DeepSIC: Deep Soft Interference Cancellation for Multiuser MIMO Detection

Nir Shlezinger , *Member, IEEE*, Rong Fu, *Graduate Student Member, IEEE*, and Yonina C. Eldar, *Fellow, IEEE*

Abstract—Digital receivers are required to recover the transmitted symbols from their observed channel output. In multiuser multiple-input multiple-output (MIMO) setups, where multiple symbols are simultaneously transmitted, accurate symbol detection is challenging. A family of algorithms capable of reliably recovering multiple symbols is based on interference cancellation. However, these methods assume that the channel is linear, a model which does not reflect many relevant channels, as well as require accurate channel state information (CSI), which may not be available. In this work we propose a multiuser MIMO receiver which learns to jointly detect in a data-driven fashion, without assuming a specific channel model or requiring CSI. In particular, we propose a data-driven implementation of the iterative soft interference cancellation (SIC) algorithm which we refer to as DeepSIC. The resulting symbol detector is based on integrating dedicated machine-learning methods into the iterative SIC algorithm. DeepSIC learns to carry out joint detection from a limited set of training samples without requiring the channel to be linear and its parameters to be known. Our numerical evaluations demonstrate that for linear channels with full CSI, DeepSIC approaches the performance of iterative SIC, which is comparable to the optimal performance, and outperforms previously proposed learning-based MIMO receivers. Furthermore, in the presence of CSI uncertainty, DeepSIC significantly outperforms model-based approaches. Finally, we show that DeepSIC accurately detects symbols in non-linear channels, where conventional iterative SIC fails even when accurate CSI is available.

Index Terms—Multi-user multiple-input multiple-output (MIMO), deep learning, interference cancellation.

I. INTRODUCTION

MODERN communications systems are subject to constantly growing throughput requirements. In order to meet these demands, receivers are commonly equipped with multiple antennas, and communicate with several transmitters simultaneously to increase the spectral efficiency [2]. Such scenarios, referred to as multiuser multiple-input multiple-output (MIMO) networks, are typically encountered in uplink cellular systems, where the number of transmitters as well as receiver antennas can be very large, as in, e.g., massive MIMO communications [3].

One of the main challenges in multiuser MIMO systems is symbol detection, namely, the recovery of the multiple transmitted symbols at the receiver. Conventional detection algorithms, such as those based on the maximum a-posteriori probability (MAP) rule which jointly recovers all the symbols simultaneously, become infeasible as the number of symbols grows. Alternatively, low complexity separate detection, in which each symbol is recovered individually while treating the rest of the symbols, i.e., the interference, as noise, is strictly sub-optimal [4, Ch. 6], and thus results in degraded throughput [5]. An attractive approach, both in terms of complexity and in performance, is interference cancellation [6]. This family of algorithms implement separate detection, either successively or in parallel, and uses the estimates to facilitate the recovery of the remaining symbols, essentially trading complexity for detection delay. While these methods are prone to error propagation, its effect can be dramatically mitigated by using soft symbol estimates [7]–[9], achieving near MAP performance with controllable complexity.

These aforementioned detection strategies are model-based, namely, they require complete knowledge of the channel model as well as its parameters. When the channel model is unknown, highly complex, or does not faithfully represent the physical environment, these methods cannot be applied. Furthermore, common model-based detection techniques, including interference cancellation schemes, typically assume linear Gaussian channels, in which the noise obeys a Gaussian distribution and the effect of the interference is additive and can thus be canceled by subtraction. Many important future wireless communication scenarios, involving, e.g., quantization-constrained receivers [10], [11], transmission with non-linear amplifiers [12], and communication in the presence of radar interference [13], do not obey the linear Gaussian model. Furthermore, various other communication systems, such as optical networks [14], power-line communications [15], and molecular communications [16], cannot be accurately modeled as linear Gaussian channels. Consequently, the applicability of model-based interference cancellation methods is limited. In addition, even when the channel model is linear and known, inaccurate knowledge of the parameters of the channel, namely, channel state information (CSI)

uncertainty, can significantly degrade the performance of model-based detection mechanisms. This motivates the study of data-driven model-agnostic detection methods.

An alternative to model-based detection algorithms, which is gaining considerable interest recently, is to utilize machine learning tools. Over the last decade, machine learning based systems, and particularly deep neural networks (DNNs), have revolutionized numerous research areas, including computer vision and speech processing [17]. machine learning schemes are gradually influencing the design of digital communication systems, resulting in a multitude of recent works on the application of DNNs in communications; see detailed surveys in [18]–[23]. Unlike model-based receivers, which implement a specified detection rule, machine learning based receivers learn how to map the channel outputs into the transmitted symbols from training, namely, they operate in a data-driven manner, and are typically capable of disentangling the semantic information in complex environments [24]. Furthermore, once trained, DNN-based receivers can implement complicated computations with affordable complexity, making them a promising approach to implement MIMO detection.

Broadly speaking, previously proposed machine learning based receivers can be divided into two main categories: Conventional DNNs and unfolded networks. The first group replaces the receiver processing with a DNN architecture which is established in the machine learning literature. A-priori knowledge of the channel model is accounted for in the selection of the network type, which is typically treated as a black box. For example, recurrent neural networks (RNNs) were applied for decoding sequential codes in [25]; the works [26], [27] used sliding bi-directional RNNs for intersymbol interference (ISI) channels with long memory; reservoir computing was proposed for recovering distorted orthogonal frequency division multiplexing (OFDM) signals in [28]; and the work [29] used variational autoencoders for unsupervised equalization. Such DNNs, which use conventional network architectures that are ignorant of the underlying channel model, can typically operate reliably in various scenarios with or without CSI and channel model knowledge [30], assuming that they were properly trained for the specific setup. Nonetheless, black box DNNs tend to have a large number of parameters, and thus require a large number of samples to train [22], limiting their application in dynamic environments, which are commonly encountered in communications.

Unlike conventional DNNs, which utilize established architectures, in unfolded receivers the network structure is designed following a model-based algorithm. In particular, deep unfolding is a method for converting an iterative algorithm into a DNN by designing each layer of the network to resemble a single iteration [31]. The resulting DNN tends to demonstrate improved convergence speed and robustness compared to the model-based algorithm [32]–[34]. In the context of MIMO symbol detection, the works [35]–[37] designed deep receivers by unfolding the projected gradient descent algorithm for recovering the MAP solution, and [38] proposed to recover continuous-valued signals obtained from one-bit quantized measurements by unfolding gradient descent optimization. Iterative message passing algorithms,

which are known to facilitate multi-user MIMO detection and decoding at controllable complexity [39], [40], were used as a basis for designing data-driven MIMO detectors in [41]–[43] as well as for channel estimation and user activity detection in [44]. Compared to conventional DNNs, unfolded networks are typically interpretable, and tend to have a smaller number of parameters, and can thus be trained quicker [22]. However, these previously proposed receivers all assume a linear channel with Gaussian noise, in which CSI is either available [35]–[38], [41] or estimated from pilots [42]. Consequently, these methods thus do not capture the potential of machine learning in being model independent, and are applicable only under specific channel setups.

In our previous work [45] we proposed ViterbiNet, which is a data-driven implementation of the Viterbi algorithm for detecting symbols transmitted over channels with finite memory. Instead of implementing the receiver as a conventional DNN, or alternatively, unfolding the Viterbi algorithm, we replaced its model-based computations with simple dedicated DNNs. The resulting receiver was thus capable of implementing Viterbi detection in a data-driven fashion using a relatively small number of parameters, while being channel-model-independent. Since ViterbiNet is a data-driven implementation of the Viterbi algorithm whose computational complexity grows rapidly with the cardinality of the channel input, it is not suitable in its current form for MIMO detection. However, the fact that it is capable of learning to implement symbol detection from small training sets motivates the design of a data-driven model-ignorant MIMO symbol detector by integrating machine learning into an established detection algorithm, which is the focus of the current work.

