LoRD-Net: Unfolded Deep Detection Network with Low-Resolution Receivers

Shahin Khobahi, Nir Shlezinger, Mojtaba Soltanalian, and Yonina C. Eldar

***Abstract*—The need to recover high-dimensional signals from their noisy low-resolution quantized measurements is widely encountered in communications and sensing. In this paper, we focus on the extreme case of one-bit quantizers, and propose a deep detector entitled LoRD-Net for recovering information symbols from one-bit measurements. Our method is a model-aware data-driven architecture based on deep unfolding of first-order optimization iterations. LoRD-Net has a task-based architecture dedicated to recovering the underlying signal of interest from the one-bit noisy measurements without requiring prior knowledge of the channel matrix through which the one-bit measurements are obtained. The proposed deep detector has much fewer parameters compared to black-box deep networks due to the incorporation of domain-knowledge in the design of its architecture, allowing it to operate in a data-driven fashion while benefiting from the flexibility, versatility, and reliability of model-based optimization methods. LoRD-Net operates in a blind fashion, which requires addressing both the non-linear nature of the data-acquisition system as well as identifying a proper optimization objective for signal recovery. Accordingly, we propose a two-stage training method for LoRD-Net, in which the first stage is dedicated to identifying the proper form of the optimization process to unfold, while the latter trains the resulting model in an end-to-end manner. We numerically evaluate the proposed receiver architecture for one-bit signal recovery in wireless communications and demonstrate that the proposed hybrid methodology outperforms both data-driven and model-based state-of-the-art methods, while utilizing small datasets, on the order of merely ~ 500 samples, for training.**

I. Introduction

Analog-to-digital conversion plays an important role in digital signal processing systems. While physical signals take values in continuous-time over continuous sets, they must be represented using a finite number of bits in order to be processed in digital hardware [2]. This operation is carried out using analog-to-digital converters (ADCs), which typically perform uniform sampling followed by a uniform quantization of the discrete-time samples. When using high-resolution ADCs, this conversion induces a minimal distortion, allowing to effectively process the signal using methods derived assuming access to the continuous-amplitude samples. However, the cost, power consumption and memory requirements of ADCs grow with the sampling rate and the number of bits assigned to each sample [3]. Consequently, recent years have witnessed an increasing interest in digital signal processing systems operating with low-resolution ADCs. Particularly, in multiple-input multiple-output (MIMO) communication receivers, which are required to simultaneously capture multiple analog signals with high bandwidth, there is a growing need to operate reliably with low-resolution ADCs [4]. The most coarse form of quantization is reduction of the signal to a single bit per sample, which may be accomplished via comparing the sample to some reference level, and recording whether the signal is above or below the reference. One-bit acquisition allows using high sampling rates at a low cost and low energy consumption. Due to such favorable properties of one-bit ADCs, they have been employed in a wide array of applications, including in wireless communications [1], [5], [6], radar signal processing [7]–[9], and sparse signal recovery [10], [11].

The non-linear nature of low-resolution quantization makes symbol detection a challenging task. This situation is significantly exacerbated in practical one-bit communication and sensing where the channel is to be estimated in conjunction with symbol detection. A *coherent* symbol detection task is concerned with recovering the underlying signal of interest from the one-bit measurements assuming the channel state information (CSI) is known at the receiver. On the other hand, the more difficult task of *blind* symbol detection, which is the focus here, carries out recovery of the underlying transmitted symbols when CSI is not available.

Two main strategies have been proposed in the literature to facilitate operation with low-resolution ADCs: The first designs the overall acquisition system in light of the task for which the signals are acquired. For instance, MIMO communication receivers acquire their channel output in order to extract some underlying information, e.g., symbol detection. As the analog signals are not required to be recovered from their digital representation, one can design the acquisition system to reliably infer the desired information while operating with low resolution ADCs [12]–[16]. Such task-based quantization systems rely on pre-quantization processing, which requires dedicated hardware in the form of hybrid receiver architectures [17], [18] or unique antenna structures [19], [20], which are configured along with the quantization rule.

An alternative approach to task-based quantization, which does not require additional configurable analog hardware and is the focus of the current work, is to recover the desired information from the distorted coarsely discretized representation of the signal in the digital domain. The main benefit of schemes carried out only in the digital domain is their simplicity of implementation, as they do not require to introduce modifications to the quantization system and circumvent the need for adding pre-quantization analog processing hardware. In the context of MIMO systems, various methods

Parts of this work was presented in the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, United Kingdom, 2019 [1]. This paper has received supports from the Benoziyo Endowment Fund for the Advancement of Science, the Estate of Olga Klein – Astrachan, the European Union's Horizon 2020 research and innovation program under grant No. 646804-ERC-COG-BNYQ, from the Israel Science Foundation under grant No. 0100101, from the U.S. National Science Foundation under grant No. CCF-1704401, and from an Illinois Discovery Partners Institute (DPI) Seed Award. S. Khobahi and M. Soltanalian are with the ECE Dept., University of Illinois at Chicago, Chicago, IL (e-mail: {skhoba2, msol}@uic.edu). N. Shlezinger is with the School of ECE, Ben-Gurion University of the Negev, Be'er-Sheva, Israel (e-mail: nirshl@bgu.ac.il). Y. C. Eldar is with the Faculty of Math and CS, Weizmann Institute of Science, Rehovot, Israel (e-mail: yonina.eldar@weizmann.ac.il).

have been proposed in the literature for channel estimation and signal decoding from quantized outputs, including model-based signal processing methods as surveyed in [21], as well as model-agnostic systems based on machine learning and data-driven techniques [22]–[29].

Most existing model-based detection algorithms require coherent operation, i.e., they rely on prior knowledge of the CSI and other system parameters. Among these works are the near-Maximum Likelihood (nML) detector proposed for one-bit MIMO receivers in [30], the linear receivers studied in [31], [32], and the message passing based detectors considered in [33], [34]. The fact that such approaches require accurate CSI led to several works specifically dedicated to CSI estimation in the presence of low-resolution ADCs. These include [30], [35], which studied maximum-likelihood estimation for recovering the CSI in the presence of one-bit data, the works in [36], [37], which developed linear estimators for CSI estimation purposes in one-bit MIMO systems, and [38] which focuses on sparse channels and utilizes one-bit sparse recovery methods for CSI estimation. However, all these strategies inevitably induce non-negligible CSI estimation error, which may notably degrade the accuracy in signal detection based on the estimated CSI.

Over the past several years, data-driven methods, and specifically deep neural networks (DNNs), have attracted unprecedented attention from research communities across the board. The advent of low-cost specialized powerful computing resources and the continually increasing amount of massive data generated by the human population and machines, along with new optimization and learning methods, have paved the way for DNNs and machine learning-based models to prove their effectiveness in many engineering areas, such as computer vision and natural language processing [39]. DNNs learn their mapping from data in a model-agnostic manner, and can thus facilitate non-coherent (blind) detection.

Previously proposed DNN-aided symbol detection techniques for communication receivers can be divided based on their receiver architectures; namely, those that utilize conventional machine learning architectures for detection, including [40]–[42], and schemes combining DNNs with model-based detection methods, such as the blind DNN-aided receivers proposed in [43]–[46] and the coherent detectors of [47], [48], see also surveys in [49], [50]. In the context of one-bit DNN-aided receivers, previous works to date focus mainly on the first approach, i.e., applying conventional DNNs for the overall detection task. Among these works are [22], [25] and [23], which applied generic DNNs for channel estimation in one-bit MIMO receivers. The application of conventional architectures for symbol detection was studied in [24], [27] and [28], while [26] showed that autoencoders can facilitate the design of error correction codes for communications with one-bit receivers. Recently, the authors in [29] considered the problem of symbol detection for a one-bit massive MIMO system and proposed a linear estimator module based on the Bussgang decomposition technique combined with a model-driven neural network.

The vast majority of the aforementioned works on learning-aided one-bit receivers rely on conventional DNN architectures. Such DNNs require a massive amount of training samples and must be trained on data from the same (or a similar) statistical model as the one under which they are required to operate, imposing a major challenge in dynamic wireless communications. In fact, the use of generic black-box DNNs is mostly justified in applications where a satisfactory description of the underlying governing dynamics of the system is not achievable, as is the case in computer vision and natural language processing fields. As surveyed above, this is not the case in the field of one-bit MIMO systems. This gives rise to the need that is bridging the gap between data-driven and model-based approaches in this context, and moving towards specialized deep learning models for signal processing techniques in one-bit MIMO systems—which is the aim of this work.

In this paper, we develop a hybrid model-based and data-driven system which learns to carry out blind symbol detection from one-bit measurements. The proposed architecture, referred to as LoRD-Net (Low Resolution Detection Network), combines the well-established model-based maximum-likelihood estimator (MLE) with machine learning tools through the deep unfolding method [51]–[56] for designing DNNs based on model-based optimization algorithms. To derive LoRD-Net, we first formulate the MLE for the task of symbol detection from one-bit samples. Next, we resort to first-order gradient-based methods for the MLE computation, and unfold the iterations onto layers of a DNN. The resulting LoRD-Net learns to carry out MLE-approaching symbol detection without requiring CSI.

Applying conventional gradient-based optimization methods requires knowledge of the underlying system parameters, i.e., full CSI. Hence, a typical approach to unfold such a symbol detection algorithm would be to estimate the unknown parameters from training, and substitute it into the unfolded network [46]. We show that instead of estimating the unknown system parameters, it is preferable to learn an *alternative channel* which allows the receiver to detect the symbols reliably. Surprisingly, we demonstrate that the alternative channel learned by LoRD-Net is in general not the true channel. Based on this observation, we propose a two-stage training procedure, comprised of learning the proper optimization process to unfold, followed by an end-to-end training of the unfolded DNN.

The proposed LoRD-Net has thus the following properties:

i) Compared to the vanilla MLE symbol detector, our model does not need to estimate the channel separately.

ii) Owing to its hybrid nature, it has low computational cost in operation and is highly scalable, facilitating much faster inference as compared to its black-box data-driven and model-based counterparts.

iii) The proposed deep architecture is interpretable and has far fewer parameters compared to existing black-box deep learning solutions. This follows from the incorporation of domain-knowledge in the design of the network architecture (i.e., being model-based), allowing LoRD-Net to train with much fewer labeled samples as compared to existing data-driven one-bit receivers.

We verify the above characteristics of LoRD-Net in an experimental study, where we show that training of the proposed LoRD-Net architecture can be performed with far fewer samples as compared to its data-driven counterparts, and demonstrate substantially superior performance compared to existing model-based and data-driven algorithms for symbol detection in massive MIMO channels with one-bit ADCs.

