^T.
\end{aligned}$$

where $\boldsymbol{s}_T^T = \boldsymbol{S}_{Tk,:}$ and $\boldsymbol{G}_{k,:} = \boldsymbol{W}_{k,:}\boldsymbol{H}$. The affine constraint corresponds to the feasible set $\mathcal{F}_k = \{\boldsymbol{e}_k^T + \boldsymbol{\eta}^T | \boldsymbol{\eta} \in N_L(\tilde{\boldsymbol{S}}_T)\}$. Regarding *Setting IIc*, we make the following observations:

i. $\boldsymbol{e}_k^T \in \mathcal{F}_k$ and $J(\boldsymbol{e}_k^T) = 1$ is an upper bound on the optimal value,
ii. The training signals $\boldsymbol{s}_T$ constructed from the corner points of the constellation, and the training symbols in $\tilde{\boldsymbol{s}}$ are the subset of elements of $\tilde{\boldsymbol{G}}_{k,:}\tilde{\boldsymbol{S}}$.

As a result, the optimal value for *Setting IIc* is equal to 1, and $\boldsymbol{e}_k^T$ is an optimal solution. We are interested in bounding the probability that the set of optimal solutions $\mathcal{O}_k$ is equal to the singletone set $\{\boldsymbol{e}_k^T\}$.

Observe that the entries in $\tilde{\boldsymbol{S}}$ are not independent of each other due to isomorphic operation in (11). Therefore, we column-wise partition $\tilde{\boldsymbol{S}}$ into two matrices, $\boldsymbol{S}^{(A)} = \tilde{\boldsymbol{S}}_{:,1:\Gamma}$ and $\boldsymbol{S}^{(B)} = \tilde{\boldsymbol{S}}_{:,\Gamma+1:2\Gamma}$, such that all the entries within each partition are independent. Moreover, since the cost function value is determined by the inner products of the constraint set vectors with the columns of $\tilde{\boldsymbol{S}}$, we further column-wise partition $\boldsymbol{\mathcal{S}}^{(i)}$ according to whether its k^{th} row contains the peak magnitude source value $\mp(\sqrt{\beta}-1)$ where $i \in \{A, B\}$. Therefore, we define the index sets

$$\begin{aligned}
\mathcal{I}_{i1} &= \{j \mid \boldsymbol{S}^{(i)}_{k,j} = 1-\sqrt{\beta}\},\\
\mathcal{I}_{i2} &= \{j \mid \boldsymbol{S}^{(i)}_{k,j} = \sqrt{\beta}-1\},\\
\mathcal{I}_{i3} &= \{j \mid |\boldsymbol{S}^{(i)}_{k,j}| \neq \sqrt{\beta}-1\},
\end{aligned}$$

which leads to the partitioning

$$\tilde{\boldsymbol{S}} = [\boldsymbol{S}^{(A)}_{:,\mathcal{I}_{A1}}\ \boldsymbol{S}^{(A)}_{:,\mathcal{I}_{A2}}\ \boldsymbol{S}^{(A)}_{:,\mathcal{I}_{A3}}\ \boldsymbol{S}^{(B)}_{:,\mathcal{I}_{B1}}\ \boldsymbol{S}^{(B)}_{:,\mathcal{I}_{B2}}\ \boldsymbol{S}^{(B)}_{:,\mathcal{I}_{B3}}]\boldsymbol{P}$$

where $\boldsymbol{P}$ is a permutation matrix. It is clear that for an optimal solution $\tilde{\boldsymbol{G}}^{(o)}_{k,:} \in \mathcal{O}_k$, $\|\tilde{\boldsymbol{G}}^{(o)}_{k,:}\tilde{\boldsymbol{S}}\|_\infty = \sqrt{\beta}-1$, and therefore,

$$\frac{\|\tilde{\boldsymbol{G}}^{(o)}_{k,:}\boldsymbol{S}^{(A)}_{:,\mathcal{I}_{Aj}}\|_\infty}{\sqrt{\beta}-1} \leq 1, \text{ and } \frac{\|\tilde{\boldsymbol{G}}^{(o)}_{k,:}\boldsymbol{S}^{(B)}_{:,\mathcal{I}_{Bj}}\|_\infty}{\sqrt{\beta}-1} \leq 1, \tag{30}$$

for $j \in \{1,2,3\}$. Substituting $\tilde{\boldsymbol{G}}^{(o)}_{k,:} = \boldsymbol{e}_k^T + \boldsymbol{\eta}^{(o)^T}$ in (30), we obtain

$$\frac{\|\boldsymbol{\eta}^{(o)^T}\boldsymbol{S}^{(A)}_{:,\mathcal{I}_{A1}} - (\sqrt{\beta}-1)\mathbf{1}^T\|_\infty}{\sqrt{\beta}-1} \leq 1. \tag{31}$$