Here, we design a data-driven MIMO detector which is based on model-based interference cancellation methods while being channel-model-independent. In particular, we base our approach on the iterative soft interference cancellation (SIC) symbol detection algorithm proposed in [7] as a model-based method which is capable of approaching MAP performance at affordable complexity. Then, we propose DeepSIC, in which the model-based building blocks of iterative SIC are replaced with simple dedicated DNNs. DeepSIC thus implements iterative SIC in a data-driven fashion, learning to implement MAP-comparable symbol detection from a limited training set. Furthermore, the fact that DeepSIC learns to cancel the interference from training allows it to operate in non-linear setups, where model-based iterative SIC, which assume that the interference is additive and can be canceled by subtracting it, is not applicable.

We propose two methods for training DeepSIC. First we discuss how to jointly train the building blocks consisting of the iterations of iterative SIC in an end-to-end manner. Then, we show how the same training set can be used to train different subsets of the overall network sequentially, by exploiting our prior knowledge of the specific role of each block in the iterative SIC algorithm. The resulting sequential training method allows the receiver to learn its symbol detection mapping from a smaller number of samples. The ability to accurately train with a small training set facilitates its applicability for online training, making the sequential

approach attractive for dynamic environments in which the receiver has to frequently retrain by exploiting the inherent redundancy induced by digital communication protocols [45].

We numerically demonstrate the benefits of DeepSIC in a simulation study. We show that it is capable of approaching the performance of the model-based iterative SIC algorithm in standard linear MIMO channels with Gaussian noise, and that it achieves improved error rate performance compared to previously proposed data-driven MIMO receivers. We also observe that, in the presence of CSI uncertainty, the performance of the model-based iterative SIC, as well as the MAP detector, is significantly degraded, while DeepSIC, which exploits the generalization capabilities of DNNs, is capable of reliably detecting the transmitted symbols. Then, we consider non-linear channels, where DeepSIC is shown to continue to achieve MAP comparable performance, while substantially outperforming iterative SIC, which is incapable of canceling the non-additive interference. Finally, we show how the ability of DeepSIC to train with small training sets can facilitate online tracking of dynamic channels in a self-supervised manner. Our results demonstrate that efficient and robust communication systems can be realized by properly integrating machine learning methods into model-based algorithms.

The rest of this paper is organized as follows: In Section II we present our system model and review the iterative SIC algorithm. Section III proposes DeepSIC, which is a receiver architecture integrating DNNs into the iterative SIC method. Section IV details numerical training and performance results of the proposed receiver, and Section V provides concluding remarks.

Throughout the paper, we use upper-case letters for random variables (RVs), e.g. X, and calligraphic letter for sets, for example, $\mathcal{X}$. Boldface lower-case letters denote vectors, e.g., $\boldsymbol{x}$ is a deterministic vector, and $\boldsymbol{X}$ is a random vector, and the ith element of $\boldsymbol{x}$ is written as $(\boldsymbol{x})_i$. Since upper-case boldface letters are reserved for random vectors, we use upper-case Sans-Sarif fonts for matrices as in [10], [15], [45], [46], e.g., X is a deterministic matrix. The probability measure of an RV X evaluated at x is denoted $p_X(x)$, $\mathcal{R}$ is the set of real numbers, and $(\cdot)^T$ is the transpose operator. All logarithms are taken to basis 2.

II. SYSTEM MODEL AND ITERATIVE SIC

A. System Model

We consider an uplink system in which K single antenna users communicate with a receiver equipped with n_r antennas over a memoryless stationary channel. At each time instance i, the kth user, $k \in \{1, 2, \ldots, K\} \triangleq \mathcal{K}$, transmits a symbol $S_k[i]$ drawn from a constellation $\mathcal{S}$ of size M, i.e., $|\mathcal{S}| = M$. Each symbol is uniformly distributed over $\mathcal{S}$, and the symbols transmitted by different users are mutually independent. We use $\boldsymbol{Y}[i] \in \mathcal{R}^{n_r}$ to denote the channel output at time index i. While we focus on real-valued channels, the system model can be adapted to complex-valued channels, as complex vectors can be equivalently represented using real vectors of extended dimensions. Accordingly, we do not restrict the constellation set $\mathcal{S}$ to take real values, and the receiver
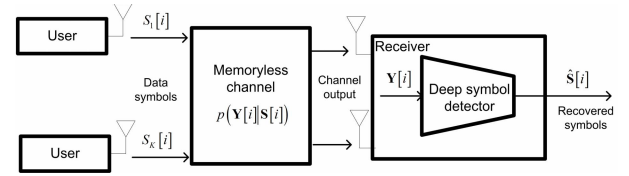

Fig. 1. System model.

architectures detailed in the sequel, which are formulated for real-valued systems, can be applied in complex channels. Since the channel is memoryless, $\boldsymbol{Y}[i]$ is given by some stochastic mapping of $\boldsymbol{S}[i] \triangleq \begin{bmatrix} S_1[i], S_2[i], \ldots, S_K[i] \end{bmatrix}^T$, represented by the conditional distribution measure $p_{\boldsymbol{Y}[i]|\boldsymbol{S}[i]}(\cdot|\cdot)$. The fact that the channel is stationary implies that this conditional distribution does not depend on the index i, and is thus denoted henceforth by $p_{\boldsymbol{Y}|\boldsymbol{S}}(\cdot|\cdot)$. An illustration of the system is depicted in Fig. 1.

We focus on the problem of recovering the transmitted symbols $\boldsymbol{S}[i]$ from the channel output $\boldsymbol{Y}[i]$. The optimal detection rule which minimizes the probability of error given a channel output realization $\boldsymbol{Y}[i] = \boldsymbol{y}$ is the MAP detector. Letting $p_{\boldsymbol{S}|\boldsymbol{Y}}(\cdot|\cdot)$ be the conditional distribution of $\boldsymbol{S}[i]$ given $\boldsymbol{Y}[i]$, the MAP rule is given by

$$\hat{\boldsymbol{s}}_{\text{MAP}}[i] \triangleq \underset{\boldsymbol{s} \in \mathcal{S}^K}{\arg\max} \, p_{\boldsymbol{S}|\boldsymbol{Y}}(\boldsymbol{s}|\boldsymbol{y}). \qquad (1)$$

The MAP detector jointly recovers the symbols of all users by searching over a set of M^K different possible input combinations, and thus becomes infeasible when the number of users K grows. For example, when binary constellations are used, i.e., $M = 2$, the number of different channel inputs is larger than 10^6 for merely $K = 20$ users. Furthermore, the MAP detector requires accurate knowledge of the channel model, i.e., the conditional distribution $p_{\boldsymbol{Y}|\boldsymbol{S}}(\cdot|\cdot)$ must be fully known. A common strategy to implement joint detection with affordable computational complexity, suitable for channels in which $\boldsymbol{Y}[i]$ is given by a linear transformation of $\boldsymbol{S}[i]$ corrupted by additive noise, is interference cancellation [6]. Interference cancellation refers to a family of algorithms which implement joint detection in an iterative fashion by recovering a subset of $\boldsymbol{S}[i]$ based on the channel output as well as an estimate of the remaining interfering symbols. These algorithms facilitate the recovery of the subset of $\boldsymbol{S}[i]$ from the channel output by canceling the effect of the estimated interference using knowledge of the channel parameters, and specifically, how each interfering symbol contributes to the channel output.