The rest of the paper is organized as follows. In Section II, we present the considered system model and the corresponding MLE formulation. In Section III, we derive LoRD-Net by unfolding the first-order gradient iterations associated with the MLE computation, and present its two-stage training procedure. Section IV provides a detailed numerical analysis of LoRD-Net applied to MIMO communications. Finally, Section V concludes the paper.

Throughout the paper, we use the following notation. Bold lowercase and bold uppercase letters denote vectors and matrices, respectively. We use $(\cdot)^T$, $\mathrm{Diag}(\cdot)$, and $\mathrm{sign}(\cdot)$, and $\log\{\cdot\}$ to denote the transpose operator, the diagonal matrix formed by the entries of the vector argument, the sign operator, and the natural logarithm, respectively. The symbol $\odot$ represents the Hadamard product, while $\mathbf{1}$ and $\mathbf{0}$ are the all-one and all-zero vectors/matrices. The i-th entry of the vector $\boldsymbol{x}$ is x_i, and $\|\boldsymbol{x}\|_p$ is the ℓ_p-norm of $\boldsymbol{x}$; $\mathcal{M}^n$ is the n-ary Cartesian product of a set $\mathcal{M}$, and $\mathbb{S}_+$ denotes the cone of symmetric positive definite matrices.

II. System Model and Preliminaries

In this section, we discuss the considered system model. We focus on one-bit data acquisition and blind signal recovery. We then formulate the MLE for this problem, which is used in designing the LoRD-Net architecture in Section III.

A. Problem Formulation

We consider a low-resolution data-acquisition system which utilizes m one-bit ADCs. By letting $\boldsymbol{y} \in \mathbb{R}^m$ denote the received signal, the discrete output of the ADCs can be written as $\boldsymbol{r} = \mathrm{sign}\left(\boldsymbol{y} - \boldsymbol{b}\right)$, where $\boldsymbol{b} \in \mathbb{R}^m$ denotes the vector of quantization thresholds, and $\mathrm{sign}(\cdot)$ is the sign function, i.e., $\mathrm{sign}(x) = +1$ if $x \geq 0$ and $\mathrm{sign}(x) = -1$ otherwise. The received vector $\boldsymbol{y}$ is statistically related to the unknown vector of interest $\boldsymbol{x} \in \mathcal{M}^n \subseteq \mathbb{R}^n$ according to the following linear relationship:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C})$ denotes additive Gaussian noise with a covariance matrix of the form $\boldsymbol{C} = \mathrm{Diag}(\sigma_0^2, \sigma_1^2, \ldots, \sigma_{m-1}^2)$ with diagonal entries $\{\sigma_i^2\}_{i=0}^{m-1}$ representing the noise variance at each respective dimension, and $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ is the channel matrix. We assume that the elements of the unknown vector $\boldsymbol{x}$ are chosen independently from a finite alphabet $\mathcal{M} = \{s_1, s_2, \cdots, s_{|\mathcal{M}|}\}$. This setup represents low-resolution receivers in uplink multi-user MIMO systems, where $\boldsymbol{x}$ is the symbols transmitted by the users, and $\boldsymbol{y}$ is the corresponding channel output, as illustrated in Fig. 1.

The overall dynamics of the system are thus compactly expressed as:

$$\boldsymbol{r} = \mathrm{sign}(\boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} - \boldsymbol{b}). \tag{2}$$

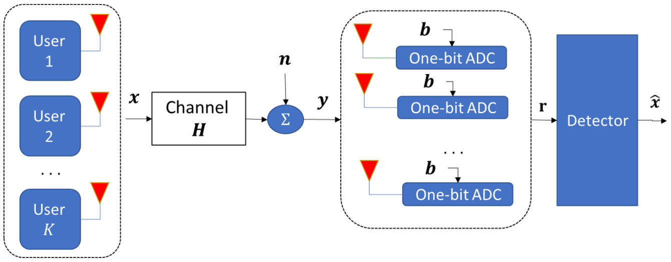


Fig. 1. System model illustration.

In the sequel, we refer to $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ as the system parameters. Note that the above system model can be modified using conventional transformations to accommodate a complex-valued system model.

Our main goal is to perform the task of symbol detection, i.e., recover $\boldsymbol{x}$, from the collected one-bit measurements $\boldsymbol{r}$. We focus on blind (non-coherent) recovery, namely, the system parameters $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$, i.e., the channel matrix and the covariance of the noise, are not available to the receiver. Nonetheless, the receiver has access to a limited set of B labeled samples $\{\boldsymbol{x}_p^b, \boldsymbol{r}_p^b\}_{b=0}^{B-1}$, representing, e.g., pilot transmissions. The quantization thresholds of the ADCs, i.e., the vector $\boldsymbol{b}$, are assumed to be fixed and known. While we do not consider the selection of $\boldsymbol{b}$ in the following, we discuss in the sequel how its optimization can be incorporated into the detection method.

B. Maximum Likelihood Recovery

To understand the challenges associated with blind low-resolution detection, we next discuss the MLE for recovering $\boldsymbol{x}$ from $\boldsymbol{r}$. In particular, the intuitive model-based approach is to utilize the labeled data to estimate the system parameters Θ, and then to use this estimation to compute the coherent (non-blind) MLE. Therefore, to highlight the limitations of this strategy, we assume here that the system parameters $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ are fully known at the receiver. Let

$$\begin{aligned}\mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}) &\triangleq \log \Pr(\boldsymbol{r}|\boldsymbol{x}, \Theta)\\ &\overset{(a)}{=} -\sum_{i=0}^{m-1} \log\left\{Q\left(\frac{r_i}{\sigma_i}\left(b_i - \boldsymbol{h}_i^T\boldsymbol{x}\right)\right)\right\},\end{aligned} \tag{3}$$

represent the log-likelihood objective for a given vector of one-bit observations $\boldsymbol{r}$, where (a) is proven in [1], and $\boldsymbol{h}_i$ represent the i-th column of the matrix $\boldsymbol{H}^T$. The coherent MLE is then given by

$$\hat{\boldsymbol{x}}_{\mathrm{ML}}(\boldsymbol{r}) = \underset{\boldsymbol{x}\in\mathcal{M}^n}{\mathrm{argmax}}\ \mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}). \tag{4}$$

Although the MLE in (4) has full accurate knowledge of the parameters Θ, its computation is still challenging. The main difficulty emanates from solving the underlying optimization problem in the discrete domain, implying that the MLE requires an exhaustive search over the discrete domain $\mathcal{M}^n$, whose computational complexity grows exponentially with n. A common strategy to tackle the discrete optimization problem in (4) is to relax the search space to be continuous. This results in the following relaxed unconstrained MLE rule:

$$\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}) = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathrm{argmax}}\ \mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}). \tag{5}$$

The optimization problem in (5) is convex due to the log-concavity of $Q(\cdot)$, and thus can be solved using first-order gradient optimization. In particular, the negative log-likelihood function with respect to the unknown vector $\boldsymbol{x}$ can be compactly expressed as [1]:

$$\nabla_{\boldsymbol{x}}\mathcal{F}_{\Theta}(\boldsymbol{x};\boldsymbol{r}) = \boldsymbol{H}^{T}\tilde{\boldsymbol{R}}\,\boldsymbol{\eta}\left(\tilde{\boldsymbol{R}}\left(\boldsymbol{b}-\boldsymbol{H}\boldsymbol{x}\right)\right), \tag{6}$$

where $\boldsymbol{\eta}$ is a non-linear function defined as $\boldsymbol{\eta}(\boldsymbol{x}) \triangleq Q^{'}(\boldsymbol{x}) \oslash Q(\boldsymbol{x})$, in which the operator $\oslash$ denotes the element-wise division operation, $Q^{'}(x)$ is the derivative of $Q(x)$, that is given by the negative probability density function of a standard Normal distribution, and $\tilde{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{C}^{-\frac{1}{2}}$ is the semi-whitened version of the *one-bit matrix* $\boldsymbol{R} = \mathrm{Diag}\left(r_0,\ldots,r_{m-1}\right)$.

As $\bar{\boldsymbol{x}}_{\Theta}(\boldsymbol{r})$ obtained via (5) is not guaranteed to take values in $\mathcal{M}^n$, the final estimate of the symbols is obtained by applying a projection operator $\mathcal{P}_{\mathcal{M}^n} : \mathbb{R}^n \mapsto \mathcal{M}^n$ to $\bar{\boldsymbol{x}}(\boldsymbol{r})$. This operator maps the continuous input vector onto its closest lattice point on the discrete set $\mathcal{M}^n$, i.e.,

$$\mathcal{P}_{\mathcal{M}^n}(\boldsymbol{x}) = \underset{\boldsymbol{z}\in\mathcal{M}^n}{\mathrm{argmin}}\|\boldsymbol{z}-\boldsymbol{x}\|_2^2. \tag{7}$$

Tackling a discrete program via continuous relaxation, as done in (5), is subject to an inherent drawback. As a case in point, one can only expect $\bar{\boldsymbol{x}}_{\Theta}(\boldsymbol{r})$ to provide an accurate approximation of the true MLE if the real-valued vector $\bar{\boldsymbol{x}}_{\Theta}(\boldsymbol{r})$ is very close to the discrete valued MLE $\hat{\boldsymbol{x}}_{\mathrm{ML}}(\boldsymbol{r})$. In such a case, the MLE is obtained by projecting into the lattice points in $\mathcal{M}^n$. However, this is not the case in many scenarios, and specifically, when the noise variance in each respective dimension is high. In other words, it is not necessarily the case that the minimizer of the objective function on the continuous domain (5) is close to the MLE, which takes values in the discrete set $\mathcal{M}^n$. Note that utilizing the true system parameters will only lead to optimal estimates when considering the original discrete problem (4). In fact, one can no longer necessarily argue that the true system parameters are optimal choices for Θ in the relaxed MLE. This insight, which is obtained from the computation of the coherent MLE, is used in our derivation of the blind unfolded detector in the following section.

III. Proposed Methodology

In this section, we present the proposed **Low Resolution Detection Network**, abbreviated as **LoRD-Net**. We begin with a high-level description of LoRD-Net in Subsection III-A. Then, we present the unfolded architecture in Subsection III-B and discuss the training procedure in Subsection III-C. Finally, we provide a discussion in Subsection III-D.