Note that $\mathcal{S}_{A1} = \{\boldsymbol{\eta}^T\boldsymbol{S}^{(A)}_{:,\mathcal{I}_{A1}} | \boldsymbol{\eta} \in N_L(\tilde{\boldsymbol{S}}_T)\}$ defines a random subspace with dimension $d_{A1} = \min(m_{A1}, d_{N_L})$ where m_{A1} is the cardinality of $\mathcal{I}_{A1}$ and $d_{N_L} = 2(K-\tau_T)$, which is the dimension of $N_L(\tilde{\boldsymbol{S}}_T)$. If this subspace intersects the non-negative orthant, $\mathbb{R}_+^{m_{A1}}$, only at the origin, only choice for $\boldsymbol{\eta}^{(o)}$ satisfying (31) is $\boldsymbol{\eta}^{(o)} = \mathbf{0}$, which corresponds to the case of unique optimal solution $\mathcal{O}_k = \{\boldsymbol{e}_k^T\}$. Based on [26], the probability of the complement condition is given by

$$\begin{aligned}
P(\mathcal{S}_{A1} \cap \mathbb{R}_+^{m_{A1}} \neq \{\mathbf{0}\}) &= \frac{1}{2^{m_{A1}-1}}\sum_{j=0}^{d_{A1}-1}\binom{m_{A1}-1}{j}\\
&\triangleq \mathbb{P}_\chi(m_{A1}, d_{A1}).
\end{aligned}$$

Therefore, we can lower bound the probability of $\boldsymbol{e}_k^T$ being the unique solution of *Setting IIc* as

$$P(\mathcal{O}_k = \{\boldsymbol{e}_k^T\}) \geq 1 - \sum_{m_{A1}=1}^{\Gamma} \mathbb{P}_{|\mathcal{I}_{A1}|}(m_{A1})\mathbb{P}_\chi(m_{A1}, d_{A1}), \tag{32}$$

where $\mathbb{P}_{|\mathcal{I}_{A1}|}(m_{A1})$ is the probability that the cardinality of $\mathcal{I}_{A1}$ is equal to m_{A1}, and it is given by

$$\mathbb{P}_{|\mathcal{I}_{A1}|}(m_{A1}) = \binom{\Gamma}{m_{A1}}\frac{1}{\sqrt{\beta}^{m_{A1}}}\left(1-\frac{1}{\sqrt{\beta}}\right)^{\Gamma-m_{A1}}.$$

The lower bound in (32) can be replaced by a more closed-form but less tight bound. For this purpose, we employ $\sum_{i=0}^{k}\binom{n}{i} \leq 2^{nH(\frac{k}{n})}$, where $H(\frac{k}{n}) = -\log_2(\frac{k}{n})\frac{k}{n} - \log_2(\frac{n-k}{n})\frac{n-k}{n}$ is the Bernoulli entropy, to bound

$$\mathbb{P}_\chi(m_{A1}, d_{A1}) \leq 2^{-(m_{A1}-1)(1-H(\frac{d_{A1}-1}{m_{A1}-1}))}. \tag{33}$$

Note that this bound is valid for $m_{A1}-1 \geq 2(d_{A1}-1)$ which implies $m_{A1}-1 \geq 2(d_{N_L}-1)$ and $d_{A1} = d_{N_L}$. This upper bound contains a nonlinear function of m_{A1} at the exponent, whereas we would like to have a linear functional so that the summation in (32) can be converted into the form of binomial expansion. In order to achieve this goal, we exploit the concavity of the exponent term. In fact, the second derivative of the exponent function is equal to,

$$-\frac{1}{m_{A1}-d_{NL}} + \frac{1}{m_{A1}-1}, \tag{34}$$

which is negative for $m_{A1} > d_{N_L}$. Therefore, we can write an affine upper bound for this function using a supporting hyperplane at a point m_o. This yields a new upper bound with an affine exponent function of the form,