Our goal is to design a data-driven method for recovering $\boldsymbol{S}[i]$ from the channel output $\boldsymbol{Y}[i]$, which learns its detection mapping using a training set of n_t pairs of realizations of $\boldsymbol{Y}[i]$ and corresponding $\boldsymbol{S}[i]$, denoted $\{\tilde{\boldsymbol{s}}_j, \tilde{\boldsymbol{y}}_j\}_{j=1}^{n_t}$. In particular, in our model the receiver knows the constellation $\mathcal{S}$, and that the channel is stationary and memoryless. We do not assume that the channel is linear nor that the receiver knows the conditional probability measure $p_{\boldsymbol{Y}|\boldsymbol{S}}(\cdot|\cdot)$. Following the approach of [45], [47], [48], we design our network to implement interference cancellation in a data-driven fashion. In particular, our proposed receiver is based on the iterative
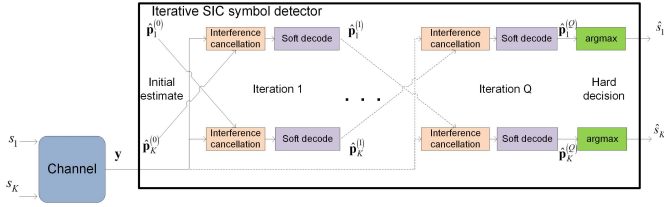
Fig. 2. Soft iterative interference cancellation illustration.

SIC algorithm proposed in [7]. Therefore, as a preliminary step to designing the data-driven detector, we review iterative SIC in the following subsection.

B. Iterative Soft Interference Cancellation

The iterative SIC algorithm proposed in [7] is a multi-user detection method which combines multi-stage (parallel) interference cancellation [49] with soft decisions. Broadly speaking, the detection method operates in an iterative fashion, where in each iteration, an estimate of the conditional distribution of $S_k[i]$ given the channel output $\boldsymbol{Y}[i] = \boldsymbol{y}$ is generated for every user $k \in \mathcal{K}$ using the corresponding estimates of the interfering symbols $\{S_l[i]\}_{l \neq k}$ obtained in the previous iteration. By repeating this procedure iteratively, the conditional distribution estimates are refined, allowing to accurately recover each symbol from the output of the last layer using hard decision. This iterative procedure is illustrated in Fig. 2.

To formulate the algorithm, we consider a channel whose output is obtained as a linear transformation of its input corrupted by additive white Gaussian noise (AWGN), i.e.,

$$\boldsymbol{Y}[i] = \mathsf{H}\boldsymbol{S}[i] + \boldsymbol{W}[i], \tag{2}$$

where $\mathsf{H} \in \mathcal{R}^{n_r \times K}$ is an a-priori known channel matrix, and $\boldsymbol{W}[i] \in \mathcal{R}^{n_r}$ is a zero-mean multivariate Gaussian vector with covariance $\sigma_w^2 \mathsf{I}_K$, independent of $\boldsymbol{S}[i]$.

Iterative soft interference cancellation consists of Q iterations, where each iteration indexed $q \in \{1, 2, \ldots, Q\} \triangleq \mathcal{Q}$ generates K distribution vectors $\hat{\boldsymbol{p}}_k^{(q)} \in \mathcal{R}^M$, $k \in \mathcal{K}$. These vectors are computed from the channel output $\boldsymbol{y}$ as well as the distribution vectors obtained at the previous iteration, $\{\hat{\boldsymbol{p}}_k^{(q-1)}\}_{k=1}^K$, as detailed in the sequel. The entries of $\hat{\boldsymbol{p}}_k^{(q)}$ are estimates of the distribution of $S_k[i]$ for each possible symbol in $\mathcal{S}$, given the channel output $\boldsymbol{Y}[i] = \boldsymbol{y}$ and assuming that the interfering symbols $\{S_l[i]\}_{l \neq k}$ are distributed via $\{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}$. Note that for binary constellations, i.e., $M = 2$, $\hat{\boldsymbol{p}}_k^{(q)}$ can be represented using a single scalar value, as $\left(\hat{\boldsymbol{p}}_k^{(q)}\right)_2 = 1 - \left(\hat{\boldsymbol{p}}_k^{(q)}\right)_1$.

Every iteration consists of two steps, carried out in parallel for each user: *Interference cancellation*, and *soft detection*. Focusing on the kth user and the qth iteration, the interference cancellation stage first computes the expected values and variances of $\{S_l[i]\}_{l \neq k}$ based on $\{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}$. Letting $\{\alpha_m\}_{m=1}^M$ be the indexed elements of the constellation set $\mathcal{S}$, the expected values and variances are computed via

$$e_l^{(q-1)} = \sum_{\alpha_m \in \mathcal{S}} \alpha_m \left(\hat{\boldsymbol{p}}_l^{(q-1)}\right)_m, \tag{3}$$

and

$$v_l^{(q-1)} = \sum_{\alpha_m \in \mathcal{S}} \left(\alpha_m - e_l^{(q-1)}\right)^2 \left(\hat{\boldsymbol{p}}_l^{(q-1)}\right)_m, \tag{4}$$

respectively. The contribution of the interfering symbols from $\boldsymbol{y}$ is then canceled by replacing these symbols with $\{e_l^{(q-1)}\}$ and subtracting their contribution from the channel output. By letting $\boldsymbol{h}_l$ denote the lth column of H, the interference canceled channel output is given by

$$\boldsymbol{Z}_k^{(q)}[i] = \boldsymbol{Y}[i] - \sum_{l \neq k} \boldsymbol{h}_l e_l^{(q-1)} \tag{5a}$$

$$= \boldsymbol{h}_k S_k[i] + \sum_{l \neq k} \boldsymbol{h}_l (S_l[i] - e_l^{(q-1)}) + \boldsymbol{W}[i]. \tag{5b}$$

Substituting the channel output $\boldsymbol{y}$ into (5a), the realization of the interference canceled $\boldsymbol{Z}_k^{(q)}[i]$, denoted $\boldsymbol{z}_k^{(q)}$, is obtained.

To implement soft detection, it is assumed that $\tilde{W}_k^{(q)}[i] \triangleq \sum_{l \neq k} \boldsymbol{h}_l (S_l[i] - e_l^{(q-1)}) + \boldsymbol{W}[i]$ obeys a zero-mean Gaussian distribution, independent of $S_k[i]$, and that its covariance is given by

$$\Sigma_{\tilde{W}_k^{(q)}} = \sigma_w^2 \mathsf{I}_K + \sum_{l \neq k} v_l^{(q-1)} \boldsymbol{h}_l \boldsymbol{h}_l^T. \tag{6}$$