A. High-Level Description

As noted in the previous section, the intuitive approach to blind symbol detection is to utilize the labeled data $\{\boldsymbol{x}_p^b, \boldsymbol{r}_p^b\}_{b=0}^{B-1}$ to estimate the true system model Θ, and then to recover the symbol vector $\boldsymbol{x}$ from $\boldsymbol{r}$ using the MLE. Nonetheless, the coherent MLE (4) is computationally prohibitive, while its relaxed version in (5) may be inaccurate. Alternatively, one can seek a purely data-driven strategy, using the data to train a black-box highly-parameterized DNN for detection, requiring a massive amount of labeled samples. Consequently, to facilitate accurate detection at affordable complexity and with limited data, we design LoRD-Net via model-based deep learning [57], by combining the learning of a *competitive objective*, combined with *deep unfolding* of the relaxed MLE.

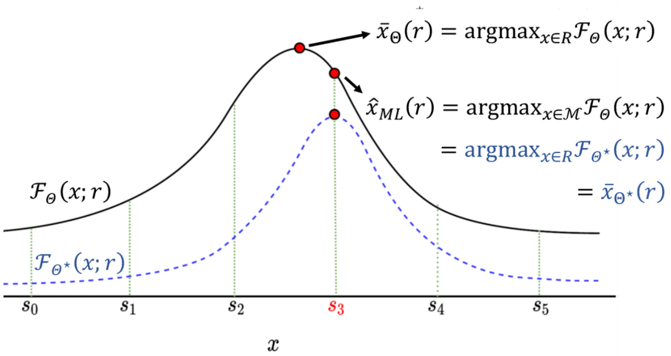


Fig. 2. An illustration of the relation between the optimal point of a competitive objective function (dashed blue line) and the true MLE $\hat{\boldsymbol{x}}_{\mathrm{ML}}$ obtained by an exact maximization of the log-likelihood objective function (solid black line) over the discrete set $\mathcal{M}$ as well as an approximation of the MLE $\bar{\boldsymbol{x}}_{\Theta}$ obtained by a maximization of the log-likelihood objective function over the continuous space $\mathbb{R}$, when the true transmitted symbol is $s_3 \in \mathcal{M}$.

Learning a competitive objective refers to the setting of the unknown system parameters Θ. However, the goal here is not to estimate the *true system parameters*, but rather the ones for which the solution to the relaxed MLE coincides with the true value of $\boldsymbol{x}$. This system identification problem can be written as

$$\mathcal{F}_{\Theta^\star}(\boldsymbol{r};\boldsymbol{x}) = \min_{\Theta}\ \frac{1}{B}\sum_{b=0}^{B-1}\left\|\bar{\boldsymbol{x}}_{\Theta}(\boldsymbol{r}_p^b) - \boldsymbol{x}_p^b\right\|_2^2, \tag{8}$$

where $\bar{\boldsymbol{x}}_{\Theta}$ is the relaxed MLE (5). The optimization problem (8) yields a surrogate objective function $\mathcal{F}_{\Theta^\star}$, or equivalently, a set of system parameters $\Theta^\star$, referred to as a *competitive objective* to the true $\mathcal{F}_{\Theta}$. An illustration of such a competitive objective obtained for the case of $n=1$ is depicted in Fig. 2.

The main difficulty in solving (8) stems from the fact that $\bar{\boldsymbol{x}}_{\Theta}(\boldsymbol{r}) = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathrm{argmax}}\ \mathcal{F}_{\Theta}(\boldsymbol{x};\boldsymbol{r})$ is not differentiable with respect to the system parameters Θ. We overcome this obstacle by applying a differentiable approximation of $\bar{\boldsymbol{x}}(\boldsymbol{r})$, or equivalently, an algorithm that approximates the argmax operator specific to our problem. Since $\bar{\boldsymbol{x}}_{\Theta}(\boldsymbol{r})$ can be computed by first-order gradient methods, we design a deep unfolded network [52] to compute the relaxed MLE in manner which is differentiable with respect to Θ. The usage of deep unfolding allows not only to learn a competitive objective via (8), but also results in accurate inference with a reduced number of iterations compared to model-based first-order gradient optimization. Furthermore, the unfolded network utilizes a relatively small amount of trainable parameters, thus enabling learning from small amounts of labeled samples.

B. LoRD-Net Architecture

We now present the architecture of LoRD-Net, which maps the low resolution $\boldsymbol{r}$ into an estimated $\hat{\boldsymbol{x}}$. For given system parameters Θ whose learning is detailed in Subsection III-C based on the competitive objective rationale described above, LoRD-Net is obtained by unfolding the iterations of a first-order optimization of the relaxed MLE (5). Our derivation thus begins by formulating the first-order methods to iteratively solve (5) for a given Θ.

Let $g_{\phi_i} : \mathbb{R}^n \mapsto \mathbb{R}$ be a parametrized operator defined as $g_{\phi_i}(\boldsymbol{x}; \Theta, \boldsymbol{r}) = \boldsymbol{x} - \boldsymbol{G}_i \nabla_{\boldsymbol{x}} \mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r})$, where $\boldsymbol{G}_i \in \mathbb{R}^{n \times n}$ is a positive-definite weight matrix and $\phi_i = \{\boldsymbol{G}_i\}$ denotes the set of parameters of the operator g_{ϕ_i}. Such a linear operator can be used to model a first-order optimization solver by considering a composition of t mappings of the form:

$$\begin{aligned} \boldsymbol{x}_{t+1} = \mathcal{G}^t_{\boldsymbol{\phi}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}) &= \boldsymbol{x}_t - \boldsymbol{G}_t \nabla_{\boldsymbol{x}} \mathcal{F}_\Theta(\boldsymbol{x}_t; \Theta, \boldsymbol{r}), \\ &\triangleq g_{\phi_t} \circ \cdots \circ g_{\phi_1} \circ g_{\phi_0}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}) \end{aligned} \tag{9}$$

where $\boldsymbol{x}_0$ is an initial point, $\boldsymbol{\phi} = \{\phi_0, \cdots, \phi_{t-1}\}$ is the set of parameters of the overall mapping $\mathcal{G}^t_{\boldsymbol{\phi}}$. The mapping (9) is differentiable with respect to the system parameters Θ, and its local weights $\boldsymbol{\phi}$. For a fixed number of iterations L, the resulting function $\mathcal{G}^L_{\boldsymbol{\phi}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r})$ is thus differentiable with respect to the set of parameters $\{\boldsymbol{\phi}, \Theta\}$ and its input (unlike the original argmax operator). Therefore, it can now be used as a differentiable approximation of $\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r})$, which allows for a training (optimization) over the set of its parameters based on the gradient-based training algorithms and the back-propagation technique.

Following the deep unfolding framework [52], the function $\mathcal{G}^L_{\boldsymbol{\phi}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r})$ can be implemented as a L-layer feed-forward neural network, where the initial point $\boldsymbol{x}_0$ and the one-bit samples $\boldsymbol{r}$ constitute the input to the network, and with trainable parameters that are given by $\{\Theta, \boldsymbol{\phi}\}$. By (6), the i-th layer computes:

$$g_{\phi_i}(\boldsymbol{x}_i; \Theta, \boldsymbol{r}) = \boldsymbol{x}_i - \boldsymbol{G}_i \boldsymbol{z}_i, \text{ with} \tag{10}$$

$$\boldsymbol{z}_i = \boldsymbol{H}^T \tilde{\boldsymbol{R}} \, \boldsymbol{\eta} \left(\tilde{\boldsymbol{R}} (\boldsymbol{b} - \boldsymbol{H} \boldsymbol{x}_i) \right), \tag{11}$$

where the overall dynamics of the LoRD-Net is given by:

$$\mathcal{G}^L_{\boldsymbol{\phi}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}) = g_{\phi_{L-1}} \circ g_{\phi_{L-2}} \circ \cdots \circ g_{\phi_0}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}). \tag{12}$$

Each vector $\boldsymbol{x}_i$ in (10) represents the input to the i-th layer (or equivalently, the output of the previous iteration), with $\boldsymbol{x}_0$ being the input of the entire network (which represents the initial point for the optimization task). Upon the arrival of any new one-bit measurement $\boldsymbol{r}$, the recovered symbols $\hat{\boldsymbol{x}}$ are obtained by feed-forwarding $\boldsymbol{r}$ through the L layers of LoRD-Net. In order to obtain discrete samples, the output of LoRD-Net is projected into the feasible discrete set $\mathcal{M}^n$ via

$$\hat{\boldsymbol{x}} = \mathcal{P}_{\mathcal{M}^n} \left(\mathcal{G}^L_{\boldsymbol{\phi}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}) \right). \tag{13}$$

An illustration of LoRD-Net is depicted in Fig. 3.

We note that one can also propose an alternative architecture, derived by applying the projection operator $\mathcal{P}_{\mathcal{M}^n}$ at the output of each layer, i.e., by defining $g_{\phi_i}(\boldsymbol{x}_i; \theta, \boldsymbol{r}) = \mathcal{P}_{\mathcal{M}^n}(\boldsymbol{x}_i - \boldsymbol{G}_i \boldsymbol{z}_i)$. Such a setting corresponds to the unfolding of a projected gradient descent method. However, our numerical investigations have consistently shown that such an architecture suffers from the vanishing gradient problem during training and a significant degradation in performance. As a result, we implement LoRD-Net while applying the projection operator once on the output of the network, and only during inference, as discussed above.

In principle, one can fix $\boldsymbol{G}_i = \delta \boldsymbol{I}$ for some $\delta > 0$, for which (12) represents L steps of gradient descent with step size δ. In the unfolded implementation, the weights $\{\boldsymbol{G}_i\}$ are tuned from data, allowing to detect with less iterations, i.e., layers. As a result, once LoRD-Net is trained, i.e., its weight matrices $\boldsymbol{\phi} = \{\boldsymbol{G}_i\}$ and the unknown system parameters Θ are learned from data, it is capable of carrying out fast inference, owing to its hybrid model-based/data-driven structure. Furthermore, the number of iterations L is optimized to boost fast inference in the training procedure, as detailed in the following.

C. Training Procedure

Herein, we present the training procedure for LoRD-Net. In particular, our main goal is to perform inference of the unknown system parameters Θ based on the rationale detailed in Subsection III-A, i.e., to obtain a competitive objective. The learning competitive objective is used to tune the weights of the unfolded network $\boldsymbol{\phi}$. Accordingly, we present a two-stage training procedure for LoRD-Net (12). Once the training of the LoRD-Net is completed, it carries out symbol detection from one-bit information without requiring the knowledge of system parameters Θ.

