

Soft-Output Joint Channel Estimation and Data Detection using Deep Unfolding

Haochuan Song^{1,2}, Xiaohu You^{1,2}, Chuan Zhang^{1,2}, and Christoph Studer³

¹*National Mobile Communications Research Laboratory, Southeast University, Nanjing, China*

²*Purple Mountain Laboratories, Nanjing, China*

³*Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich, Switzerland*
e-mail: {hcsong, xhyu, chzhang}@seu.edu.cn; studer@ethz.ch

Abstract—We propose a novel soft-output joint channel estimation and data detection (JED) algorithm for multiuser (MU) multiple-input multiple-output (MIMO) wireless communication systems. Our algorithm approximately solves a maximum a-posteriori JED optimization problem using deep unfolding and generates soft-output information for the transmitted bits in every iteration. The parameters of the unfolded algorithm are computed by a hyper-network that is trained with a binary cross entropy (BCE) loss. We evaluate the performance of our algorithm in a coded MU-MIMO system with 8 basestation antennas and 4 user equipments and compare it to state-of-the-art algorithms separate channel estimation from soft-output data detection. Our results demonstrate that our JED algorithm outperforms such data detectors with as few as 10 iterations.

I. INTRODUCTION

In the uplink of multiuser multiple-input multiple-output (MU-MIMO) systems, where user equipments (UEs) transmit pilots and data to a base station (BS), deploying optimal joint channel estimation and data detection (JED) is an elusive goal, and has been practicable only for small-scale MIMO systems due to the combinatorial nature and the high-dimensionality of the JED problem [1]–[3]. To overcome the high complexity of such methods, references [4], [5] proposed gradient-based algorithms that efficiently compute approximate solutions to the single-UE JED problem for large BS antenna arrays. The multi-UE JED case has been tackled recently in [6] for cell-free massive MU-MIMO systems, also using a gradient-based algorithm. All of these methods compute hard-output estimates and are, thus, unable to realize the full potential of coded data transmission. A soft-output algorithm that approximates JED using iterative estimation and detection (IED) has been proposed recently in [7], which alternates between channel estimation and data detection. Similar alternating optimization methods have been used before for IED in [8], [9].

A. Contributions

With the recent progress in deep neural networks, optimal JED [1]–[3] is suddenly within reach by leveraging deep

The work of HS, XY, and CZ was supported in part by the National Key R&D Program of China under Grant 2020YFB2205503, by the NSFC under Grants 61871115 and 62122020, and by the Jiangsu NSF under Grant BK20211512. The work of CS was supported in part by ComSenTer, one of six centers in JUMP, a SRC program sponsored by DARPA, by an ETH Research Grant, and by the US NSF under grants CNS-1717559 and ECCS-1824379.

The authors thank Gian Marti for comments and suggestions.

unfolding of iterative algorithms [7], [10]–[13]. In this paper, we propose a novel soft-output JED algorithm that builds upon a maximum a-posteriori (MAP) JED problem formulation which we solve approximately using deep unfolding. We derive an iterative algorithm with soft-output capabilities by utilizing the approximate posterior mean estimator (PME) put forward in [14], [15]. All algorithm parameters are generated by a hyper-network that processes estimated channel state information (CSI). The hyper-network is trained with a binary cross entropy (BCE) loss that exploits the soft-output capabilities of our JED algorithm. We provide simulation results for an 8 BS antenna, 4 UE MU-MIMO system and compare our algorithm to state-of-the-art methods that separate channel estimation from soft-output data detection.

B. Notation

Lower case letters denote matrices and upper case boldface letters denote vectors. We use $A_{b,u}$, \mathbf{a}_u , and b_k to represent the entry in the b th row and u th column of the matrix \mathbf{A} , the u th column of matrix \mathbf{A} , and the k th entry in the vector \mathbf{b} , respectively. The superscripts T and H denote the transpose and Hermitian transpose, respectively. The Frobenius norm and trace of a matrix \mathbf{A} is $\|\mathbf{A}\|_F$ and $\text{Tr}(\mathbf{A})$. The $U \times U$ identity matrix is \mathbf{I}_U . Sets are denoted by calligraphic letters and the cardinality of \mathcal{Q} is $|\mathcal{Q}|$. The operator \mathbb{E} denotes expectation.