Combining this assumption with (5b), the conditional distribution of $\boldsymbol{Z}_k^{(q)}$ given $S_k[i] = \alpha_m$ is multivariate Gaussian with mean value $\boldsymbol{h}_k \alpha_m$ and covariance $\Sigma_{\tilde{W}_k^{(q)}}$. Since $\boldsymbol{Z}_k^{(q)}[i]$ is given by a bijective transformation of $\boldsymbol{Y}[i]$, it holds that $p_{S_k|\boldsymbol{Y}}(\alpha_m|\boldsymbol{y}) = p_{S_k|\boldsymbol{Z}_k^{(q)}}(\alpha_m|\boldsymbol{z}_k^{(q)})$ for each $\alpha_m \in \mathcal{S}$ under the above assumptions. Consequently, the conditional distribution of $S_k[i]$ given $\boldsymbol{Y}[i]$ is approximated from the conditional distribution of $\boldsymbol{Z}_k^{(q)}$ given $S_k[i]$ via Bayes theorem. Since the symbols are equiprobable, this estimated conditional distribution is computed as

$$
\begin{aligned}
\left(\hat{\boldsymbol{p}}_k^{(q)}\right)_m &= \frac{p_{\boldsymbol{Z}_k^{(q)}|S_k}(\boldsymbol{z}_k^{(q)}|\alpha_m)}{\sum_{\alpha_{m'} \in \mathcal{S}} p_{\boldsymbol{Z}_k^{(q)}|S_k}(\boldsymbol{z}_k^{(q)}|\alpha_{m'})} \\
&= \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{z}_k^{(q)} - \boldsymbol{h}_k \alpha_m\right)^T \Sigma_{\tilde{W}_k^{(q)}}^{-1}\left(\boldsymbol{z}_k^{(q)} - \boldsymbol{h}_k \alpha_m\right)\right\}}{\sum_{\alpha_{m'} \in \mathcal{S}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{z}_k^{(q)} - \boldsymbol{h}_k \alpha_{m'}\right)^T \Sigma_{\tilde{W}_k^{(q)}}^{-1}\left(\boldsymbol{z}_k^{(q)} - \boldsymbol{h}_k \alpha_{m'}\right)\right\}}.
\end{aligned}
\tag{7}
$$

After the final iteration, the symbols are detected by taking the symbol which maximizes the estimated conditional distribution for each user, i.e.,

$$\hat{s}_k = \arg\max_{m \in \{1, \ldots, M\}} \left(\hat{\boldsymbol{p}}_k^{(Q)}\right)_m. \tag{8}$$

The overall joint detection scheme is summarized below as Algorithm 1. The initial estimates $\{\hat{\boldsymbol{p}}_k^{(0)}\}_{k=1}^K$ can be arbitrarily set. For example, these may be chosen based on a linear separate estimation of each symbol for $\boldsymbol{y}$, as proposed in [7].

Algorithm 1 Iterative Soft Interference Cancellation

1: Input: Channel output $\boldsymbol{y}$.
2: Initialization: Set $q = 1$, and generate an initial guess of the conditional distributions $\{\hat{\boldsymbol{p}}_k^{(0)}\}_{k=1}^{K}$.
3: Compute the expected values $\{e_k^{(q-1)}\}$ and variances $\{v_k^{q-1}\}$ via (3)-(4), respectively.
4: *Interference cancellation:* For each $k \in \mathcal{K}$ compute $\boldsymbol{z}_k^{(q)}$ via (5a).
5: *soft detection:* For each $k \in \mathcal{K}$, estimate the conditional distribution $\hat{\boldsymbol{p}}_k^{(q)}$ via (7).
6: Set $q := q + 1$. If $q \leq Q$ go to Step 3.
7: Output: Hard detected output $\hat{\boldsymbol{s}}$, obtained via (8).

C. Advantages and Challenges of Iterative SIC

Iterative SIC has several notable advantages as a joint detection method: In terms of computational complexity, it replaces the joint exhaustive search over all different channel input combinations, required by the MAP detector (1), with a set of computations carried out separately for each user. Hence, its computational complexity only grows linearly with the number of users [6], making it feasible also with large values of K. Unlike conventional separate detection, in which the symbol of each user is recovered individually while treating the interference as noise, the iterative procedure refines the separate estimates sequentially, and the usage of soft values mitigates the effect of error propagation. Consequently, Algorithm 1 is capable of achieving performance approaching that of the MAP detector, which is only feasible for small values of K. The iterative process trades computational complexity for increased detection delay. However, it is numerically observed in [7] that a relatively small number of iterations, such as $Q = 5$, are sufficient for achieving substantial performance gains over separate detection with only a small additional detection delay.

Iterative SIC is specifically designed for linear channels of the form (2). In particular, the interference cancellation in Step 4 of Algorithm 1 requires the contribution of the interfering symbols to be additive. This limits the application of the algorithm in non-linear channels, such as those encountered in optical communications [14], or, alternatively, the models arising in the presence of low-resolution quantizers [11] and non-linear power amplifiers [12]. Additionally, the fact that the distribution of the interference canceled channel output $\boldsymbol{Z}_k^{(q)}$ given $S_k[i]$ is approximated as Gaussian in Algorithm 1 may degrade the performance in channels which do not obey the linear Gaussian model (2).

Furthermore, even when the channel obeys the linear model of (2), iterative SIC requires full CSI, i.e., knowledge of the channel matrix H and the noise variance σ_w^2. Acquiring such knowledge may entail substantial overhead. It is in fact crucial for the receiver to have accurate CSI, since the performance of Algorithm 1 is heavily degraded in the presence of CSI errors, as empirically demonstrated in Section IV.

The dependence on accurate CSI and the assumption of linear channels are not unique to iterative SIC, and are in fact common to most interference cancellation based joint detection algorithms [6]. These limitations motivate the design of a joint detector which exploits the computational feasibility of interference cancellation methods while operating in a data-driven fashion. We specifically select iterative SIC since it is capable of achieving MAP-comparable performance, with a structure that can be readily converted to be data-driven. This is a result of the fact that its specific model-based computations, i.e., Steps 4-5 in Algorithm 1, can be naturally implemented using relatively simple machine learning methods. The resulting receiver, detailed in the following section, integrates machine learning methods into Algorithm 1, allowing it to be implemented for arbitrary memoryless stationary channels without requiring a-priori knowledge of the channel model and its parameters.

III. DeepSIC

In this section, we present a data-driven implementation of iterative SIC. To formulate the proposed receiver, we first derive the machine learning based receiver architecture, which we call DeepSIC, in Subsection III-A. Then, we present methods for training the DNNs embedded in the receiver in Subsection III-B, and discuss its pros and cons in Subsection III-C.

A. Data-Driven Receiver Architecture

Here, we present a receiver architecture which implements iterative SIC in a data-driven fashion. Following the approach of [45], [47], we wish to keep the overall structure of the iterative SIC algorithm, depicted in Fig. 2, while replacing the channel-model-based computations with dedicated suitable DNNs. To that aim, we note that iterative SIC can be viewed as a set of interconnected basic building blocks, each implementing the two stages of interference cancellation and soft detection, i.e., Steps 4-5 of Algorithm 1. While the high level architecture of Fig. 2 is ignorant of the underlying channel model, its basic building blocks are channel-model-dependent. In particular, interference cancellation requires the contribution of the interference to be additive, i.e., a linear model channel as in (2), as well as full CSI, in order to cancel the contribution of the interference. Soft detection requires complete knowledge of the channel input-output relationship in order to estimate the conditional probabilities via (7).