$$\mathbb{P}_\chi(m_{A1}, d_{N_L}) \leq 2^{-\chi m_{A1}+\upsilon}, \tag{35}$$

where $\chi = 1-\log_2(\frac{m_o-1}{m_o-d_{N_L}})$ and $\upsilon = -(d_{N_L}-1)\log_2(d_{N_L}-1) - \log_2(m_o-1) + d_{N_L}\log_2(m_o-d_{N_L}) - 1$, m_o is the support point, used in upper bounding the concave function with the affine function, chosen in $(2d_{N_L}, \Gamma)$. This range for m_o guarantees that $\chi > 0$, so that the lower bound in (35) is strictly decreasing with m_{A1}. For $m_{A1} = 2d_{N_L}-1$, the bound in (33) is equal to 1, therefore, the bound in (35) is greater than or equal to 1 for this choice of m_{A1}. As $\chi > 0$, the bound in (35) is a decreasing function, therefore, its value for $m_{A1} < 2d_{N_L}-1$ is greater than 1. As a result, the bound in (35) is a valid upper bound for $\mathbb{P}_{|\mathcal{I}_{A1}|}(m_{A1})$ for all $m_{A1} \geq 0$, although it is useless in $m_{A1} < 2d_{N_L}-1$. However, its form simplifies the lower bound in (32) to binomial sum form.

Plugging this new upper bound (35) in (32), we obtain

$$\begin{aligned}
P(\mathcal{O}_k = \{\boldsymbol{e}_k^T\}) &\geq 1 - \sum_{m=0}^{\Gamma}\binom{\Gamma}{m}\frac{2^\upsilon\left(1-\frac{1}{\sqrt{\beta}}\right)^{\Gamma-m}}{(\sqrt{\beta}2^\chi)^m}\\
&= 1 - 2^\upsilon\left(1-\frac{1}{\sqrt{\beta}}\frac{2^\chi-1}{2^\chi}\right)^\Gamma.
\end{aligned} \tag{36}$$

The probability of unique solution simultaneously for all users can be lower bounded by modifying this bound as

$$P(\bigcap_{k=1}^{K}\mathcal{O}_k = \{\boldsymbol{e}_k^T\}) \geq 1 - K\sum_{m_{A1}=1}^{\Gamma}\mathbb{P}_{|\mathcal{I}_{A1}|}(m_{A1})\mathbb{P}_\chi(m_{A1}, d_{A1}) \tag{37}$$

$$\geq 1 - K2^\upsilon\left(1-\frac{1}{\sqrt{\beta}}\frac{2^\chi-1}{2^\chi}\right)^\Gamma. \tag{38}$$

We can also obtain an upper bound for the probability of unique optimal solution. We first note that we can define the subspaces $\mathcal{S}_{A2}$, $\mathcal{S}_{B1}$ and $\mathcal{S}_{B2}$ similar to $\mathcal{S}_{A1}$ as the image of $N_L(\tilde{\boldsymbol{S}}_T)$ under the corresponding matrix partitions. The relevant conditions for uniqueness is determined by the intersection of $\mathcal{S}_{i1}$ with $\mathbb{R}_+^{m_{i1}}$ and $\mathcal{S}_{i2}$ with $\mathbb{R}_-^{m_{i2}}$ only at the

origin for $i \in A, B$, which all have the same probability bound as (32). For the upper bound, we consider the $4-QAM$ constellation for which case, all transmit symbols come from corner points of the constellation, and therefore, $|\mathcal{I}_{A3}| = |\mathcal{I}_{B3}| = 0$. As a result, the desired upper-bound can be obtained through the union bound

$$\begin{aligned} P(\cap_{k=1}^{K}\mathcal{O}_k = \{\boldsymbol{e}_k^T\}) &\leq P(\mathcal{O}_1 = \{\boldsymbol{e}_1^T\}) \\ &= P((\bigcup_{i=A,B}\mathcal{S}_{i1}\cap\mathbb{R}_+^{m_{i1}} = \{\mathbf{0}\})\bigcup(\bigcup_{i=A,B}\mathcal{S}_{i2}\cap\mathbb{R}_-^{m_{i2}} = \{\mathbf{0}\})) \\ &\leq \sum_{i=A,B}P(\mathcal{S}_{i1}\cap\mathbb{R}_+^{m_{i1}} = \{\mathbf{0}\}) + \sum_{i=A,B}P(\mathcal{S}_{i2}\cap\mathbb{R}_-^{m_{i2}} = \{\mathbf{0}\}) \\ &\leq 4(1-\mathbb{P}_\chi(\Gamma, d_{N_L})), \end{aligned} \tag{39}$$

where the last inequality follows from the fact that $P_\chi(m_{A1}, d_{N_L})$ is minimized when the ambient dimension m_{A1} is set to its maximum value Γ. ■

REFERENCES

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

[2] T. L. Marzetta, "Massive MIMO: an introduction," *Bell Labs Technical Journal*, vol. 20, pp. 11–22, 2015.

[3] E. Bjrnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: ten myths and one critical question," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, February 2016.