II. PREREQUISITES

We now introduce the system model and MAP-JED optimization problem from which we derive a computationally efficient soft-output JED algorithm in Section III.

A. System Model

We focus on the uplink of a MU-MIMO communication system in which U single-antenna UEs transmit pilots and data to a BS equipped with B antennas. We assume a block-fading scenario with a coherence time of $K = T + D$ time slots; T time slots are reserved for pilots and D time slots are used for payload data. The transmitted data matrix $\mathbf{S} = [\mathbf{S}_T, \mathbf{S}_D]$ contains the pilots $\mathbf{S}_T \in \mathbb{C}^{U \times T}$ and the transmit symbols $\mathbf{S}_D \in \mathcal{Q}^{U \times D}$ of all UEs, where \mathcal{Q} is the constellation set

(e.g., QPSK). In what follows, we consider a frequency-flat channel¹ with the following input-output relation [16]:

$$\mathbf{Y} = \mathbf{H}\mathbf{S} + \mathbf{N}. \quad (1)$$

Here, $\mathbf{Y} \in \mathbb{C}^{B \times K}$ is the receive matrix containing all received symbols at the B BS antennas over the K time slots, $\mathbf{H} \in \mathbb{C}^{B \times U}$ is the (unknown) channel matrix, and $\mathbf{N} \in \mathbb{C}^{B \times K}$ models thermal noise with i.i.d. circularly-symmetric complex Gaussian entries and variance N_0 per complex dimension.

B. MAP-JED Optimization Problem

We start by formulating the MAP-JED problem. Our goal is to *jointly* estimate the channel matrix and recover the most likely transmit symbols which requires us to assume priors for the channel matrix and the transmit symbols. For simplicity, we assume i.i.d. circularly-symmetric Gaussian entries in \mathbf{H} with entry-wise variance E_h and equally likely transmit symbols. These assumptions result in the MAP-JED problem

$$\{\hat{\mathbf{H}}, \hat{\mathbf{S}}_D\} = \arg \min_{\substack{\mathbf{H} \in \mathbb{C}^{B \times U} \\ \mathbf{S}_D \in \mathcal{Q}^{U \times D}}} \|\mathbf{Y} - \mathbf{H}\mathbf{S}\|_F^2 + \lambda \|\mathbf{H}\|_F^2 \quad (2)$$

with $\lambda = N_0/E_h$.² For simplicity of exposition, we will directly work with the transmit symbol matrix \mathbf{S} instead of the pilot and data matrices \mathbf{S}_T and \mathbf{S}_D , respectively.

Since (2) can be written as two nested optimization problems in the variables \mathbf{S} and \mathbf{H} , we can first determine the optimal channel estimate $\hat{\mathbf{H}}$ given the transmit symbol matrix \mathbf{S} and then find the optimal transmit symbol matrix $\hat{\mathbf{S}}$. Since the optimization problem is quadratic in \mathbf{H} , the optimal channel estimate $\hat{\mathbf{H}}$ has the following closed-form solution:

$$\hat{\mathbf{H}} = \mathbf{Y}\mathbf{S}^H\mathbf{M}^{-1}, \quad (3)$$

where we use the auxiliary matrix $\mathbf{M} = \mathbf{S}\mathbf{S}^H + \lambda\mathbf{I}_U$. We can now substitute $\hat{\mathbf{H}}$ into the objective of (2) and perform algebraic simplifications, which leads to an equivalent MAP-JED problem that only depends on the transmit symbol matrix:

$$\hat{\mathbf{S}} = \arg \max_{\mathbf{S} \in \mathcal{Q}^{U \times K}} \text{Tr}[\mathbf{Y}^H\mathbf{Y}\mathbf{S}^H\mathbf{M}^{-1}\mathbf{S}]. \quad (4)$$

After solving the MAP-JED problem in (4), one can determine the optimal channel estimate by plugging $\hat{\mathbf{S}}$ into (3). Note that for $\lambda = 0$, the MAP-JED problem in (4) reduces to the well-known maximum likelihood JED problem in [3, Eq. 6].