Although each of these basic building blocks consists of two sequential procedures which are completely channel-model-based, we note that the purpose of these computations is to carry out a classification task. In particular, the kth building block of the qth iteration, $k \in \mathcal{K}$, $q \in \mathcal{Q}$, produces $\hat{\boldsymbol{p}}_k^{(q)}$, which is an estimate of the conditional distribution of $S_k[i]$ given $\boldsymbol{Y}[i] = \boldsymbol{y}$ based on $\{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}$. Such computations are naturally implemented by classification DNNs, e.g., fully-connected networks with softmax output layer. An illustration of such a network implementing the kth basic block of the qth iteration is depicted in Fig. 3. Specifically, in Fig. 3 we depict a fully-connected multi-layered network with M output nodes and a softmax output layer, whose inputs are the $n_r \times 1$ channel output $\boldsymbol{y}$ and the previous interference conditional distribution estimates $\{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}$. While the latter consists of $(K-1)M$ entries, it can be represented using $(K-1)(M-1)$
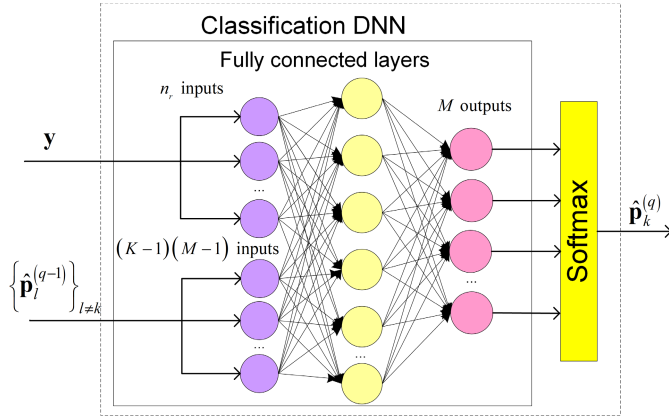
Fig. 3. Conditional probability estimation network model. Here, the network has M output nodes, representing classification with M possible classes.
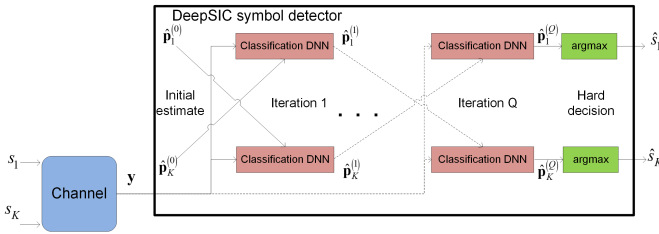

Fig. 4. DeepSIC illustration.

values, since the sum of the entries of each such probability vector is one. When trained to minimize the cross-entropy loss, the building block DNN implements a neural classifier for M possible labels, i.e., constellation points. As such, the entries of the output vector of the softmax layer represents an estimate of the conditional probability for each possible symbol conditioned on $\boldsymbol{y}$ and $\{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}$. Embedding these machine learning based conditional distribution computations into the iterative SIC block diagram in Fig. 2 yields the overall receiver architecture depicted in Fig. 4. We set the initial estimates $\{\hat{\boldsymbol{p}}_k^{(0)}\}_{k=1}^K$ to represent a uniform distribution, i.e., $(\hat{\boldsymbol{p}}_k^{(0)})_m = \frac{1}{M}$ for each $m \in \{1, 2, \ldots, M\}$ and $k \in \mathcal{K}$. We leave the study of different initial estimates and their effect on the overall receiver performance for future research. The resulting data-driven implementation of Algorithm 1 is repeated below as Algorithm 2. Note that the model-based Steps 3-5 of Algorithm 1 whose purpose is to estimate the conditional distributions, are replaced with the machine learning based conditional distribution estimation Step 3 in Algorithm 2.

Algorithm 2 Deep Soft Interference Cancellation (DeepSIC)

1: Input: Channel output $\boldsymbol{y}$.
2: Initialization: Set $q = 1$, and generate an initial guess of the conditional distributions $\{\hat{\boldsymbol{p}}_k^{(0)}\}_{k=1}^K$.
3: *Conditional distribution estimation:* For each $k \in \mathcal{K}$, estimate the conditional distribution $\hat{\boldsymbol{p}}_k^{(q)}$ from $\boldsymbol{y}$ and $\{\hat{\boldsymbol{p}}_k^{(q-1)}\}_{l \neq k}$ using the (q, k)th classification DNN.
4: Set $q := q + 1$. If $q \leq Q$ go to Step 3.
5: Output: Hard detected output $\hat{\boldsymbol{s}}$, obtained via (8).

A major advantage of using classification DNNs as the basic building blocks in Fig. 4 stems from the fact that such machine learning methods are capable of accurately computing conditional distributions in complex non-linear setups without requiring a-priori knowledge of the channel model and its parameters. Consequently, when these building blocks are trained to properly implement their classification task, the receiver essentially realizes iterative soft interference cancellation for arbitrary channel models in a data-driven fashion. In the following subsection we discuss how to train these classification DNNs.

B. Training the DNNs

In order for the machine learning based receiver structure of Fig. 4 to reliably implement joint detection, its building block classification DNNs must be properly trained. Here, we consider two possible approaches to train the receiver based on the training set of n_t pairs of channel inputs and their corresponding outputs $\{\tilde{\boldsymbol{s}}_j, \tilde{\boldsymbol{y}}_j\}_{j=1}^{n_t}$: *End-to-end training*, and *sequential training*.

1) End-to-End Training: The first approach jointly trains the entire network, i.e., all the building block DNNs. While the output of each building block is an $M \times 1$ vector, as illustrated in Fig. 3, the output of the overall interconnection of these DNNs is the set of conditional distributions $\{\hat{\boldsymbol{p}}_k^{(Q)}\}_{k=1}^K$ as illustrated in Fig. 4. Since each vector $\hat{\boldsymbol{p}}_k^{(Q)}$ is used to estimate $S_k[i]$, we use the sum-cross entropy loss as the training objective. Let $\boldsymbol{\theta}$ represent the parameters of the entire network, and $\hat{\boldsymbol{p}}_k^{(Q)}(\boldsymbol{y}, \alpha; \boldsymbol{\theta})$ be the entry of $\hat{\boldsymbol{p}}_k^{(Q)}$ corresponding to $S_k[i] = \alpha$ when the input to the network is $\boldsymbol{y}$ and its parameters are $\boldsymbol{\theta}$. The sum-cross entropy loss over the training set $\{\tilde{\boldsymbol{s}}_j, \tilde{\boldsymbol{y}}_j\}_{j=1}^{n_t}$ can be written as

$$\mathcal{L}_{\mathrm{SumCE}}(\boldsymbol{\theta}) = \frac{1}{n_t} \sum_{j=1}^{n_t} \sum_{k=1}^K - \log \hat{\boldsymbol{p}}_k^{(Q)}\big(\tilde{\boldsymbol{y}}_j, (\tilde{\boldsymbol{s}}_j)_k; \boldsymbol{\theta}\big). \quad (9)$$

Training the receiver in Fig. 4 in this end-to-end manner based on the objective (9) jointly updates the coefficients of all the $K \cdot Q$ building block DNNs. Since for a large number of users, training so many parameters simultaneously is expected to require a large number of input-output pairs, we further propose a sequential training approach detailed next.