1) Training Stage 1 - Learning a Competitive Objective: The first stage corresponds to learning the unknown system parameter Θ. However, as formulated in (8), we do not seek to estimate the true values of the channel matrix $\boldsymbol{H}$ and noise covariance $\boldsymbol{C}$, but rather learn the surrogate values which will facilitate accurate detection using the relaxed MLE formulation. We do this by taking advantage of two propitieties of LoRD-Net: The first is the differentiability of the unfolded architecture with respect to $\boldsymbol{\Theta}$, which facilitates gradient-based optimization. The second is the fact that for $\boldsymbol{G}_i = \delta \boldsymbol{I}$, LoRD-Net essentially implements L steps of gradient descent with step size δ over the convex objective (5), and is thus expected to reach its maxima.

Based on the above properties, we fix a relatively large number of layers/iterations L for this training stage, and fix the weights $\boldsymbol{\phi}$ to $\boldsymbol{G}_i = \delta \boldsymbol{I}$. Under this setting, the output of LoRD-Net $\mathcal{G}^L_{\boldsymbol{\phi}=\{\delta \boldsymbol{I}\}}(\boldsymbol{x}; \Theta, \boldsymbol{r})$ represents an approximation of the relaxed MLE for a given parameter Θ, denoted $\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r})$, i.e., we have that

$$\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}) \approx \mathcal{G}^L_{\boldsymbol{\phi}=\{\delta \boldsymbol{I}\}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}). \tag{14}$$

We refer to the setting $\boldsymbol{\phi} = \{\delta \boldsymbol{I}\}$ using in this stage as the *basic* optimization policy. Note that as the number of layers grows large, the above approximation becomes more accurate. Hence, by substituting (14) into (8) and replacing $\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}^i_p)$ with the corresponding outputs of LoRD-Net, we formulate the loss measure of the first training stage of LoRD-Net as:

$$\Theta^\star = \underset{\Theta}{\text{argmin}} \; \frac{1}{B} \sum_{i=0}^{B-1} \left\| \mathcal{G}^L_{\boldsymbol{\phi}=\{\delta \boldsymbol{I}\}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}^i_p) - \boldsymbol{x}^i_p \right\|_2^2. \tag{15}$$

Owing to the differentiable nature of $\mathcal{G}^L_{\boldsymbol{\phi}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r})$ with respect to Θ, we recover $\Theta^\star$ based on (15) using conventional gradient-based training, e.g., stochastic gradient descent with

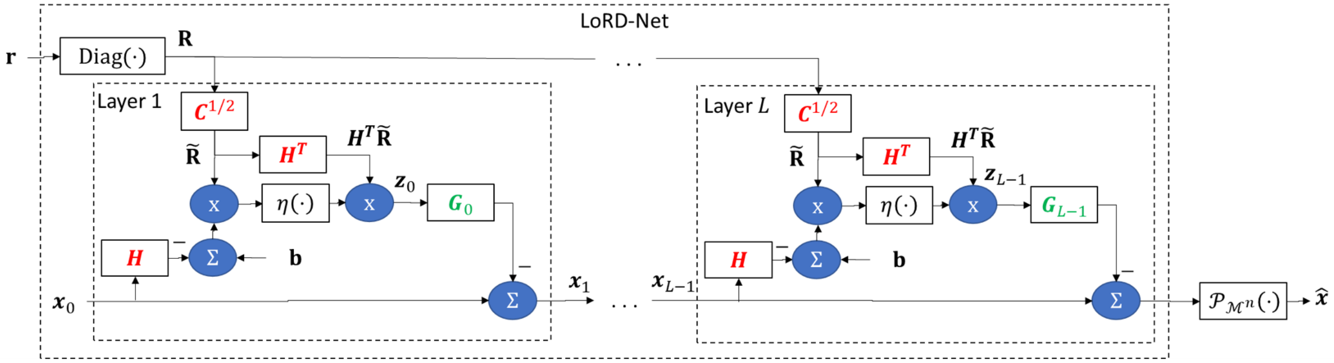


Fig. 3. An illustration of LoRD-Net, where trainable system parameters and unfolded weights are highlighted in red and green colors, respectively.

backpropagation, as detailed in our numerical evaluations description in Section IV

2) Training Stage 2 - Learning the Unfolded Weights: Having learned the unknown system parameters Θ in Stage 1, we turn to tuning the parameters of LoRD-Net, i.e., the set $\boldsymbol{\phi} = \{\boldsymbol{G}_i\}$. We note that in Stage 1, the rationale was to use the basic optimization policy $\boldsymbol{\phi} = \{\boldsymbol{G}_i = \delta\boldsymbol{I}\}_{i=0}^{L-1}$ with a large number of layers L, exploiting the insight that under this setting, LoRD-Net effectively implements conventional gradient descent. However, once Stage 1 is concluded and $\Theta^\star$ is learned, it is preferable to reduce the number of layers L compared to that used in Stage 1, thus exploiting the ability of the unfolded network to carry out faster inference compared to their model-based iterative counterparts by learning the weights applied in each iteration [52], [58]. Consequently, the first step in this stage is to set a number of layers to a value which can potentially be smaller than that used in the first training stage, and then optimize the weights according to the following criterion:

$$\boldsymbol{\phi}^\star = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \frac{1}{B}\sum_{i=0}^{B-1}\left\|\mathcal{G}^L_{\boldsymbol{\phi}=\{\boldsymbol{G}_l\}_{l=1}^L}(\boldsymbol{x}_0;\Theta^\star,\boldsymbol{r}_p^i)-\boldsymbol{x}_p^i\right\|_2^2. \quad (16)$$

Generally speaking, in order for a first-order optimizer (LoRD-Net in this case) to provide a descent direction at each iteration (layer), the pre-conditioning matrices must be positive-semidefinite so that each iteration does not reverse the gradient direction. To incorporate this requirement into LoRD-Net training, we re-parameterize the pre-conditioning matrices by writing $\{\boldsymbol{G}_i = \boldsymbol{W}_i\boldsymbol{W}_i^T\}$ and performing the traning over the matrices $\{\boldsymbol{W}_i\}$. The resulting two-stage training algorithm is summarized as Algorithm 1.

```
Algorithm 1: Training LoRD-Net
  Input: Labeled data $\{\boldsymbol{x}_p^b, \boldsymbol{r}_p^b\}_{b=0}^B$
  Stage 1 Init: Fix (large) $L$, step-size $\delta \in (0,1)$, and
    weights $\boldsymbol{G}_l = \delta\boldsymbol{I}$. Initialize $\boldsymbol{x}_0$ ;
  Optimize $\Theta^\star$ via (15) ;                    // Stage 1
  Stage 2 Init: Fix (small) $L$. Initialize $\boldsymbol{x}_0$ ;
  Set the trainable parameters to $\{\boldsymbol{G}_i = \boldsymbol{W}_i\boldsymbol{W}_i^T\}$ ;
  Optimize $\boldsymbol{\phi}^\star$ according to (16) ;          // Stage 2
  Output: LoRD-Net parameters $\{\Theta^\star, \boldsymbol{\phi}^\star\}$
```

When the network is properly trained, LoRD-Net is expected to carry out learned and accelerated first-order optimization, tuned to operate even in channel conditions for which such an approach does not yield the MLE for the true channel.

D. Discussion

LoRD-Net is a data-driven acquisition system based on unfolding first-order gradient optimization methods, designed for low-resolution MIMO receivers operating without analog processing. Its model-awareness enables the receiver to learn to accurately infer from smaller training sets compared to conventional DNN architectures applied to such setups, as suggested, e.g., in [24], giving rise to the possibility of tracking block-fading channel conditions via online training, as in [43]. Furthermore, LoRD-Net differs from previously proposed deep unfolded MIMO receivers as surveyed in [49] in two key aspects: First, LoRD-Net is particularly designed for one-bit observations, being derived from the iterative optimization formulation which arises from such setups. Second, previous unfolded MIMO receivers either assumed prior knowledge of the channel parameters, as in [47], or alternatively, utilize external modules to directly estimate the CSI as in [46]. LoRD-Net exploits the fact that, for its unfolded relaxed convex optimization algorithm to yield the desired MLE, an alternative channel parameters, which differ from the true Θ, should be estimated. Consequently, the training procedure of LoRD-Net does not aim to recover the true CSI, but the one which yields a competitive objective which facilitates symbol detection, thus accounting for the overall system task.

The proposed training procedure detailed in Algorithm 1 carries out each training stage once in a sequential manner. This strategy can be extended to optimizing the hyperparameters and the weights in an alternating fashion, i.e. repeating the stages multiple times, while using the learned $\boldsymbol{\phi}$ in Stage 2 in the Stage 1 that follows. Alternatively, the hyperparameters and the weights can be learned jointly in an end-to-end manner, by optimizing (16) with respect to both Θ and $\boldsymbol{\phi}$ simultaneously. The main requirement for carrying out these training strategies compared to that detailed in Subsection III-C is that the same number of layers L should be used when learning both Θ and $\boldsymbol{\phi}$, while when these stages are carried out once sequentially, it is preferable to use large L at Stage 1 and a smaller value, which dictates the number of learned weights, in Stage 2. Furthermore, our numerical evaluations show that training once in a two-stage fashion via Algorithm 1 yields similar and sometimes even improved performance compared to learning both Θ and $\boldsymbol{\phi}$ simultaneously in a one-stage manner, as well as when alternating between these two stages, as demonstrated in Section IV.

A possible extension of the training procedure is to account for ADCs with more than one bit, as well as allow LoRD-Net to optimize the quantization thresholds $\boldsymbol{b}$ in light of the overall symbol recovery task. While accounting for multi-level ADCs is a rather simple extension achieved by reformulating the objective function (3), optimizing the quantization thresholds requires modifying the overall training strategy. The challenge here is that modifying $\boldsymbol{b}$ results in different one-bit measurements $\boldsymbol{r}$. In a communication setup, in which periodic pilots are transmitted, one can envision gradual optimization of $\boldsymbol{b}$ between consecutive pilot sequences, using their corresponding one-bit observations to further optimize LoRD-Net. The study of LoRD-Net with multi-level ADCs and optimized thresholds is left for future work.

IV. Numerical Study

In this section, we numerically evaluate LoRD-Net[1], and compare its performance with state-of-the-art model-based and data-driven methodologies. As a motivating application for the proposed LoRD-Net, we focus on the evaluation of LoRD-Net for blind symbol detection task in one-bit MIMO wireless communications. In the following, we first detail the considered one-bit MIMO simulation settings in Subsection IV-A, after which we evaluate the receiver performance, compare LoRD-Net to alternative unfolded architectures, and numerically investigate its training procedure in Subsections IV-B, IV-C, and IV-D, respectively.

