Model-Inspired Deep Detection with Low-Resolution Receivers

Shahin Khobahi*, Nir Shlezinger†, Mojtaba Soltanalian*, and Yonina C. Eldar‡
*University of Illinois at Chicago, IL, USA, {skhoba2, msol}@uic.edu
†Ben-Gurion University of the Negev, Be'er-Sheva, Israel, nirshl@bgu.ac.il
‡Weizmann Institute of Science, Rehovot, Israel, yonina.eldar@weizmann.ac.il

***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 network, called LoRD-Net, for signal recovering from one-bit measurements. Our approach relies on a model-aware data-driven architecture, based on a 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. 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 [1]. 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 [2]. 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 [3]–[5]. 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 [4]–[8], radar signal processing [9]–[11], and sparse signal recovery [12]–[14]. 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. In the context of MIMO systems, various methods have been proposed in the literature for channel estimation and signal decoding from quantized outputs, including model-based signal processing methods as surveyed in [15], as well as model-agnostic systems based on machine learning and data-driven techniques [16]–[24]. 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.

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 [25]–[30] 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.

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. (*Corresponding author: Shahin Khobahi.*)

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 $\boldsymbol{r} = \text{sign}\,(\boldsymbol{y} - \boldsymbol{b})$, where $\boldsymbol{b} \in \mathbb{R}^m$ denotes the vector of quantization thresholds, and $\text{sign}(\cdot)$ is the sign function, i.e., $\text{sign}(x) = +1$ if $x \geq 0$ and $\text{sign}(x) = -1$ 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 relationship $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$ denotes additive Gaussian noise with a covariance matrix of the form $\boldsymbol{C} = \text{Diag}(\sigma_0^2, \sigma_1^2, \ldots, \sigma_{m-1}^2)$ 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. The overall dynamics of the system are thus compactly expressed as:

$$\boldsymbol{r} = \text{sign}(\boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} - \boldsymbol{b}). \tag{1}$$

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.

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

$$\mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}) \overset{(a)}{=} -\sum_{i=0}^{m-1} \log\left\{Q\left(\frac{r_i}{\sigma_i}\left(b_i - \boldsymbol{h}_i^T\boldsymbol{x}\right)\right)\right\}, \tag{2}$$

represent the log-likelihood objective for a given vector of one-bit observations $\boldsymbol{r}$, where (a) is shown in [8], [14]. The coherent MLE is then given by

$$\hat{\boldsymbol{x}}_{\text{ML}}(\boldsymbol{r}) = \underset{\boldsymbol{x}\in\mathcal{M}^n}{\text{argmax}}\;\; \mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}). \tag{3}$$

Although the MLE in (3) 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 (3) is to relax the search space to be continuous. This results in the following relaxed unconstrained MLE rule:

$$\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}) = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{argmax}}\;\; \mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}). \tag{4}$$

The optimization problem in (4) is convex due to the log-concavity of $Q(\cdot)$, and thus can be solved using first-order gradient optimization. In particular, the gradient of the negative log-likelihood function with respect to the unknown vector $\boldsymbol{x}$ can be compactly expressed as [8], [14]:

$$\nabla_{\boldsymbol{x}}\mathcal{F}_\Theta(\boldsymbol{x}; \boldsymbol{r}) = \boldsymbol{H}^T\tilde{\boldsymbol{R}}\,\boldsymbol{\eta}\left(\tilde{\boldsymbol{R}}\left(\boldsymbol{b} - \boldsymbol{H}\boldsymbol{x}\right)\right), \tag{5}$$

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* $\boldsymbol{R} = \text{Diag}\left(r_0, \ldots, r_{m-1}\right)$.

As $\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r})$ obtained via (4) 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 $\bar{\boldsymbol{x}}(\boldsymbol{r})$, viz. $\mathcal{P}_{\mathcal{M}^n}(\boldsymbol{x}) = \underset{\boldsymbol{z}\in\mathcal{M}^n}{\text{argmin}}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2$.