III. S-JED: SOFT-OUTPUT JOINT CHANNEL ESTIMATION AND DATA DETECTION

The combinatorial nature of solving (4) exactly would quickly result in prohibitive complexity which necessitates approximate algorithms. Furthermore, even solving (4) approximately would lead to a hard-output JED method that is unable to realize the full potential of coded data transmission. We now derive a computationally efficient algorithm to approximately solve (4) while being able to compute soft-output information.

¹Frequency selective channels can be transformed into parallel frequency-flat subcarriers by means of orthogonal frequency-division multiplexing.

²In Section IV, we let the hyper-network determine an appropriate choice of the parameter $\lambda^{(t)}$ for every iteration t , which enables us to apply our JED algorithm to channel matrices that are not necessarily i.i.d. Gaussian.

A. Smoothing the MAP-JED Problem

The discrete nature of the constellation \mathcal{Q} is the culprit of preventing gradient-descent-like methods (and deep unfolding) to approximately solve (4). We therefore propose to first relax the set \mathcal{Q} to its convex hull, which is defined as [5]

$$\mathcal{C} = \left\{ \sum_{i=1}^{|\mathcal{Q}|} \alpha_i s_i \mid (\alpha_i \in \mathbb{R}_+, \forall i) \wedge \sum_{i=1}^{|\mathcal{Q}|} \alpha_i = 1 \right\}. \quad (5)$$

Here, s_i is the i th symbol in the constellation \mathcal{Q} . We can now replace the discrete constellation \mathcal{Q} in (4) by the convex set \mathcal{C} , which leads to the smoothed optimization problem

$$\hat{\mathbf{S}}^{\text{sm}} = \arg \max_{\mathbf{S} \in \mathcal{C}^{U \times K}} \text{Tr}[\mathbf{Y}^H\mathbf{Y}\mathbf{S}^H\mathbf{M}^{-1}\mathbf{S}]. \quad (6)$$

The resulting smoothed optimization problem has a differentiable objective and a convex constraint, which permits the use of gradient-based methods and deep unfolding.

B. Smoothed MAP-JED via Forward-Backward Splitting

We now show how to use forward-backward splitting (FBS) [17] to approximately solve (6). FBS iteratively solves convex optimization problems of the form

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} f(\mathbf{s}) + g(\mathbf{s}), \quad (7)$$

where the function f is differentiable and convex, and g is convex but not necessarily smooth or bounded. By initializing FBS with $\mathbf{s}^{(1)}$, it solves the problem in (7) for the iterations $t = 1, 2, \dots$ until convergence by computing

$$\mathbf{s}^{(t+1)} = \text{prox}_g(\mathbf{s}^{(t)} - \tau^{(t)} \nabla f(\mathbf{s}^{(t)}); \tau^{(t)}), \quad (8)$$

where $\nabla f(\mathbf{s})$ is the gradient of $f(\mathbf{s})$ and $\tau^{(t)}$ is a carefully-chosen step size at iteration t . The proximal operator for $g(\mathbf{s})$ is defined as follows [18]:

$$\text{prox}_g(\mathbf{s}; \tau) = \arg \min_{\mathbf{z}} \tau g(\mathbf{z}) + \frac{1}{2} \|\mathbf{z} - \mathbf{s}\|_2^2. \quad (9)$$

While FBS is able to exactly solve convex optimization problems, it can be used to approximately solve many non-convex problems [17], which will be described next.

For our MAP-JED problem, we define f and g in (7) as

$$f(\mathbf{S}) = \text{Tr}[\mathbf{Y}^H\mathbf{Y}\mathbf{S}^H\mathbf{M}^{-1}\mathbf{S}] \quad \text{and} \quad g(\mathbf{S}) = \chi_{\mathcal{C}}(\mathbf{S}), \quad (10)$$

where the indicator function $\chi_{\mathcal{C}}(\mathbf{S})$ implements the convex constraint $\mathbf{S} \in \mathcal{C}^{U \times K}$ in (6), and is zero if $\mathbf{S} \in \mathcal{C}$ and infinity otherwise. The gradient of the function f in \mathbf{S} is given by

$$\nabla f(\mathbf{S}) = \mathbf{M}^{-1}\mathbf{S}\mathbf{Y}^H\mathbf{Y}(\mathbf{I}_U - \mathbf{S}^H\mathbf{M}^{-1}\mathbf{S}). \quad (11)$$