A. Simulation Setting

We consider an up-link one-bit multi-user MIMO scenario as in (2). We focus on a single cell in which a base station (BS) equipped with m antenna elements serves n single-antenna users. Specifically, we consider two cases of $(m,n) = (128,16)$ and $(m,n) = (64,10)$, i.e., a 128×16 and a 64×10 MIMO channel setup. The transmitted symbols of the users, represented by the unknown vector $\boldsymbol{x}$, are randomized in an independent and identically distributed (i.i.d.) fashion from a BPSK constellation set $\mathcal{M} = \{-1,+1\}$. The projection mapping is thus $\mathcal{P}_{\mathcal{M}^n}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{x})$, where the sign function is applied element-wise on the vector argument. In the sequel, we assume that while the channel matrix $\boldsymbol{H}$, representing the CSI, is not available at the BS, the noise statistics $\boldsymbol{C}$ are known and are fixed to $\boldsymbol{C} = \boldsymbol{I}$. Accordingly, our goal is to utilize LoRD-Net to recover the transmitted symbols from the one-bit measurements. Note that, as we will show later in this section, the proposed methodology can carry out the task of symbol detection even for the case in which the noise statistics $\boldsymbol{C}$ is unknown. For instance, consider having the domain knowledge that the covariance matrix admits a diagonal structure $\boldsymbol{C} = \operatorname{Diag}(\sigma_0^2, \sigma_1^2, \cdots, \sigma_{m-1}^2)$. Then, one can cast such a diagonal matrix as a trainable parameter in addition to other trainable parameters of the network, and successfuly carry out the training procedure. In such scenarios, we note that a covariance matrix is always positive definite, and hence, one can incorporate this knowledge by performing the training on squared variables $\{\sigma_i^2\}$ to ensure positive definiteness of the learned matrix $\boldsymbol{C}$—more on this below.

Channel Models: We evaluate LoRD-Net under two channel models: *(i)* i.i.d. Rayleigh fading channels, where $\boldsymbol{H} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$; and *(ii)* the COST-2100 massive MIMO channel [59]. The COST-2100 channel model is a realistic geometry-based stochastic model which accounts for prominent characteristics of massive MIMO channels, and is considered to be an established benchmark for evaluating MIMO communication systems. We generate the channel matrices for the COST-2100 model for a narrow-band indoor scenario with closely-spaced users at 2.6 GHz band, where the BS is equipped with a uniform linear array (ULA) that has m omni-directional receive antenna elements. The one-bit ADC operation uses zero thresholds, i.e. $\boldsymbol{b} = \boldsymbol{0}$. We define the signal-to-noise ratio (SNR) as:

$$\mathrm{SNR} = \mathbb{E}\left\{\|\boldsymbol{H}\boldsymbol{x}\|_2^2\right\} / \mathbb{E}\left\{\|\boldsymbol{n}\|_2^2\right\}. \tag{17}$$

Benchmark Algorithms: As LoRD-Net combines both model-based and data-driven inference, we compare its performance with state-of-the-art model-based and data-driven methodologies in a one-bit MIMO receiver scenario. In particular, we use the following benchmarking detection algorithms:

- The model-based nML proposed in [30]. The nML algorithm is based on a convex relaxation of the conventional ML estimator, and requires the exact knowledge of the channel parameters $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$. We set the number of iterations of the nML algorithm to 700, and the step-size is chosen using a grid search method to further improve the performance of the nML, while the remaining parameters are those reported in [30]. In addition, each iteration of the nML algorithm admits a similar computational burden as compared to the proposed LoRD-Net. Hence, we later study the convergence speed of both algorithms in terms of total number of iterations.
- The data-driven Deep Soft Interference Cancellation (DeepSIC) methodology proposed in [44], with five learned interference cancellation iterations. DeepSIC is channel-model-agnostic and can be utilized for symbol detection in non-linear settings such as low-resolution quantization setups. Unlike LoRD-Net, which is designed particularly for observations of the form (2) where $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ is unknown, DeepSIC has no prior knowledge of neither the channel model nor its parameters.

LoRD-Net Setting: The LoRD-Net receiver is implemented with $L = 30$ layers. Recall that the first training stage of the LoRD-Net is concerned with finding a competitive objective by carrying out the training of the network over the unknown set of channel parameters $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$. Unless otherwise specified, we focus on the case where only $\boldsymbol{H}$ is unknown, and the correlation matrix of the noise $\boldsymbol{C}$ is available.

[1] The source code is available at: https://github.com/skhobahi/LoRD-Net.

During the first training stage, we set $\delta = 0.01$, and recover $\Theta^\star$ based on the objective (15) using the Adam stochastic optimizer [60] with a constant learning rate of 10^{-3}. Next, we carry out the training of the LoRD-Net during the second stage according to the objective function defined in (16) and over the set of trainable parameters $\boldsymbol{\phi}$, using the Adam optimizer with a learning rate of 10^{-4}, and a mini-batch of size 512. We consider the learning of diagonal pre-conditioning matrices (unfolded weights) during the second training stage. The network is trained for 400 epochs during the first training stage, and 400 epochs during the second training stage, with the same value of $L = 30$ used in both stages.

B. Receiver Performance

Here, we evaluate the performance of the proposed LoRD-Net, comparing it to the aforementioned benchmarks as well as examining its dependence on the number of training samples B. In particular, we numerically evaluate the bit-error-rate (BER) performance versus SNR using different training sizes $B \in \{1024, 2048\}$, for both 128×16 and 64×10 channel configurations. For DeepSIC, we use only $B = 2048$, while the nML recever of [30] operates with perfect CSI, i.e., with full accurate knowledge of Θ. All data-driven receivers are trained for each SNR separately, using a dataset corresponding to that specific SNR value.

The results are depicted in Figs. 4(a) and 4(b) for a 128×16 channel configuration under the Rayleigh fading and COST-2100 channel models, respectively. Furthermore, the BER performance for a 64×10 configuration under both channel models are illustrated in Fig. 5(a) for the Rayleigh fading channel, and in Fig. 5(b), for the COST-2100 channel model. Based on the results presented in Figs. 4 and 5, one can observe that LoRD-Net significantly outperforms the competing model-based and data-driven algorithms and achieves improved detection performance under both simulated channels, as well as both MIMO configurations.

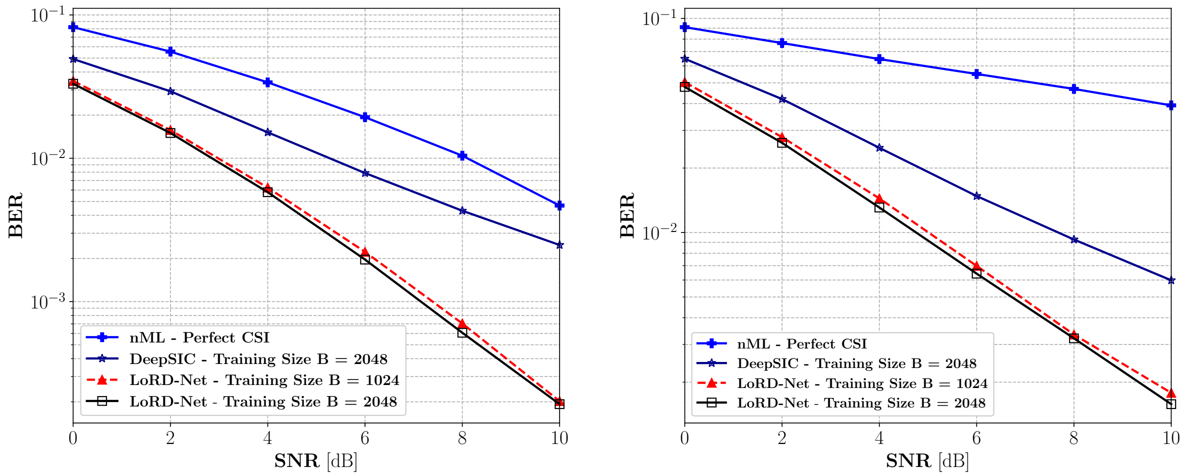
In particular, the nML algorithm, which is designed to iteratively approach the MLE using ideal CSI (prior knowledge of the channel matrix), is notably outperformed by LoRD-Net. Such gains by LoRD-Net, which learns to compute the MLE from data without requiring CSI, compared to the model-based nML algorithm, demonstrate the benefits of learning a competitive objective function combined with a relaxed deep unfolded optimization process. Specifically, the results depicted in Figs. 4-5 illustrate that one can significantly improve the receiver performance by learning a new channel matrix $\boldsymbol{H}$ upon which the learned competitive objective function admits optimal points near the true symbols. The learning of the competitive objective function is possible due to the hybrid model-based/data-driven nature of LoRD-Net, and the fact that it is derived based on the unfolding of first-order optimization techniques. From a computational complexity point-of-view, the depicted performance of the nML algorithm in Figs. 4-5 is achieved by employing 700 iterations of a first-order optimization algorithm, while LoRD-Net uses only $L = 30$ layers/iterations—exhibiting a significant reduction in the computational cost during inference as compared to the nML algorithm.

Comparing LoRD-Net to DeepSIC illustrates that LoRD-Net benefits considerably from its model-aware architecture. The fact that LoRD-Net is particularly tailored to the one-bit system model of (2) allows it to achieve improved accuracy, even in the case of training with small amounts of data. For instance, for the 128×16 MIMO Rayleigh fading channel (see Fig. 4(a)), LoRD-Net trained with $B = 2048$ samples, achieves BER of 10^{-2} at SNR of 3dB, while DeepSIC trained with the same dataset requires SNR as high as 5dB to achieve such an error rate. Considering Fig. 4(b), a similar behavior is observed in the COST-2100 channel, for a BER of 3×10^{-2}. A similar performance gain for LoRD-Net can be observed in a 64×10 configuration; see Fig. 5. Furthermore, it can be observed that the LoRD-Net still outperforms the DeepSIC methodology, even when trained on 2 times less training samples. In particular, for the (128×16) channel setup considered in this part, the total number of trainable parameters of LoRD-Net is merely $|\Theta = \{\boldsymbol{H}\}| + |\boldsymbol{\phi}| = n\,(L + m) = 2528$. For comparison, DeepSIC, which uses and trains a multi-layer fully-connected network for each user at each interference cancellation iterations, consists here of over 8×10^5 trainable parameters. Such a reduction in the number of parameters allows for achieving substantially improved performance with much smaller training points, as observed in Figs. 4-5. Finally, we note that the small number of trainable parameters of LoRD-Net shows its potential for online or real-time training, as proposed in [43]. This can be achieved by using periodic pilots with minimal overhead on the communication, while inducing a relatively low computational burden in its periodic retraining.