Tackling a discrete program via continuous relaxation, as done in (4), 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 $\hat{\boldsymbol{x}}_{\text{ML}}(\boldsymbol{r})$. 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 (4) 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 (3). 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.

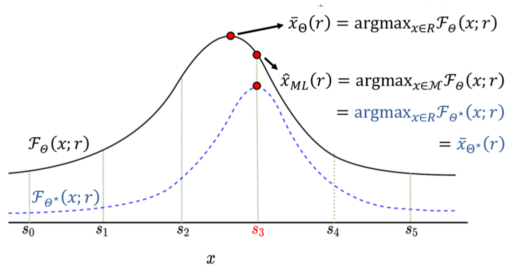


Fig. 1. An illustration of the relation between the optimal point of a competitive objective function (dashed blue line) and the true MLE $\hat{\boldsymbol{x}}_{\rm ML}$ obtained by an exact maximization of the log-likelihood objective function (solid black line) over the discrete set $\mathcal{M}$ as 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 $s_3 \in \mathcal{M}$.

Then, we present the unfolded architecture in Subsection III-B and discuss the training procedure in Subsection III-C.

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 (3) is computationally prohibitive, while its relaxed version in (4) 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 [31], by combining the learning of a *competitive objective*, combined with *deep unfolding* of the relaxed MLE. 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

$$\mathcal{F}_{\Theta^\star}(\boldsymbol{r};\boldsymbol{x}) = \min_{\Theta} \frac{1}{B} \sum_{b=0}^{B-1} \left\|\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}_p^b) - \boldsymbol{x}_p^b\right\|_2^2, \tag{6}$$

where $\bar{\boldsymbol{x}}_\Theta$ is the relaxed MLE (4). The optimization problem (6) 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. 1.

The main difficulty in solving (6) stems from the fact that $\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}) = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{argmax}}\ \mathcal{F}_\Theta(\boldsymbol{x};\boldsymbol{r})$ 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 [26] to compute the relaxed MLE in manner which is differentiable with respect to Θ, as detailed in the following.

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 (4). Our derivation thus begins by formulating the first-order methods to iteratively solve (4) 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:

$$\boldsymbol{x}_{t+1} = \mathcal{G}_{\boldsymbol{\phi}}^t(\boldsymbol{x}_0;\Theta,\boldsymbol{r}) = \boldsymbol{x}_t - \boldsymbol{G}_t \nabla_{\boldsymbol{x}} \mathcal{F}_\Theta(\boldsymbol{x}_t;\Theta,\boldsymbol{r}), \tag{7}$$

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 $\mathcal{G}_{\boldsymbol{\phi}}^t$. The mapping (7) 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 $\mathcal{G}_{\boldsymbol{\phi}}^L(\boldsymbol{x}_0;\Theta,\boldsymbol{r})$ 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 [26], the function $\mathcal{G}_{\boldsymbol{\phi}}^L(\boldsymbol{x}_0;\Theta,\boldsymbol{r})$ 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 (5), the i-th layer computes:

$$g_{\phi_i}(\boldsymbol{x}_i;\Theta,\boldsymbol{r}) = \boldsymbol{x}_i - \boldsymbol{G}_i \boldsymbol{z}_i, \text{ with} \tag{8}$$

$$\boldsymbol{z}_i = \boldsymbol{H}^T \tilde{\boldsymbol{R}}\, \boldsymbol{\eta}\left(\tilde{\boldsymbol{R}}(\boldsymbol{b} - \boldsymbol{H}\boldsymbol{x}_i)\right), \tag{9}$$

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

$$\mathcal{G}_{\boldsymbol{\phi}}^L(\boldsymbol{x}_0;\Theta,\boldsymbol{r}) = g_{\phi_{L-1}} \circ g_{\phi_{L-2}} \circ \cdots \circ g_{\phi_0}(\boldsymbol{x}_0;\Theta,\boldsymbol{r}). \tag{10}$$