2) Sequential Training: To allow the network to be trained with a reduced number of training samples, we note that the goal of each building block DNN does not depend on the iteration index: The kth building block of the qth iteration outputs a soft estimate of $S_k[i]$ for each $q \in \mathcal{Q}$; this estimation is iteratively refined as the iteration index grows. Therefore, each building block DNN can be trained individually, by minimizing the conventional cross entropy loss. To formulate this objective, let $\boldsymbol{\theta}_k^{(q)}$ represent the parameters of the kth DNN at iteration q, and write $\hat{\boldsymbol{p}}_k^{(q)}\big(\boldsymbol{y}, \{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}, \alpha; \boldsymbol{\theta}_k^{(q)}\big)$ as the entry of $\hat{\boldsymbol{p}}_k^{(q)}$ corresponding to $S_k[i] = \alpha$ when the DNN parameters are $\boldsymbol{\theta}_k^{(q)}$ and its inputs are $\boldsymbol{y}$ and $\{\hat{\boldsymbol{p}}_l^{(q-1)}\}_{l \neq k}$. The

cross entropy loss is given by

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{\theta}_k^{(q)}) = \frac{1}{n_t} \sum_{j=1}^{n_t} -\log \hat{\boldsymbol{p}}_k^{(q)} \big(\tilde{\boldsymbol{y}}_j, \{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\}_{l \neq k}, (\tilde{\boldsymbol{s}}_j)_k; \boldsymbol{\theta}_k^{(q)} \big),$$

(10)

where $\{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\}$ represent the estimated probabilities associated with $\tilde{\boldsymbol{y}}_j$ computed at the previous iteration. The problem with training each DNN individually to minimize (10) is that the soft estimates $\{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\}$ are not provided as part of the training set. This challenge can be tackled by training the DNNs corresponding to each layer in a sequential manner, where for each layer the outputs of the trained DNN corresponding to the previous iterations are used to generate the soft estimates fed as training samples. This process is summarized below as Algorithm 3.

Sequential training uses the n_t input-output pairs to train each DNN individually. Compared to the end-to-end training that utilizes the training samples to learn the complete set of parameters, which can be quite large, sequential training uses the same set of input-output pairs to learn a significantly smaller number of parameters, reduced by a factor of $K \cdot Q$, multiple times. Consequently, this approach is expected to require a much smaller number of training samples, at the cost of a longer learning procedure for a given training set, due to its sequential operation, and possible performance degradation as the building blocks are not jointly trained. This behavior is numerically demonstrated in the simulation study detailed in Section IV, where the performance gap between sequential training and end-to-end training is shown to be relatively minor.

Algorithm 3 Sequential Training

1: Input: Training samples $\{\tilde{\boldsymbol{s}}_j, \tilde{\boldsymbol{y}}_j\}_{j=1}^{n_t}$.
2: Initialization: Set $q = 1$, generate an initial guess of the conditional distributions $\{\hat{\boldsymbol{p}}_k^{(0)}\}_{k=1}^K$, and set $\hat{\boldsymbol{p}}_{j,k}^{(0)} = \hat{\boldsymbol{p}}_k^{(0)}$ for each $k \in \mathcal{K}$ and $j \in \{1, 2, \ldots, n_t\}$.
3: **for** each $k \in \mathcal{K}$ **do**
4:     Randomize initial weights $\boldsymbol{\theta}_k^{(q)}$.
5:     Train $\boldsymbol{\theta}_k^{(q)}$ to minimize (10).
6:     Feed $\big\{\tilde{\boldsymbol{y}}_j, \{\hat{\boldsymbol{p}}_{j,l}^{(q-1)}\}_{l \neq k}\big\}_{j=1}^{n_t}$ to the trained DNN, producing $\{\hat{\boldsymbol{p}}_{j,k}^{(q)}\}_{j=1}^{n_t}$.
7: **end for**
8: Set $q := q + 1$. If $q \leq Q$ go to Step 3.
9: Output: Trained network parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k^{(q)}\}$.

C. Discussion

DeepSIC learns to implement the iterative SIC algorithm from training. Consequently, once trained, it shares the main advantages of the model-based algorithm: For scenarios in which iterative SIC is applicable, i.e., linear channels of the form (2), the performance of deepSIC is expected to approach that of the optimal MAP detector. This behavior is numerically observed in the simulation study detailed in Section IV.

Similarly to the model-based iterative SIC algorithm from which DeepSIC originates, the computational complexity of applying DeepSIC grows linearly with the number of users. This makes DeepSIC applicable in MIMO scenarios where conventional MAP detection is infeasible. Furthermore, as DeepSIC consists of an interconnection of relatively compact DNNs, it shares the advantage of DNN-based receivers over iterative model-based receivers in terms of inference speed [22]. Explicitly characterizing the complexity of Deep-SIC is challenging due to the inherent difficulty in quantifying the complexity of training and applying DNNs, which is heavily dependent on the number of parameters and the learning algorithm. In particular, by letting P be the number of parameters in each building block DNN, the overall number of parameters of DeepSIC is $P \cdot K \cdot Q$. Additionally, using sequential training, the training set is required to adapt only P parameters, as each building block is trained individually. This indicates that DeepSIC can learn to implement iterative SIC using a relatively small number of training samples, allowing it to achieve improved performance over previously proposed deep receivers with smaller training sets, as we numerically demonstrate in Subsection IV-A. Additionally, the ability of DeepSIC to accurately train with small data sets facilitates tracking of dynamic channel conditions via self-supervised online training by exploiting the inherent structure of communication protocols [45, Sec. IV], as we demonstrate in Subsection IV-D.

In addition to its ability to implement iterative SIC without prior knowledge of the channel model and its parameters, DeepSIC has two main advantages over the model-based algorithm from which it originates: First, since DeepSIC learns to cancel the interference from training, and does not assume that its contribution is additive, it is applicable also in non-linear channels. Iterative SIC, which attempts to cancel the interference by subtracting its estimate from the channel output, results in increased errors in non-linear channels. Furthermore, even in linear scenarios where iterative SIC is applicable, its performance is highly sensitive to inaccurate CSI. By exploiting the known generalization properties of DNNs, DeepSIC is capable of operating reliably when trained under different channel parameters, which is equivalent to having inaccurate CSI. These advantages are clearly observed in our numerical study in Section IV.

To identify the advantages of DeepSIC over previously proposed machine learning based detectors, we first recall that, as discussed in the introduction, these data-driven receivers can be divided into two main types: The first family of deep receivers implements symbol detection using a single conventional network, generally treated as a black box. While their architecture can account for some a-priori knowledge of the scenario, such as OFDM signaling [28], [30], channel memory [26], [27], and the presence of low resolution quantizers [50], [51], the design is typically not based on established detection algorithms. Compared to such deep receivers, DeepSIC, which learns to implement only the model-based computations of an established algorithm, has fewer parameters, and can thus be trained using smaller training sets, allowing it to be quickly retrained in the presence of dynamic environments. Furthermore, unlike black box DNNs, the architecture of DeepSIC

is interpretable, and, when properly trained, it is expected to achieve the MAP-comparable performance of iterative SIC.

The second family of machine learning-driven MIMO receivers unfolds a model-based iterative optimization algorithm for finding the MAP solution (1), such as projected gradient descent, into a DNN, see, e.g., [35]–[37], [41] as well as [22, Sec. II]. However, these methods assume a linear channel model of the form (2). Furthermore, these previous receivers typically require CSI, obtained either from a-priori knowledge or via channel estimation as in [42]. Unlike these previous receivers, DeepSIC, which is also based on a model-based algorithm, is independent of the channel model, and can efficiently learn to detect in a wide variety of channel conditions, ranging from linear Gaussian channels to non-linear Poisson channels, as numerically demonstrated in Section IV.

Finally, we note that the architecture of DeepSIC, which is depicted in Fig. 4, is related to the concepts of deep mutual learning [52], as well as plug-and-play methods [53]. In particular, in deep mutual learning, a set of relatively small DNNs are each trained individually while allowing to share some information between intermediate layers to facilitate training. By considering the Q building blocks corresponding to each user as a single DNN, DeepSIC can be considered as a form of mutual learning between those networks. However, while conventional mutual learning is based on heuristic arguments, DeepSIC arises from an established detection algorithm, which is particularly suitable for the problem of MIMO detection. As such, the building blocks of DeepSIC are not arbitrary layers, but classification networks designed and trained to implement the corresponding computation of iterative SIC in a data-driven fashion. Plug-and-play methods implement regularized optimization algorithms involving proximal operations by replacing these computations with denoiser DNNs, allowing the optimization process to be carried out in a data-driven manner without analytically accounting for the regularization. Our approach in designing DeepSIC thus bears some similarity to plug-and-play methods, in the sense that both approaches integrate DNNs into an established algorithm by replacing some specific model-based computations with DNNs. While their underlying rationale may be similar, DeepSIC and plug-and-play methods are fundamentally different in the algorithm from which they arise, as well as their target application, i.e., MIMO detection versus inverse problems in image processing.