So far, we have investigated the performance of the proposed LoRD-Net for scenarios with known noise statistics, and unknown $\boldsymbol{H}$ (i.e., $\Theta = \{\boldsymbol{H}\}$). Next, we investigate the detection performance of LoRD-Net when both the channel and noise covariance matrices are not available, i.e., we set $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ and carry out the training according to the proposed two stage methodology. Specifically, we consider the learning of a diagonally structured $\boldsymbol{C}$ in addition to the channel matrix $\boldsymbol{H}$ for this scenario. Fig. 6 demonstrates the BER versus SNR performance of LoRD-Net under both channel models, when trained using a dataset of size $B = 1024$. The performance of LoRD-Net for the case of $\Theta = \{\boldsymbol{H}\}$ is further provided for comparison purposes. Observing Fig. 6, one can readily conclude that the proposed network can successfully perform the task of symbol detection also when $\boldsymbol{C}$ in unknown. Furthermore, it can be observed that a small gain in performance is achieved for both channel models when $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ as compared to the case of $\Theta = \{\boldsymbol{H}\}$, which is presumably due to the careful addition of more degrees of freedom in learning a competitive surrogate model.

C. Performance of Competing Deep Unfolded Architectures

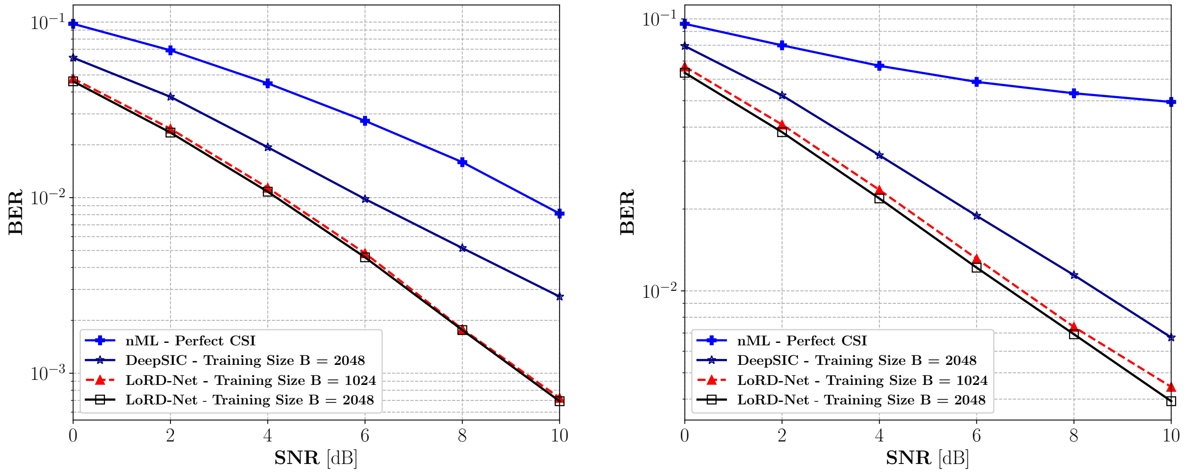
In this part, we compare the performance of the proposed LoRD-Net with alternative deep unfolding-based architectures tailored for the problem at hand. Recall that the architecture of LoRD-Net uses trainable parameters which are shared among the different layers, as illustrated in Fig. 3. Thus, LoRD-Net is comprised of a relatively small number of trainable



(a) Rayleigh fading

(b) COST-2100 massive MIMO

Fig. 4. BER performance versus SNR over a 128 × 16 channel configuration.



(a) Rayleigh fading

(b) COST-2100 massive MIMO

Fig. 5. BER performance versus SNR over a 64 × 10 channel configuration.

parameters, and uses a two-stage learning method to train the shared parameters, representing the competitive model, and the iteration-specific weights, encapsulating the first-order optimization coefficient. Nonetheless, the conventional approach for unfolding first-order optimization techniques is to over-parameterize the iterations, and then, train in an end-to-end manner using a one-stage training procedure discussed earlier. Therefore, to numerically evaluate the proposed unfolding mechanism of LoRD-Net, we next compare it to two conventional unfolding based benchmarks derived from the relaxed MLE:

- *Benchmark 1*: An over-parameterized deep unfolded architecture obtained by setting the computational dynamics for the ith layer as:
$$\bar{g}_{\phi_i}(\boldsymbol{x}_i;\boldsymbol{r}) = \boldsymbol{x}_i - \boldsymbol{A}_i\boldsymbol{R}\,\boldsymbol{\eta}\left(\boldsymbol{R}\left(\boldsymbol{b}-\boldsymbol{B}_i\boldsymbol{x}_i\right)\right). \qquad (18)$$
Here, $\phi_i = \{\boldsymbol{A}_i \in \mathbb{R}^{n\times m}, \boldsymbol{B}_i \in \mathbb{R}^{m\times n}\}$ are the trainable parameters of the ith layer, and $\boldsymbol{R} = \mathrm{Diag}(\boldsymbol{r})$.
- *Benchmark 2*: Here, we again use the unfolded architecture given in (18), while limiting the number of trainable parameters by constraining the rank of the learned matrices. In particular, we set $\boldsymbol{A}_i = \boldsymbol{P}_i\boldsymbol{Q}_i$ and $\boldsymbol{B}_i = \boldsymbol{R}_i\boldsymbol{S}_i$, where $\phi_i = \{\boldsymbol{P}_i \in \mathbb{R}^{n\times r}, \boldsymbol{Q}_i \in \mathbb{R}^{r\times m}, \boldsymbol{R}_i \in \mathbb{R}^{m\times r}, \boldsymbol{S}_i \in \mathbb{R}^{r\times n}\}$ denotes the set of trainable parameters of the ith layer of the unfolded network. The dimension $r < \min(m,n)$ controls the rank of the resulting weight matrices $\{\boldsymbol{A}_i, \boldsymbol{B}_i\}$, and thus the number of trainable parameters.

Comparing (18) with the corresponding dynamics of LoRD-Net in (10), we note that the channel matrix $\boldsymbol{H}$, the pre-conditioning matrices $\boldsymbol{G}_i$, and the noise covariance matrix $\boldsymbol{C}$ are now absorbed into the *per-layer* trainable matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$. Accordingly, these unfolded benchmarks, which follow the conventional approach for unfolding optimization algorithms, are less faithful to the underlying model. These benchmarks also differ from LoRD-Net in their number of trainable parameters. In particular, Benchmark 1 with L layers has $2Lnm$ trainable parameters, while Benchmark 2 has $2Lr(m+n)$ weights, which can be controlled by the setting of the hyperparameter r. For comparison, LoRD-Net has $n(L+m)$ trainable parameters for the case of $\Theta = \{\boldsymbol{H}\}$ and diagonally structured pre-conditioning matrices, while

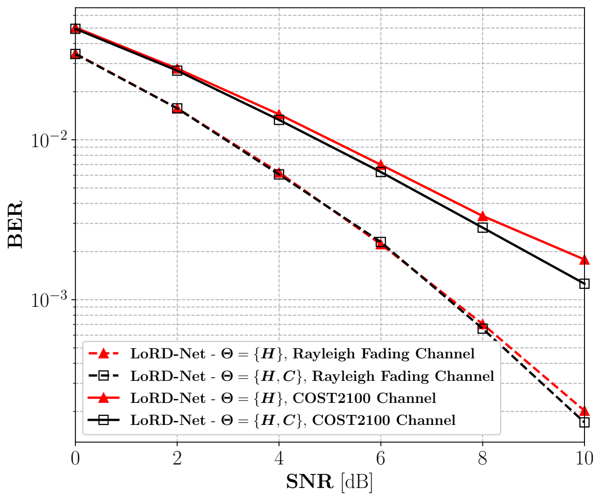


Fig. 6. BER versus SNR for both channel models and a training size of $B = 1024$. The performance of the proposed LoRD-Net is provided for both scenarios of training over $\Theta = \{\boldsymbol{H}\}$ (i.e., known noise statistics $\boldsymbol{C}$), and over $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ corresponding to unknown channel matrix and noise statistics.

for the case of $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$ with a diagonally structured pre-conditioning matrix and noise covariance matrix it has $n(L+m)+m$ trainable parameters.

We evaluate the performance of the proposed LoRD-Net compared to the unfolded benchmarks in the following simulation setup. We consider train all the considered network using a dataset of size $B = 1024$, while the highly-parameterized Benchmark 1 is also trained using $B = 2048$ samples. For Benchmark 2, we set $r = 1$. All architectures have $L = 30$ layers and their performance are evaluated on the same testing dataset of size $B = 2048$. The unfolded benchmarks are trained in the conventional end-to-end fashion. The channel model is a (128×16) Rayleigh fading channel. For the considered scenario above, the LoRD-Net admits a total of 2528 trainable parameters, while Benchmark 1 has a total of 122880 (approximately 50 times more parameters than LoRD-Net), while Benchmark 2 has 8640 trainable parameters.

Fig 7 depicts the BER versus SNR of LoRD-Net compared to the unfolded benchmarks. We observe in Fig. 7 that the proposed LoRD-Net significantly outperforms the conventional unfolding based benchmarks,indicating the gains of the increased level of domain knowledge Incorporated in to the architecture of LoRD-Net and its two stage training procedure. It is also observed that the performance of Benchmark 1 increases with more training samples. Interestingly, for a small training set of $B = 1024$ samples, Benchmark 2, which is obtained by imposing a rank constraint on the trainable parameters of Benchmark 1, achieves improved performance over Benchmark 1, due to its notable reduction in the number of trainable parameters.

D. Training Analysis

In this part, we analyze the performance of the proposed two-stage training procedure described in Subsection III-C. The training aspects of LoRD-Net are numerically evaluated for the 128×16 Rayleigh channel model detailed before.

Following our insight on the ability of LoRD-Net to accurately train with small datasets, we begin by evaluating the performance of the LoRD-Net versus the training data size B. For this study, we generate training datasets of size $B \in \{32, 64, 128, 256, 512, 1024, 2048\}$ and evaluate the performance of LoRD-Net using 2048 test samples. Fig. 8 depicts the BER achieved for each training size B, for SNR $\in \{0, 2, 4, 6, 8, 10\}$ dB. We can observe from Fig. 8 that the performance of the LoRD-Net improves across all SNR values, where the improvements are most notable for $B \leq 256$. Interestingly, it may be concluded from Fig. 8 that LoRD-Net is capable of accurately and reliably performing the task of symbol detection without CSI with as few as $B = 512$ samples. The ability of LoRD-Net to train with very few training samples (compared to the black-box DNN models for one-bit MIMO receivers [22], [24], as well as the DeepSIC architecture), stems from its incorporation of the domain-knowledge in designing the LoRD-Net architecture. This in turn leads to far fewer trainable parameters requiring much less training samples for optimizing the network.