Due to space constraints, the derivation of the gradient will be shown in the future journal version of the paper [19]. The proximal operator in (9) for \mathbf{S} is given by

$$\text{prox}_g(\mathbf{S}; \tau^{(t)}) = \arg \min_{\mathbf{X} \in \mathcal{C}^{U \times K}} \frac{1}{2} \|\mathbf{X} - \mathbf{S}\|_F^2, \quad (12)$$

where we move the indicator function in $g(\mathbf{S})$ back to the constraint. This problem has closed-form expressions for most constellation sets, e.g., for QPSK the projection operator is

$$\text{proj}_{\mathcal{C}}(\mathcal{R}\{S_{i,j}\}) = \min\{\max\{|\mathcal{R}\{S_{i,j}\}|, -\alpha\}, \alpha\} \quad (13)$$

$$\text{proj}_{\mathcal{C}}(\Im\{S_{i,j}\}) = \min\{\max\{|\Im\{S_{i,j}\}|, -\alpha\}, \alpha\}, \quad (14)$$

where $\alpha = \frac{1}{\sqrt{2}}$ for the set $\mathcal{Q} = \{\pm \frac{1}{\sqrt{2}} \pm \frac{1}{\sqrt{2}}j\}$.

This FBS algorithm efficiently (and approximately) solves the smoothened MAP-JED problem in (6), but ignores the discrete nature of the constellation set and is unable to compute soft-outputs. Sections III-C and III-D address these issues.

C. Exploiting the Constellation with the PME

In order to exploit the constellation set \mathcal{Q} , we model the iterations of FBS after evaluating the gradient step

$$\mathbf{X}^{(t)} = \mathbf{S}^{(t)} - \tau^{(t)} \nabla f(\mathbf{S}^{(t)}) \quad (15)$$

as follows:

$$\mathbf{X}^{(t)} = \mathbf{S} + \mathbf{E}^{(t)}. \quad (16)$$

Here, \mathbf{S} is the (unknown) true transmitted data matrix and $\mathbf{E}^{(t)}$ models estimation errors on these per-iteration estimates. By assuming that the distribution of $\mathbf{E}^{(t)}$ is known, one can replace the projection onto the convex hull \mathcal{C} in (12) by the entry-wise posterior mean estimate (PME)

$$S_{u,k}^{(t+1)} = \mathbb{E}[S_{u,k} | X_{u,k}^{(t)}] \quad (17)$$

with $u = 1, \dots, U$ and $k = 1, \dots, K$. We emphasize that the PME depends on the prior distribution (which is given by the constellation set \mathcal{Q}) and the statistics of $\mathbf{E}^{(t)}$. We have the following prior distribution on the transmitted data:

$$p(S_{u,k}) = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \delta(S_{u,k} - s_i). \quad (18)$$

Here, s_i is the i th symbol in the constellation \mathcal{Q} and δ is the Dirac delta function. By assuming that the estimation errors $E_{u,k}^{(t)}$ are circularly-symmetric complex Gaussian with variance $\nu_{u,k}^{(t)}$, the PME in (17) has a closed form (see, e.g., [20]) and one can replace the proximal operator in (12) with the PME in (17). This step requires knowledge of the per-iteration estimation error variances $\nu_{u,k}^{(t)}$, $\forall u, k$, which are difficult to obtain in practice. However, as shown in Section IV, we can use a hyper-network to determine these variances.

D. Approximate PME and Soft-Output Generation

In order to extract soft-outputs, we build upon the approximate PME put forward in [14], [15]. The key idea is to replace (17) by an approximate three-step procedure that (i) converts the per-iteration estimates in (15) into soft-outputs in the form of log-likelihood ratios (LLRs) for every transmitted bit, (ii) transforms these LLR values into probabilities, and (iii) converts these probabilities back into the soft-symbol estimates. We now summarize this approach for QPSK modulation.