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

In principle, one can fix $\boldsymbol{G}_i = \delta \boldsymbol{I}$ for some $\delta > 0$, for which (10) 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 (10). 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 (6), 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 propertities of LoRD-Net: The first is the differentiability of the unfolded architecture with respect to $\boldsymbol{\Theta}$, which facilitates gradient-based optimization 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 (4), 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

$$\bar{\boldsymbol{x}}_\Theta(\boldsymbol{r}) \approx \mathcal{G}^L_{\boldsymbol{\phi}=\{\delta \boldsymbol{I}\}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}). \tag{11}$$

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 (11) into (6) 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:

$$\Theta^\star = \underset{\Theta}{\operatorname{argmin}} \ \frac{1}{B} \sum_{i=0}^{B-1} \left\| \mathcal{G}^L_{\boldsymbol{\phi}=\{\delta \boldsymbol{I}\}}(\boldsymbol{x}_0; \Theta, \boldsymbol{r}^i_p) - \boldsymbol{x}^i_p \right\|_2^2. \tag{12}$$

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 (12) using conventional gradient-based training, e.g., stochastic gradient descent with 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 $\boldsymbol{\phi} = \{\boldsymbol{G}_i\}$ to carry out faster inference. Accordingly, we optimize the weights according to the following criterion:

$$\boldsymbol{\phi}^\star = \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \ \frac{1}{B} \sum_{i=0}^{B-1} \left\| \mathcal{G}^L_{\boldsymbol{\phi}=\{\boldsymbol{G}_l\}_{l=1}^L}(\boldsymbol{x}_0; \Theta^\star, \boldsymbol{r}^i_p) - \boldsymbol{x}^i_p \right\|_2^2. \tag{13}$$

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.

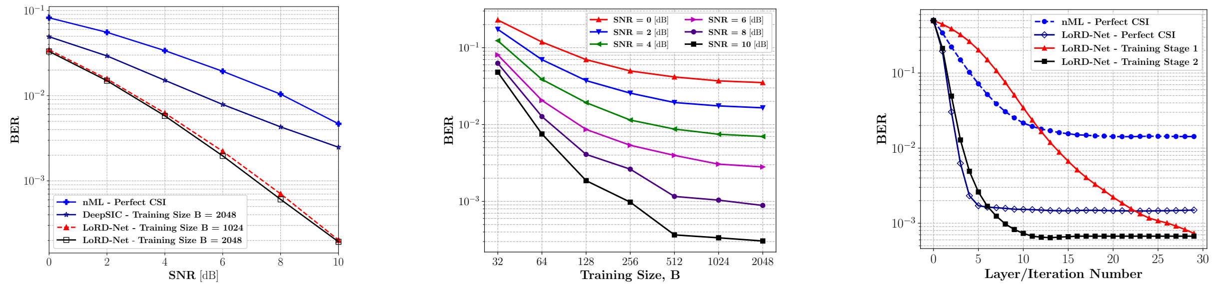
IV. Numerical Study

In this section, we numerically evaluate LoRD-Net, 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, after which we evaluate the performance of LoRD-Net under various scenarios.

Simulation Setting: We consider an up-link one-bit multi-user MIMO scenario as in (1). 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 the case of $(m, n) = (128, 16)$ corresponding to a 128×16 MIMO channel setup, where we assume a Rayleigh fading channel model, i.e., $\boldsymbol{H} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Moreover, we consider the case that the one-bit ADC operation uses zero thresholds. The transmitted symbols of the users $\boldsymbol{x}$ 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 $\boldsymbol{C} = \boldsymbol{I}$. We define the signal-to-noise ratio (SNR) as $\mathrm{SNR} = \mathbb{E}\left\{\|\boldsymbol{H}\boldsymbol{x}\|_2^2\right\} / \mathbb{E}\left\{\|\boldsymbol{n}\|_2^2\right\}$.