Next, we analyze the performance and the effect of the two stage training methodology detailed in Algorithm 1 on the detection performance of the LoRD-Net architecture. Re-

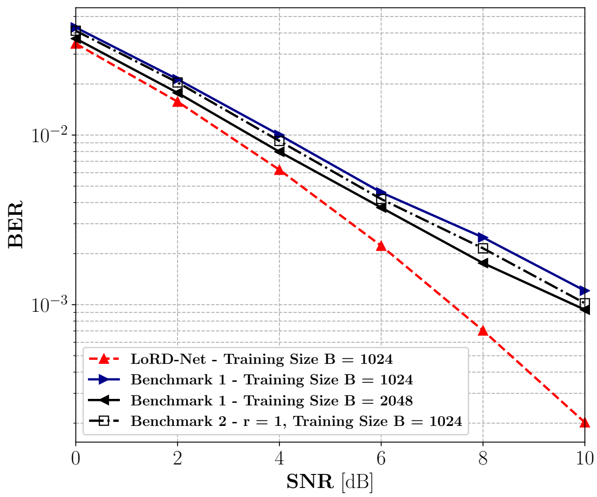


Fig. 7. BER versus SNR of LoRD-Net compared to the unfolded benchmark for a (128×16) Rayleigh fading channel model.

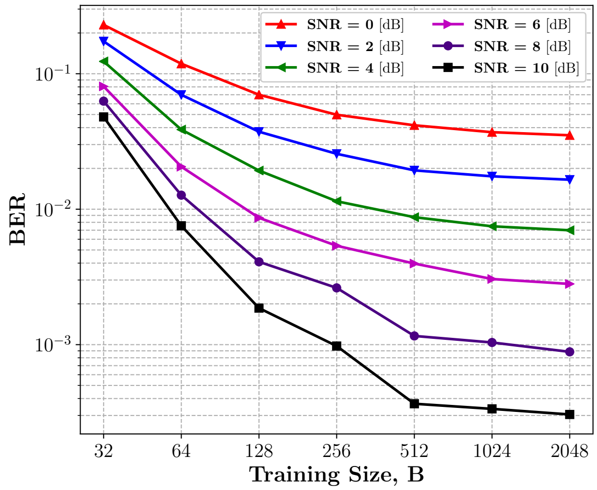


Fig. 8. BER versus training size B for the Rayleigh fading channel.

call that the first training stage is concerned with finding a competitive objective function through an optimization of LoRD-Net over the unknown system parameters Θ, while the second training stage tunes the positive definite preconditioning matrices $\boldsymbol{\phi} = \{\boldsymbol{G}_i\}$ to accelerate the convergence of the LoRD-Net to the optimal points of the obtained competitive objective. To numerically evaluate the performance of the training methodology, we set SNR $= 8$ dB, and generate a training dataset of size $B = 512$ and a testing dataset of size 2048. Then, we compare performance of Algorithm 1 with two other competing training procedures:

• *One-Stage Training*: Here, the weights $\boldsymbol{\phi}$ and the unknown system parameters Θ are jointly learned in a single stage. The objective of this one stage training procedure for LoRD-Net is

$$\min_{\boldsymbol{\phi}=\{\boldsymbol{G}_l\}_l,\Theta\in\boldsymbol{\Theta}} \frac{1}{B}\sum_{i=0}^{B-1}\left\|\mathcal{G}_{\boldsymbol{\phi}}^{L}(\boldsymbol{x}_0;\Theta,\boldsymbol{r}_p^i)-\boldsymbol{x}_p^i\right\|_2^2. \qquad (19)$$

• *Alternating Training*: This procedure is concerned with training the network by alternating between the two optimization problems (15) and (16) consecutively with respect to each training epoch. Here, the network is trained over 400 alternations, corresponding to a total of 800 training epochs. Namely, at each epoch index i, we update the variables Θ for odd i and update $\boldsymbol{\phi}$ for even i.

Before we proceed with the evaluation results, we provide some useful connections to notions widely used in the deep learning literature. Generally speaking, the performance of a statistical learning framework and its training procedure is evaluated using its generalization gap and testing error. The *generalization gap* of a model can be defined as the difference between the training and testing errors. Specifically, a model with smaller generalization gap and smaller testing error is highly favourable. Furthermore, a higher generalization gap may indicate that the network has over-fitted to the data, and hence, it does not generalize well. For two models with the same generalization gap, the one with lower testing error is favourable. Fig. 9 depicts the BER versus the training epoch for both the training and testing dataset. We first note that the proposed two-stage training method (Algorithm 1) outperforms the competing procedures, yielding lower testing error values. Interestingly, we observe that the proposed methodology successfully closes the generalization gap as the testing and training error are very close to each other. On the other hand, the other two training procedures admit relatively large generalization gaps, indicating the fact that their utilization has resulted in an over-fitting of the network to the data. Furthermore, it can be observed from Fig. 9 that the major improvement of the detection accuracy of the LoRD-Net is taking place during the first training stage when finding a competitive objective function, i.e., epochs $i < 80(\times 5)$, where a slight improvement in the BER is achieved during the second stage, i.e., $i \geq 80(\times 5)$.

The success of the proposed two stage training procedure in closing the generalization gap compared to the one stage training procedure is presumably due to the fact that the two-stage training approach leads to an *implicit regularization* on the model capacity limiting the total number of parameters used during the entire training procedure. On the contrary, the one stage training procedure allows the neural network to use its full capacity leading to an over-fitting and a larger generalization gap, as observed in Fig. 9.

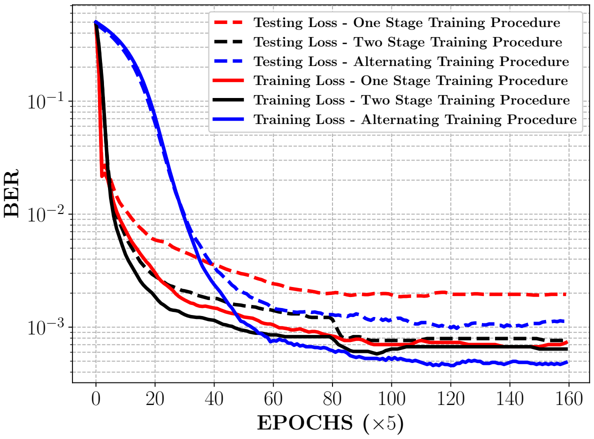
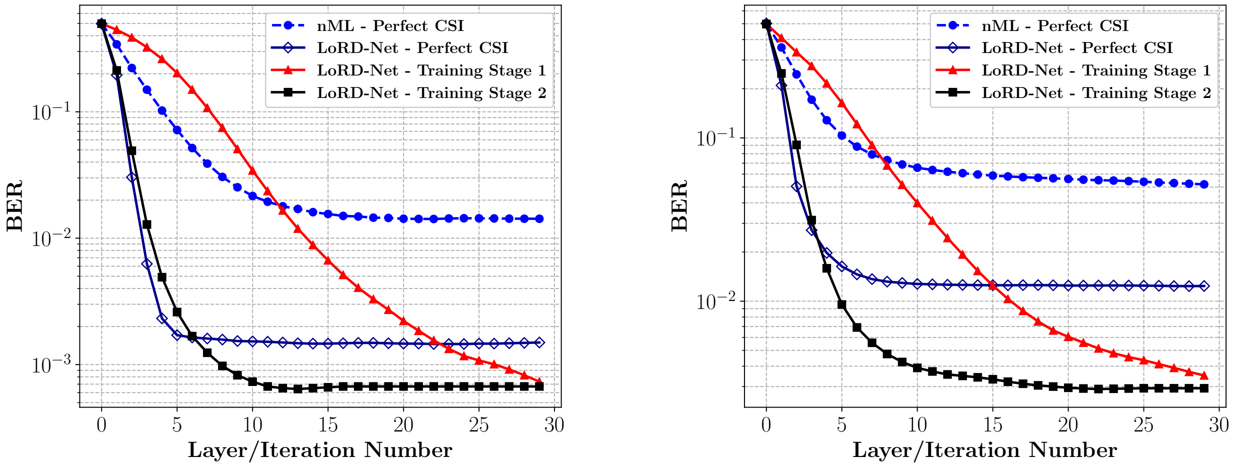


Fig. 9. BER versus the training epoch number of LoRD-Net, Rayleigh fading channel, SNR $= 8$ dB.

As discussed in Subsection III-C, the second training stage allows LoRD-Net to achieve fast inference, i.e., accelerated convergence to the optimal points of the competitive objective function. To illustrate this behavior, we perform a per-layer BER evaluation of LoRD-Net, exploiting the interpretable model-based nature of the LoRD-Net, in which each layer represents an unfolded first-order optimization iteration, and thus its output can be used as an estimate of the transmitted symbols. Figs. 10(a) and 10(b) depict the BER versus the layer/iteration number of LoRD-Net at the completion of training stages 1 and 2, for the Rayleigh fading channel and the COST-2100 channel model, respectively. We observe in Fig. 10 that the convergence of LoRD-Net after the completion of the first training stage is slow and requires at least $L = 30$ layers/iterations to converge. Interestingly, we note from Fig. 10 that the second training stage indeed results in an acceleration of the convergence of LoRD-Net via learning the best set of pre-conditioning matrices for the problem at hand in an end-to-end manner. In particular, after the completion of the second training stage, LoRD-Net can accurately and reliably recover the symbols with as few as 10 layers. This observation hints that one can consider further truncation of the LoRD-Net after the training to reduce the computational complexity while maintaining its superior performance.

In order to quantify the quality of the learned competitive objective in closing the gap between the discrete optimization problem and its continuous version, we further provide the per-iteration performance of the nML algorithm and the LoRD-Net algorithm which operate with perfect CSI. For this scenario, LoRD-Net utilizes the true Θ, and is thus optimizer only over the weights $\boldsymbol{\phi}$ while employing the exact channel model $\boldsymbol{H}$. It is observed from Figs. 10(a)-10(b) that learning a new surrogate model for the continuous optimization problem at hand is indeed highly beneficial and admits a far superior performance in recovering the transmitted symbols. The analysis provided in Fig. 10 further supports the rationale behind the proposed two-stage training methodology, and the fact that the second



(a) Rayleigh fading

(b) COST-2100 Massive MIMO

Fig. 10. BER performance of LoRD-Net after completing training stages 1 and 2 versus the layer/iteration number for (a) the Rayleigh fading channel, and (b) the COST-2100 massive MIMO channel, with $\mathrm{SNR} = 8$ dB.

training stage results in an acceleration of the underlying first-order optimization solver (i.e., achieving a much faster descent per step) upon which the layers of the LoRD-Net are based.