Step (i): Assume that the per-iteration estimation errors are circularly-symmetric complex Gaussian with variances $\nu_{u,k}^{(t)}$, $\forall u, k$. Then, the LLRs for the two bits that map to QPSK symbols are given by [21, Tbl. 4.3]

$$L_{1,u,k}^{(t)} = \frac{4\Re\{X_{u,k}^{(t)}\}}{\nu_{u,k}^{(t)}} \quad \text{and} \quad L_{2,u,k}^{(t)} = \frac{4\Im\{X_{u,k}^{(t)}\}}{\nu_{u,k}^{(t)}}. \quad (19)$$

Step (ii): We convert the LLRs in (19) into probabilities as follows [21, Eq. 3.6]:

$$P_{b,u,k}^{(t)} = \frac{1}{2} \left(1 + \tanh \left(\frac{L_{b,u,k}^{(t)}}{2} \right) \right), \quad b \in \{1, 2\}, \quad (20)$$

which express the probabilities of the b th bit that map to the symbol $S_{u,k}$ (e.g., using Gray mapping) being 1.

Step (iii): We use the probabilities in (20) to compute symbol estimates as follows [21, App. A.4]:

$$\Re\{S_{u,k}^{(t+1)}\} = \frac{1}{\sqrt{2}} (2P_{1,u,k}^{(t)} - 1) \quad (21)$$

$$\Im\{S_{u,k}^{(t+1)}\} = \frac{1}{\sqrt{2}} (2P_{2,u,k}^{(t)} - 1), \quad (22)$$

where $S_{u,k}^{(t+1)}$ approximates the PME output in (17).

We emphasize that we are using this three-step procedure instead of the PME in (17) for three reasons: First, in Step (i) we calculate LLR values for every transmitted bit in each iteration, which provides our algorithm with soft-output data detection capabilities. Second, in Step (ii) we obtain probabilities for every transmitted bit in each iteration, which will be key in learning the parameters of our soft-output JED algorithm (see Section IV-B). Third, this procedure was shown in [14], [15] to be less complex and numerically more stable than evaluating the exact PME in (17).

IV. DEEP UNFOLDING WITH A HYPER-NETWORK

We now explain our deep unfolding strategy for the soft-output JED algorithm and how to train the algorithm parameters. Due to space constraints, we focus on QPSK only—the general case will be presented in [19].

A. Deep-Unfolding Architecture

In order to determine the algorithm parameters, we use an emerging paradigm known as deep unfolding [10]–[12] which we combine with a hyper-network that provides these parameters based on estimated CSI [13]. The idea of deep unfolding is to unfold an iterative algorithm into T_{\max} layers (one for every iteration) and use tools of deep learning to determine an optimal set of the algorithm's parameters in every iteration (layer) $t = 1, \dots, T_{\max}$. Instead of hard-coding these parameters after training, we train a hyper-network that generates these algorithm parameters dependent on CSI.

The hyper-network, the unfolded algorithm (which consists of T_{\max} layers, each representing a JED iteration), and the parameters are shown in Fig. 1. At layer t , the input $\mathbf{S}^{(t)}$ is first updated by a gradient descent step in (15) to obtain the symbol estimates $\mathbf{X}^{(t)}$. Then, the three-step procedure to approximate the PME as described in Section III-D is performed to obtain the next iterate $\mathbf{S}^{(t+1)}$ as in (21) and (22). We note that the last layer $t = T_{\max}$ only requires the LLR outputs in (19).

Our unfolded architecture requires several algorithm parameters, which are generated by a hyper-network. Specifically, for each iteration (layer) $t = 1, \dots, T_{\max}$, we require the per-iteration step size $\tau^{(t)}$ and the estimation error variances $\nu_{u,k}^{(t)}$.

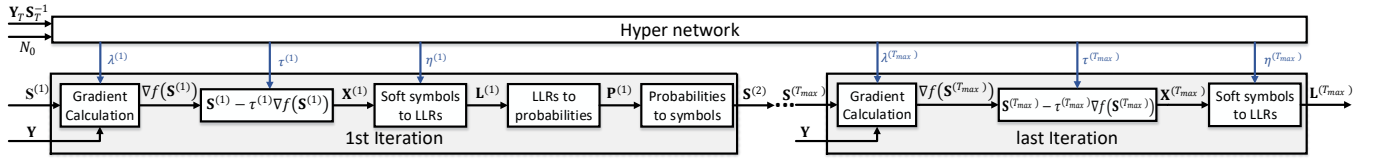


Fig. 1. Block diagram of the deep unfolded soft-output JED and the hyper-network. The unfolded algorithm consists of T_{\max} layers. Each layer takes in soft-symbols from the preceding layer and outputs new soft-symbols; the last layer outputs only LLR values. The hyper-network takes in the LS channel estimate $\hat{\mathbf{H}}^{\text{LS}}$ and noise variance N_0 in order to produce the parameters step sizes $\tau^{(t)}$, regularization parameters $\lambda^{(t)}$, and normalized error variances $\eta_u^{(t)}$.