IV. NUMERICAL STUDY

In the following section we numerically evaluate DeepSIC in several relevant multiuser MIMO detection scenarios.[1] We first consider linear Gaussian channels in Subsection IV-A, for which conventional model-based iterative SIC as well as the majority of previously proposed DNN-based MIMO detectors are applicable. Then, in Subsection IV-B, we demonstrate the performance gains of DeepSIC in two common non-linear scenarios: Quantized Gaussian setups and Poisson channels. Next, in Subsection IV-C we compare the
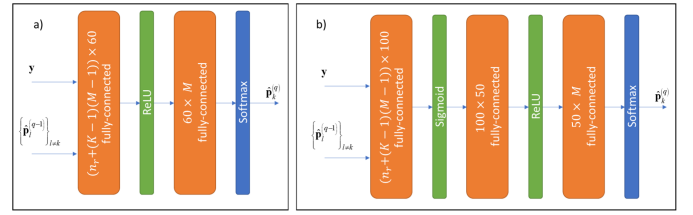
Fig. 5. DNN architectures used as DeepSIC building blocks for: a) end-to-end training; b) sequential training.

methods for training DeepSIC discussed in Subsection III-B, and in Subsection IV-D we evaluate DeepSIC in block-fading channels.

Unless stated otherwise, we trained DeepSIC using the ADAM optimizer with a relatively small training set of 5000 training samples, and tested over 20000 symbols. The motivation for using small training sets is to demonstrate the ability of DeepSIC to train with a sample set of the order of a preamble sequence, e.g., [54, Ch. 17], indicating its feasibility to exploit the structure induced by communication protocols to adapt in dynamic environments, as we demonstrate in Subsection IV-D. We simulate DeepSIC with both end-to-end training as well as sequential training. Since the latter strategy sequentially adapts subsets of the building blocks, it can tune a larger number of parameters using the same training set compared to end-to-end training, where all the building blocks are jointly trained. Consequently, in the implementation of the DNN-based building blocks of DeepSIC depicted in Fig. 3, we used a different fully-connected network for each training method. In particular, for end-to-end training we used a compact network consisting of a $(n_r + (K-1)(M-1)) \times 60$ layer followed by ReLU activation and a $60 \times M$ layer, as illustrated in Fig. 5(a). For sequential training, we used three fully-connected layers: An $(n_r + (K-1)(M-1)) \times 100$ first layer, a 100×50 second layer, and a $50 \times M$ third layer, where a sigmoid and a ReLU intermediate activation functions were used, respectively. The resulting DNN is illustrated in Fig. 5(b). We note that the different training methods are also compared with the same DNN structures in Subsection IV-C, allowing to determine which method should be used based on the training set size.

A. Linear Gaussian Channels

We first consider a linear AWGN channel whose input-output relationship is given by (2). Recall that the model-based algorithm from which DeepSIC originates, i.e., iterative SIC, as well as previously proposed unfolding-based data-driven MIMO receivers [22, Sec. II], are all designed for such channels. Consequently, the motivation of the following study is to compare DeepSIC in terms of performance and robustness to competing detectors in a scenario for which these previous schemes are applicable. In particular, we evaluate the symbol error rate (SER) of the following MIMO detectors:

- The MAP detector, given by (1).
- The iterative SIC algorithm detailed in Algorithm 1.
- DeepSIC with the sequential training method, referred to in the following as *Seq. DeepSIC*.

- DeepSIC with end-to-end training based on the sum cross entropy loss (9), referred to henceforth as *E2E DeepSIC*.
- The unfolding-based *DetNet* MIMO detector proposed in [35].

The model-based MAP and iterative SIC detectors, as well as DetNet [35], all require CSI, and specifically, accurate knowledge of the channel matrix H. DeepSIC operates without a-priori knowledge of the channel model and its parameters, learning the detection mapping from a training set sampled from the considered input-output relationship. In order to compare the robustness of the detectors to CSI uncertainty, we also evaluate them when the receiver has access to an estimate of H with entries corrupted by i.i.d. additive Gaussian noise whose variance is given by σ_e^2 times the magnitude of the corresponding entry, where $\sigma_e^2 > 0$ is referred to as the *error variance*. For DeepSIC, which is model-invariant, we compute the SER under CSI uncertainty by using a training set whose samples are randomized from a channel in which the true H is replaced with its noisy version.

We simulate two linear Gaussian channels: A 6×6 channel, i.e., $K = 6$ users and $n_r = 6$ receive antennas, and a 32×32 setup. Since the computational complexity of the MAP rule grows exponentially with K, it is simulated only for the 6×6 channel. Consequently, simulating the 6×6 channel allows us to compare the error rate achieved by DeepSIC to that of the optimal MAP rule, while the purpose of the 32×32 setup is demonstrate the feasibility of DeepSIC in large multi-user MIMO systems, where the MAP rule become computationally prohibitive. The symbols are randomized from a binary phase shift keying (BPSK) constellation, namely, $\mathcal{S} = \{-1, 1\}$ and $M = |\mathcal{S}| = 2$. The channel matrix H models spatial exponential decay, and its entries are given by

$$(\mathsf{H})_{i,j} = e^{-|i-j|}. \qquad i \in \{1, \dots, n_r\}, j \in \{1, \dots, K\}. \quad (11)$$

For each channel, the SER of the considered receivers is evaluated for both perfect CSI, i.e., $\sigma_e^2 = 0$, as well as CSI uncertainty, for which we use $\sigma_e^2 = 0.1$ and $\sigma_e^2 = 0.75$ for the 6×6 and the 32×32 channels, respectively. The numerically evaluated SER values versus the signal-to-noise ratio (SNR), defined as $1/\sigma_w^2$, are depicted in Figs. 6-7 for the 6×6 case and the 32×32 channel, respectively.

Observing Fig. 6, we note that the performance of DeepSIC with end-to-end training approaches that of the model-based iterative SIC algorithm, which is within a small gap of the optimal MAP performance. This demonstrates the ability of DeepSIC to implement iterative SIC in a data-driven fashion. The sequential training method, whose purpose is to allow DeepSIC to train with smaller data sets compared to end-to-end training, also achieves SER which is comparable to iterative SIC. In particular, the optimal MAP detector with perfect CSI achieves SER of 10^{-3} at SNR of approximately 10 dB, while iterative SIC and E2E DeepSIC require 11 dB SNR and Seq. DeepSIC needs an SNR of at least 12 dB to achieve the same SER value. The fact that E2E DeepSIC approaches the performance of iterative SIC with perfect CSI is also observed for the 32×32 channel in Fig. 7, indicating the applicability of DeepSIC in relatively large multiuser MIMO networks, where MAP-based joint detection
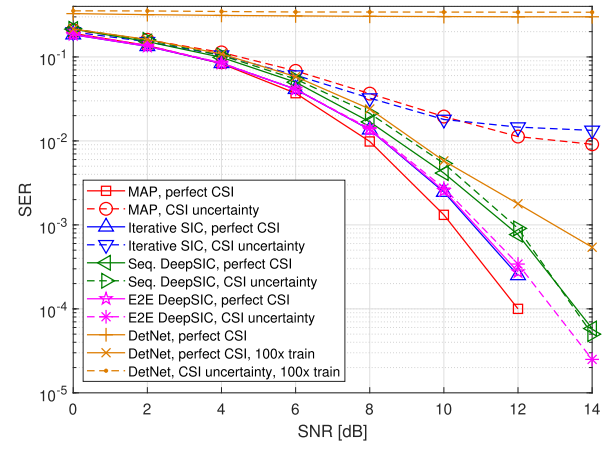