V. Conclusion

In this work, we introduced LoRD-Net, which is a hybrid data-driven and model-based deep architecture for blind symbol detection from one-bit observations. The proposed methodology is based the unfolding of first-order optimization iterations for the recovery of the MLE. We proposed a two-stage training procedure incorporating the learning of a competitive objective function, for which the unfolded network yields an accurate recovery of the transmitted symbols from one-bit noisy measurements. In particular, owing to its model-based nature, LoRD-Net has far fewer trainable parameters compared to its data-driven counterparts, and can be trained with very few training samples. Our numerical results demonstrate that the proposed LoRD-Net architecture outperforms the state-of-the-art model-based and data-driven symbol detectors in multi-user one-bit MIMO systems. We also numerically illustrate the benefits of the proposed two-stage training procedure, which allows to train with small training sets and infer quickly, due to its interpretable model-aware nature.

References

[1] S. Khobahi, N. Naimipour, M. Soltanalian, and Y. C. Eldar, “Deep signal recovery with one-bit quantization,” in *Proc. IEEE ICASSP*, May 2019, pp. 2987–2991.

[2] Y. C. Eldar, *Sampling theory: Beyond bandlimited systems*. Cambridge University Press, 2015.

[3] R. H. Walden, “Analog-to-digital converter survey and analysis,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, 1999.

[4] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, “What will 5G be?” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[5] Y.-S. Jeon, N. Lee, S.-N. Hong, and R. W. Heath, “One-bit sphere decoding for uplink massive MIMO systems with one-bit ADCs,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4509–4521, 2018.

[6] S. Rao, G. Seco-Granados, H. Pirzadeh, and A. L. Swindlehurst, “Massive MIMO channel estimation with low-resolution spatial sigma-delta ADCs,” *arXiv preprint arXiv:2005.07752*, 2020.

[7] A. Ameri, A. Bose, J. Li, and M. Soltanalian, “One-bit radar processing with time-varying sampling thresholds,” *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5297–5308, 2019.

[8] B. Jin, J. Zhu, Q. Wu, Y. Zhang, and Z. Xu, “One-bit LFMCW radar: spectrum analysis and target detection,” *IEEE Trans. Aerosp. Electron. Syst.*, 2020.

[9] F. Xi, N. Shlezinger, and Y. C. Eldar, “BiLiMO: Bit-limited MIMO radar via task-based quantization,” *arXiv preprint arXiv:2010.00195*, 2020.

[10] P. Xiao, B. Liao, and J. Li, “One-bit compressive sensing via Schur-concave function minimization,” *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4139–4151, 2019.

[11] S. Khobahi and M. Soltanalian, “Model-based deep learning for one-bit compressive sensing,” *IEEE Trans. Signal Process.*, vol. 68, pp. 5292–5307, 2020.

[12] N. Shlezinger, Y. C. Eldar, and M. R. Rodrigues, “Hardware-limited task-based quantization,” *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5223–5238, 2019.

[13] S. Salamatian, N. Shlezinger, Y. C. Eldar, and M. Médard, “Task-based quantization for recovering quadratic functions using principal inertia components,” in *Proc. IEEE ISIT*, 2019.

[14] N. Shlezinger and Y. C. Eldar, “Deep task-based quantization,” *Entropy*, vol. 23, no. 1, 2021.

[15] N. Shlezinger, R. J. G. van Sloun, I. A. M. Hujiben, G. Tsintsadze, and Y. C. Eldar, “Learning task-based analog-to-digital conversion for MIMO receivers,” in *Proc. IEEE ICASSP*, 2020.

[16] P. Neuhaus, N. Shlezinger, M. Dörpinghaus, Y. C. Eldar, and G. Fettweis, “Task-based analog-to-digital converters,” *IEEE Trans. Signal Process.*, 2021.

[17] T. Gong, N. Shlezinger, S. S. Ioushua, M. Namer, Z. Yang, and Y. C. Eldar, “RF chain reduction for MIMO systems: A hardware prototype,” *IEEE Syst. J.*, 2020.

[18] S. S. Ioushua and Y. C. Eldar, “A family of hybrid analog–digital beamforming methods for massive MIMO systems,” *IEEE Trans. Signal Process.*, vol. 67, no. 12, pp. 3243–3257, 2019.

[19] H. Wang, N. Shlezinger, Y. C. Eldar, S. Jin, M. F. Imani, I. Yoo, and D. R. Smith, “Dynamic metasurface antennas for MIMO-OFDM receivers with bit-limited ADCs,” *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2643–2659, 2020.

[20] N. Shlezinger, G. C. Alexandropoulos, M. F. Imani, Y. C. Eldar, and D. R. Smith, “Dynamic metasurface antennas for 6G extreme massive MIMO communications,” *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 106–113, 2021.

[21] J. Liu, Z. Luo, and X. Xiong, “Low-resolution ADCs for wireless communication: A comprehensive survey,” *IEEE Access*, vol. 7, pp. 91 291–91 324, 2019.

[22] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, “Deep learning for massive MIMO with 1-bit ADCs: When more antennas need fewer pilots,” *IEEE Wireless Commun. Lett.*, 2020.

[23] A. Klautau, N. González-Prelcic, A. Mezghani, and R. W. Heath, “Detection and channel equalization with deep learning for low resolution

MIMO systems," in *52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1836–1840.

[24] E. Balevi and J. G. Andrews, "One-bit OFDM receivers via deep learning," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4326–4336, 2019.

[25] ——, "Two-stage learning for uplink channel estimation in one-bit massive MIMO," *arXiv preprint arXiv:1911.12461*, 2019.

[26] ——, "Autoencoder-based error correction coding for one-bit quantization," *IEEE Trans. Commun.*, 2020.

[27] D. Kim and N. Lee, "Machine learning based detections for mmwave two-hop MIMO systems using one-bit transceivers," in *Proc. IEEE SPAWC*, 2019.

[28] L. V. Nguyen, A. L. Swindlehurst, and D. H. Nguyen, "SVM-based channel estimation and data detection for one-bit massive MIMO systems," *arXiv preprint arXiv:2003.10678*, 2020.

[29] ——, "Linear and deep neural network-based receivers for massive MIMO systems with one-bit ADCs," *arXiv preprint arXiv:2008.03757*, 2020.

[30] J. Choi, J. Mo, and R. W. Heath, "Near maximum-likelihood detector and channel estimator for uplink multiuser massive mimo systems with one-bit adcs," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 2005–2018, 2016.

[31] C. Risi, D. Persson, and E. G. Larsson, "Massive MIMO with 1-bit ADC," *arXiv preprint arXiv:1404.7736*, 2014.

[32] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "One-bit massive MIMO: Channel estimation and high-order modulations," in *Proc. IEEE ICCW*, 2015, pp. 1304–1309.

[33] M. T. Ivrlac and J. A. Nossek, "On MIMO channel estimation with single-bit signal-quantization," in *ITG smart antenna workshop*, 2007.

[34] J. Mo, P. Schniter, N. G. Prelcic, and R. W. Heath, "Channel estimation in millimeter wave mimo systems with one-bit quantization," in *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2014, pp. 957–961.

[35] A. Mezghani and A. L. Swindlehurst, "Blind estimation of sparse broadband massive MIMO channels with ideal and one-bit ADCs," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2972–2983, 2018.

[36] Y. Li, C. Tao, G. Seco-Granados, A. Mezghani, A. L. Swindlehurst, and L. Liu, "Channel estimation and performance analysis of one-bit massive MIMO systems," *IEEE Trans. Signal Process*, vol. 65, no. 15, pp. 4075–4089, 2017.

[37] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4038–4051, 2017.

[38] J. Mo, P. Schniter, and R. W. Heath, "Channel estimation in broadband millimeter wave MIMO systems with few-bit ADCs," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1141–1154, 2017.

[39] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[40] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *arXiv preprint arXiv:1705.08044*, 2017.

[41] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO detection with deep learning," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1805–1809.

[42] Y. Liao, N. Farsad, N. Shlezinger, Y. C. Eldar, and A. J. Goldsmith, "Deep neural network symbol detection for millimeter wave communications," *arXiv preprint arXiv:1907.11294*, 2019.

[43] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319–3331, 2020.

[44] N. Shlezinger, R. Fu, and Y. C. Eldar, "DeepSIC: Deep soft interference cancellation for multiuser MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1349–1362, 2021.

[45] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Data-driven factor graphs for deep symbol detection," in *Proc. IEEE ISIT*, 2020.

[46] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for joint MIMO channel estimation and signal detection," *arXiv preprint arXiv:1907.09439*, 2019.

[47] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, 2019.

[48] S. Takabe, M. Imanishi, T. Wadayama, R. Hayakawa, and K. Hayashi, "Trainable projected gradient detector for massive overloaded mimo channels: Data-driven tuning approach," *IEEE Access*, vol. 7, pp. 93 326–93 338, 2019.

[49] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," *arXiv preprint arXiv:1906.05774*, 2019.

[50] N. Farsad, N. Shlezinger, A. J. Goldsmith, and Y. C. Eldar, "Data-driven symbol detection via model-based machine learning," *arXiv preprint arXiv:2002.07806*, 2020.

[51] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[52] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, 2020.

[53] N. Naimipour, S. Khobahi, and M. Soltanalian, "Unfolded algorithms for deep phase retrieval," *arXiv preprint arXiv:2012.11102*, 2020.

[54] C. Agarwal, S. Khobahi, A. Bose, M. Soltanalian, and D. Schonfeld, "Deep-url: A model-aware approach to blind deconvolution based on deep unfolded richardson-lucy network," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3299–3303.

[55] S. Khobahi, A. Bose, and M. Soltanalian, "Deep radar waveform design for efficient automotive radar sensing," in *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 2020, pp. 1–5.

[56] N. Naimipour, S. Khobahi, and M. Soltanalian, "Upr: A model-driven architecture for deep phase retrieval," *arXiv preprint arXiv:2003.04396*, 2020.

[57] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *arXiv preprint arXiv:2012.08405*, 2020.

[58] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.

[59] J. Flordelis, X. Li, O. Edfors, and F. Tufvesson, "Massive MIMO extensions to the COST 2100 channel model: Modeling and validation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 380–394, 2019.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.