for $u = 1, \dots, U$ and $k = 1, \dots, K$. To reduce the amount of parameters per iteration, we assume that the estimation error variances are fixed with respect to the time slot k , i.e., we only require $\nu_u^{(t)}$ and use the same variance for all time slots. We note that the hyper-network does not generate $\nu_u^{(t)}$, but rather a normalized version $\eta_u^{(t)} = N_0/\nu_u^{(t)}$ to account for large variations in N_0 . We also require the parameter λ in (2); instead of using the same parameter λ for all iterations $t = 1, \dots, T_{\max}$, each layer uses a different parameter $\lambda^{(t)}$.

The inputs to the hyper-network are the vectorized least-squared channel estimate $\hat{\mathbf{H}}^{\text{LS}} = \mathbf{Y}_T \mathbf{S}_T^{-1}$ of the pilot phase (\mathbf{Y}_T contains the first T columns of the matrix \mathbf{Y}) and the noise variance N_0 . The hyper-network itself consists of five dense layers with rectified linear unit (ReLU) activations in each layer except for the last one, which uses an absolute value activation to generate non-negative parameters.

B. Hyper-Network Training

In order to train the hyper-network, we leverage the soft-output capabilities of our algorithm. Specifically, since our JED algorithm computes probabilities for the transmitted bits (20), we can train the hyper-network using the outputs in the last iteration $t = T_{\max}$ using the widely-used binary cross entropy (BCE) loss, which is defined as follows:

$$H(b_i, p(b_i)) = b_i \log(p(b_i)) + (1 - b_i) \log(1 - p(b_i)). \quad (23)$$

Here, $b_i \in \{0, 1\}$ is the label of the i th bit and $p(b_i)$ is the predicted probability of this bit being 1. In our case, we utilize the probabilities $P_{b', u, k}^{(T_{\max})}$ in (20) for every transmitted bit $b_{b', u, k}$, where $b' = 1, 2$ is the bit index, $u = 1, \dots, U$ the UE index, and $k = 1, \dots, K$ the time slot index, calculated in the last iteration $t = T_{\max}$. Hence, we define the following average BCE loss over all of these probabilities

$$L = \frac{1}{2UK} \sum_{b'=1}^2 \sum_{u=1}^U \sum_{k=1}^K H(b_{b', u, k}, P_{b', u, k}^{(T_{\max})}), \quad (24)$$

which we use to train the hyper-network parameters. We learn only a single hyper-network for all signal-to-noise-ratio (SNR) values, which is in stark contrast to the common approach of using a different hyper-network for every SNR.

V. SIMULATION RESULTS

We now demonstrate the efficacy our soft-output JED algorithm and compare it to baseline algorithms. We first detail the system setup and then show simulation results.

A. System Setup

We simulate a MU-MIMO system as described in Section II-A with $B = 8$ BS antennas and $U = 4$ single-antenna UEs transmitting QPSK symbols for $K = 244$ time slots. The UEs transmit orthogonal pilots in \mathbf{S}_T from a 4×4 Hadamard matrix. The channel matrices are modelled as Rayleigh fading with i.i.d. complex standard Gaussian entries. We consider per-UE coding with a rate-1/2 low-density parity-check (LDPC) code as in IEEE 802.11n [22] with a block-length of 480 bits; for LDPC decoding, we use a sum-product and layered decoding algorithm with 10 iterations. The hyper-network is trained using an NVIDIA GTX1080 with 1M transmissions and batch size of 1k. We use Monte-Carlo simulations to extract the coded packet error rate (PER), uncoded bit error rate (BER), and BCE as in (24). We run $T_{\max} = 10$ iterations of our soft-output JED algorithm (called ‘‘S-JED’’).