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 first algorithm is the model-based nML proposed in [32]. The nML algorithm is based on a convex relaxation of the conventional ML estimator, and requires the exact knowledge of the channel parameters $\Theta = \{\boldsymbol{H}, \boldsymbol{C}\}$. The second algorithm is the data-driven Deep Soft Interference Cancellation (DeepSIC) methodology proposed in [33], with five learned interference cancellation iterations. DeepSIC does not require prior knowledge of neither the channel model nor its parameters and can be utilized for symbol detection in non-linear settings such as low-resolution quantization setups.

LoRD-Net Setting: The LoRD-Net receiver is implemented with $L = 30$ layers. 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, i.e., we set $\Theta = \{\boldsymbol{H}\}$. The LoRD-Net is optimized using the Adam stochastic optimizer [34] with a learning rate of 10^{-3} and 10^{-4} for the first and the second training stage, respectively. Moreover, during the first training stage, we set $\delta = 0.01$. We consider the learning of diagonal pre-conditioning matrices (unfolded weights) during



(a) (b) (c)

Fig. 2. Performance of LoRD-Net for a 128×16 Rayleigh fading channel model: (a) the BER performance versus SNR, (b) the BER versus training size B, and (c) the BER performance of LoRD-Net after completing training stages 1 and 2 versus the layer/iteration number for $\mathrm{SNR} = 8$ dB.

the second training stage. The network is trained for 400 epochs during both training stages.

Receiver Performance: Here, we evaluate the performance of the proposed LoRD-Net, comparing it to the aforementioned benchmarks. In particular, we numerically evaluate the bit-error-rate (BER) performance versus SNR using different training sizes $B \in \{1024, 2048\}$, for the considered scenario. For DeepSIC, we use only $B = 2048$, while the nML recever of [32] 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 Fig. 2(a). Accordingly, one can observe that LoRD-Net significantly outperforms the competing model-based and data-driven algorithms and achieves improved detection performance under the simulated channel. 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. 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 (1) allows it to achieve improved accuracy, even in the case of training with small amounts of data. In addition, the total number of trainable parameters of LoRD-Net is merely $|\Theta = \{\boldsymbol{H}\}| + |\boldsymbol{\phi}| = n\,(L+m) = 2528$, where DeepSIC consists here of over 8×10^5 trainable parameters.

Training Analysis: In this part, we investigate 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. 2(b) depicts the BER achieved for each training size B, for $\mathrm{SNR} \in \{0, 2, 4, 6, 8, 10\}$ dB. We can observe from Fig. 2(b) 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. 2(b) that LoRD-Net is capable of accurately and reliably performing the task of symbol detection without CSI with as few as $B = 512$ samples owing to the incorporation of the domain-knowledge in designing the LoRD-Net architecture.

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. Fig. 2(c) depicts the BER versus the layer/iteration number of LoRD-Net at the completion of training stages 1 and 2. We observe in Fig. 2(c) 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. 2(c) 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. 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 Fig. 2(c) 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.

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. We numerically demonstrate that the proposed LoRD-Net outperforms the state-of-the-art model-based and data-driven symbol detectors in multi-user one-bit MIMO systems.

REFERENCES

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

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

[3] 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.

[4] C. Studer and G. Durisi, "Quantized massive mu-mimo-ofdm uplink," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2387–2399, 2016.

[5] Z. Zhang, X. Cai, C. Li, C. Zhong, and H. Dai, "One-bit quantized massive mimo detection based on variational approximate message passing," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2358–2373, 2017.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

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

[14] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, 2009.

[15] 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.

[16] 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.

[17] 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.

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

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

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

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

[22] 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.

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

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

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

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

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

[28] 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.

[29] 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.

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

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

[32] 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.

[33] 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.

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