[4] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010.

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

[6] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *Information Theory, IEEE Transactions on*, vol. 49, no. 4, pp. 951–963, 2003.

[7] T. L. Marzetta, "How much training is required for multiuser MIMO?" in *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, Oct 2006, pp. 359–363.

[8] H. Q. Ngo, M. Matthaiou, and E. G. Larsson, "Massive MIMO with optimal power and training duration allocation," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 605–608, 2014.

[9] B. B. Yilmaz and A. T. Erdogan, "Compressed training adaptive equalization," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4920–4924.

[10] ——, "Compressed training adaptive equalization: Algorithms and analysis," *IEEE Transactions on Communications*, vol. 65, no. 9, pp. 3907–3921, 2017.

[11] S. Vembu, S. Verdu, R. Kennedy, and W. Sethares, "Convex cost functions in blind equalization," *IEEE Trans. on Signal Processing*, vol. 42, pp. 1952–1960, August 1994.

[12] Z.-Q. Luo, M. Meng, K. M. Wong, and J.-K. Zhang, "A fractionally spaced blind equalizer based on linear programming," *IEEE Trans. on Signal Processing*, vol. 50, pp. 1650–1660, July 2002.

[13] Z. Ding and Z. Luo, "A fast linear programming algorithm for blind equalization," *IEEE TCOM.*, vol. 48, pp. 1432–1436, September 2000.

[14] A. T. Erdogan and C. Kizilkale, "Fast and low complexity blind equalization via subgradient projections," *IEEE Trans. on Signal Processing*, vol. 53, pp. 2513–2524, July 2005.

[15] B. B. Yilmaz and A. T. Erdogan, "Compressed training adaptive MIMO equalization," in *The 17th. IEEE International Workshop on Signal Processing Advances in Wireless Communications*. IEEE, July 2016.

[16] A. K. Jagannatham and B. D. Rao, "Whitening-rotation-based semi-blind MIMO channel estimation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 3, pp. 861–869, 2006.

[17] E. Nayebi and B. D. Rao, "Semi-blind channel estimation for multiuser massive mimo systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 540–553, 2018.

[18] S. Chen, W. Yao, and L. Hanzo, "Semi-blind adaptive spatial equalization for MIMO systems with high-order QAM signalling," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 11, pp. 4486–4491, 2008.

[19] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 32–43, 2014.

[20] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp, "Living on the edge: Phase transitions in convex programs with random data," *Information and Inference: A Journal of the IMA*, vol. 3, no. 3, pp. 224–294, 2014.

[21] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[22] R. Baraniuk, M. A. Davenport, M. F. Duarte, and C. Hegde, "An introduction to compressive sensing," *Connexions e-textbook*, 2011.

[23] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[24] W. Su, S. Boyd, and E. J. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.

[25] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO: Uniformly great service for everyone," in *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2015, pp. 201–205.

[26] K. E. Morrison, "The probability that a subspace contains a positive vector," *ArXiv e-prints*, Apr. 2010.

Baki Berkay Yilmaz (S'16) received the B.Sc. and M.Sc. degrees in Electrical and Electronics Engineering from Koc University, Turkey in 2013 and 2015 respectively. He joined Georgia Institute of Technology in Fall 2016 and he is currently pursuing his PhD in School of Electrical and Computer Engineering, focusing on quantifying covert/side-channel information leakage and capacity. He also works on channel equalization and sparse reconstruction. His research interests span areas of electromagnetics, signal processing and information theory.

Alper T. Erdogan (M'00-SM'12) was born in Ankara, Turkey, in 1971. He received the B.S. degree from the Middle East Technical University, Ankara, Turkey, in 1993, and the M.S. and Ph.D. degrees from Stanford University, CA, in 1995 and 1999, respectively. He was a Principal Research Engineer with the Globespan-Virata Corporation (formerly Excess Bandwidth and Virata Corporations) from September 1999 to November 2001. He joined the Electrical and Electronics Engineering Department, Koc University, Istanbul, Turkey, in January 2002, where he is currently a Professor. His research interests include wireless, fiber and wireline communications, adaptive signal processing, optimization, system theory and control, and information theory. Dr. Erdogan is the recipient of several awards including TUBITAK Career Award (2005), Werner Von Siemens Excellence Award (2007), TUBA GEBIP Outstanding Young Scientist Award (2008), TUBITAK Encouragement Award (2010) and Outstanding Teaching Award (2017). He served as an Associate Editor for the IEEE Transactions on Signal Processing, and he was a member of IEEE Signal Processing Theory and Methods Technical Committee.