B. Baseline Algorithms

In order to evaluate the effectiveness of our S-JED algorithm, we simulate the SIMO lower bound, which cancels MU interference with perfect CSI in a genie-aided fashion [23]. We also compare our algorithm to conventional methods that separate channel estimation from soft-output data detection. For such methods, we simulate a SIMO lower bound with estimated CSI (called ‘‘SIMO (est. CSI)’’), where we use a least-squares channel estimator to compute $\hat{\mathbf{H}}^{\text{LS}}$. We also compare S-JED to the widely used soft-output linear minimum mean-square error (L-MMSE) equalizer [21], [24] and the max-log optimal single-tree-search sphere decoder (STS-SD) [25].

C. Simulation Results

Figure 2 shows our simulation results. In Fig. 2(a), we see that S-JED approaches the SIMO lower bound by less than 3 dB at a coded PER of 0.1% and outperforms the SIMO lower bound that uses estimated CSI. S-JED significantly outperforms the max-log optimal soft-output STS-SD algorithm and the widely used L-MMSE equalizer, which both separate channel estimation from detection, by 2 dB and 4 dB, respectively. In Fig. 2(b), we see that the uncoded BER results behave similarly. The results in Fig. 2(c) demonstrate that the BCE accurately characterizes the performance of all methods, as the order between algorithms is preserved with respect to coded PER and uncoded BER—this implies that the BCE loss in (24) is well suited to train soft-output data detectors.

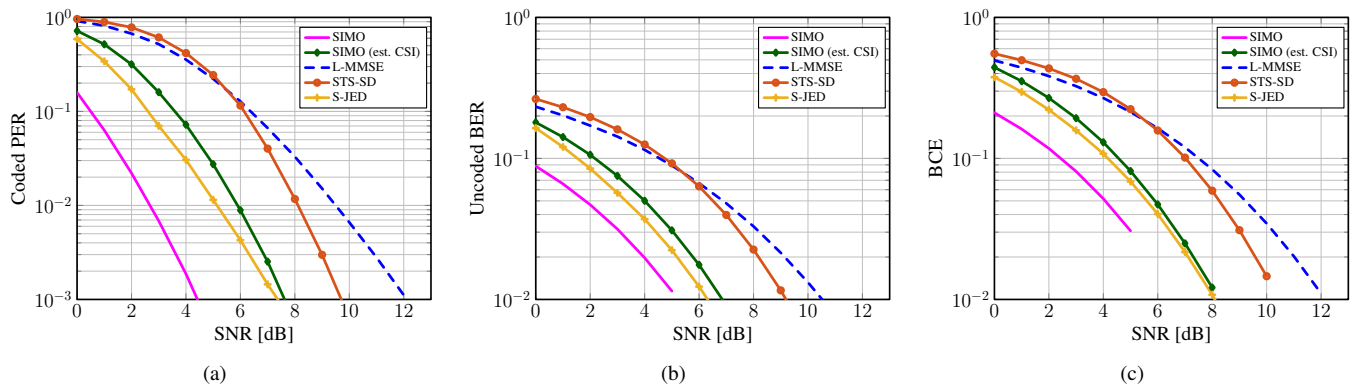


Fig. 2. Coded PER (a), uncoded BER (b), and BCE (c) performance for a $B = 8$ BS antenna, $U = 4$ UE MU-MIMO system with transmitting QPSK for $K = 240$ time slots. The proposed soft-output JED (S-JED) algorithm approaches the SIMO lower bound and outperforms the SIMO bound with estimated CSI as well as the max-log optimal soft-output STS-SD and the widely used L-MMSE equalizer which separate channel estimation from data detection.

VI. CONCLUSIONS

We have proposed a novel soft-output joint channel estimation and data detection (S-JED) algorithm for MU-MIMO systems. Our method formulates a maximum a-posteriori (MAP) optimization problem and computes approximate LLR values in every iteration. The algorithm parameters are generated by a hyper-network, which is trained using deep unfolding and a BCE cost function. Simulation results have shown that the proposed S-JED algorithm with only 10 iterations significantly outperforms the max-log optimal STS-SD and L-MMSE equalizer, which both separate channel estimation from soft-output data detection.

There are many avenues for future work. Our future journal paper in [19] will include missing derivations, a complexity comparison, and apply S-JED to higher-order modulation schemes as well as to other channel models.

REFERENCES

- [1] H. Vikalo, B. Hassibi, and P. Stoica, "Efficient joint maximum-likelihood channel estimation and signal detection," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1838–1845, Jul. 2006.
- [2] H. A. J. Alshamary, T. Al-Naffouri, A. Zaib, and W. Xu, "Optimal non-coherent data detection for massive SIMO wireless systems: A polynomial complexity solution," in *Proc. IEEE Signal Process. Signal Process. Edu. Workshop*, Aug. 2015, pp. 172–177.
- [3] W. Xu, M. Stojnic, and B. Hassibi, "On exact maximum-likelihood detection for non-coherent MIMO wireless systems: A branch-estimate-bound optimization framework," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2008, pp. 2017–2021.
- [4] O. Castañeda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Trans. Circuits Syst.*, vol. 63, no. 12, pp. 2334–2346, Dec. 2016.
- [5] O. Castañeda, T. Goldstein, and C. Studer, "VLSI designs for joint channel estimation and data detection in large SIMO wireless systems," *IEEE Trans. Circuits Syst. I*, vol. 65, no. 3, pp. 1120–1132, Mar. 2017.
- [6] H. Song, X. You, C. Zhang, O. Tirkkonen, and C. Studer, "Minimizing pilot overhead in cell-free massive MIMO systems via joint estimation and detection," in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.
- [7] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, Feb. 2020.
- [8] B. Yilmaz and A. Erdogan, "Channel estimation for massive MIMO: A semiblind algorithm exploiting QAM structure," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019.
- [9] E. Kofidis, C. Chatzichristos, and A. L. de Almeida, "Joint channel estimation/data detection in MIMO-FBMC/OQAM systems—a tensor-based approach," in *Proc. IEEE European Signal Process. Conf. (EUSIPCO)*, Sept. 2017, pp. 420–424.
- [10] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv:1409.2574*, Nov. 2014.
- [11] A. Balatsoukas-Stimming, O. Castañeda, S. Jacobsson, G. Durisi, and C. Studer, "Neural-network optimized 1-bit precoding for massive MU-MIMO," in *IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019.
- [12] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.
- [13] M. Goutay, F. Ait Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.
- [14] C. Jeon, A. Maleki, and C. Studer, "Mismatched data detection in massive MU-MIMO," *arXiv:2007.06491*, Jul. 2020.
- [15] C. Jeon, O. Castañeda, and C. Studer, "A 354 Mb/s 0.37 mm² 151 mW 32-User 256-QAM Near-MAP Soft-Input Soft-Output Massive MU-MIMO Data Detector in 28nm CMOS," *IEEE Solid-State Circuits Lett.*, vol. 2, no. 9, pp. 127–130, Sep. 2019.
- [16] D. Gesbert, M. Shafi, D.-S. Shiu, P. J. Smith, and A. Nagueb, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 281–302, Apr. 2003.
- [17] T. Goldstein, C. Studer, and R. G. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," Nov. 2014. [Online]. Available: <https://arxiv.org/abs/1411.3406>
- [18] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [19] H. Song, X. You, C. Zhang, and C. Studer, "Soft-output joint channel estimation and data detection for multiuser mimo using deep unfolding and a reweighted hyper-network," in *preparation for a journal*.
- [20] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of large MIMO detection via approximate message passing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1227–1231.
- [21] S. Fateh, C. Studer, and D. Seethaler, "VLSI implementation of soft-input soft-output MMSE parallel interference cancellation," 2009.
- [22] "IEEE standard for information technology part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Mar. 2012. [Online]. Available: <https://legal.vvv.enseirb-matmeca.fr/download/amichel>
- [23] J. Zhang, "Non-asymptotic capacity lower bound for non-coherent SIMO channels with memory," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, July 2006, pp. 1272–1276.
- [24] D. Seethaler, G. Matz, and F. Hlawatsch, "An efficient MMSE-based demodulator for MIMO bit-interleaved coded modulation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, vol. 4, Nov. 2004, pp. 2455–2459.
- [25] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.