Fig. 6. SER versus SNR of DeepSIC compared to the model-based iterative SIC and the data-driven DetNet of [35], 6×6 linear channel with AWGN. For DeepSIC and DetNet, *Perfect CSI* implies that the receiver is trained and tested using samples from the same channel, while under *CSI uncertainty* they are trained using samples from a set of different channels.
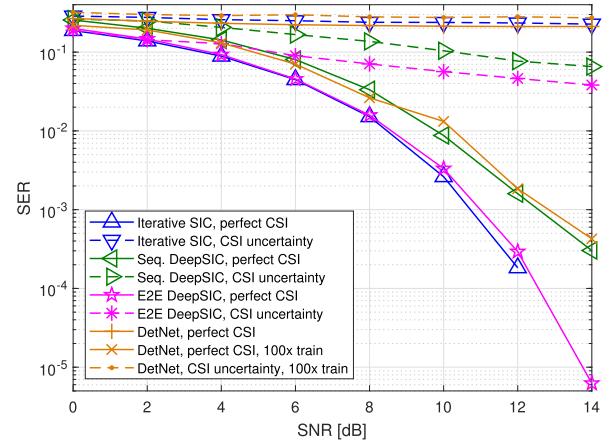

Fig. 7. SER versus SNR of DeepSIC compared to the model-based iterative SIC and the data-driven DetNet of [35], 32×32 linear channel with AWGN. For DeepSIC and DetNet, *Perfect CSI* implies that the receiver is trained and tested using samples from the same channel, while under *CSI uncertainty* they are trained using samples from a set of different channels.

is computationally infeasible. We note that the SNR gap of sequential training compared to end-to-end training is more dominant in the 32×32, and is approximately 1.5 dB for SER of 10^{-3}. The ability of DeepSIC to learn its mapping from a small training set becomes notable when comparing its performance to DetNet: For the same training set of size $n_t = 5000$, DetNet fails to properly adapt and achieves very poor SER performance. Only when provided a hundred times more training samples, denoted *100x train* in Figs. 6-7, DetNet achieves SER values within a small gap of that achieved by Seq. DeepSIC with 1% of the training set.

Figs. 6-7 indicate that the performance DeepSIC is comparable to iterative SIC with accurate CSI. However, in the presence of CSI uncertainty, DeepSIC is observed to substantially outperform the model-based iterative SIC and MAP receivers, as well as DetNet operating with a noisy version of H and trained with a hundred times more inaccurate training samples. In particular, it follows from Fig. 6 that a relatively minor estimation error of variance $\sigma_e^2 = 0.1$ severely deteriorates the

performance of the model-based methods, while the proposed data-driven DeepSIC is hardly affected by the same level of CSI uncertainty. In Fig. 7, in which the relatively large error variance $\sigma_e^2 = 0.75$ is used, we observe that while iterative SIC is inapplicable, DeepSIC is still capable of achieving SER values which decrease below 10^{-1} for SNRs larger than 10 dB in the presence of notable CSI errors. However, the fact that it is trained here using a training set which is not only relatively small, but also quite noisy, induces a notable loss compared to accurate training. Nonetheless, DeepSIC is still shown to be far more robust to CSI uncertainty compared to the model-based iterative SIC.

B. Non-Linear Channels

Here, we evaluate DeepSIC in communication channels which are not modeled as a linear system whose output is corrupted by AWGN. The purpose of this study is to demonstrate that, even though the model-based iterative SIC algorithm is derived assuming a memoryless linear AWGN channel, its data-driven adaptation can be reliably applied in a much broader family of relevant scenarios. Since the unfolding-based DetNet receiver of [35] is designed for linear AWGN setups, here we compare DeepSIC only to the model-based MAP and the iterative SIC receivers.

We begin with a quantized Gaussian channel, which typically models wireless communications in the presence of low-resolution quantizers [10]. The channel output undergoes a 2 bits uniform quantization mapping over the support $[-4, 4]$, given by

$$q(y) = \begin{cases} \text{sign}(y), & |y| < 2 \\ 3 \cdot \text{sign}(y), & |y| > 2. \end{cases}$$

Consequently, using the notations in (2), the channel input-output relationship can be written as

$$\boldsymbol{Y}[i] = q\left(\mathsf{H}\boldsymbol{S}[i] + \boldsymbol{W}[i]\right), \tag{12}$$

where the quantization in (12) is carried out entry-wise. In particular, we consider a receiver with $n_r = 4$ antennas serving $K = 4$ users, each transmitting i.i.d. BPSK symbols. The entries of the channel matrix H are given by (11).

In Fig. 8 we compare the SER achieved by DeepSIC with both end-to-end and sequential training to the performance of the model-based MAP and iterative SIC receivers versus SNR $\in [6, 20]$ dB. As in the previous subsection, we consider both the cases in which the receiver has perfect CSI as well as CSI uncertainty: Under perfect CSI, the model-based MAP and iterative SIC detectors have accurate knowledge of H and σ_w^2, while DeepSIC is trained over samples taken from the same channel model under which it is tested; In CSI uncertainty the model-based receivers have access to a noisy estimate of H with uncertainty variance $\sigma_e^2 = 0.1$, and DeepSIC is trained using samples from the corresponding inaccurate channel model.

Observing Fig. 8, we note that the performance of DeepSIC effectively coincides with that of the optimal MAP rule with perfect CSI, demonstrating its ability to learn to accurately detect in complex environments. For most considered SNR
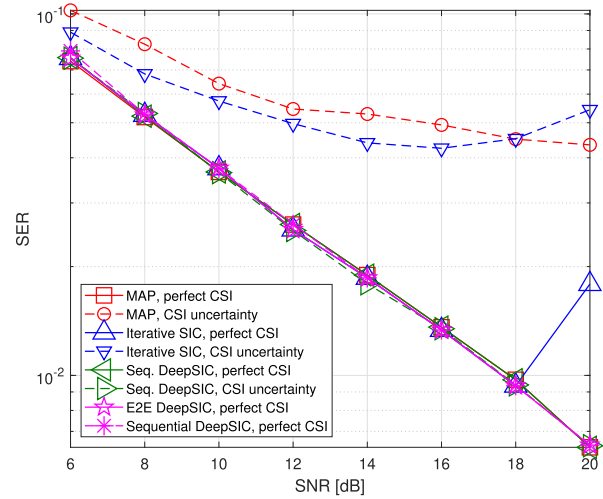

Fig. 8. SER versus SNR of DeepSIC compared to the model-based MAP rule and iterative SIC method, 4×4 quantized Gaussian channel. For DeepSIC, *Perfect CSI* implies that the receiver is trained and tested using samples from the same channel, while under *CSI uncertainty* it is trained using samples from a set of different channels.

values, iterative SIC with perfect CSI also approaches the MAP SER performance, which settles with the observation in [55] that iterative soft detection based equalization methods, such as iterative SIC, achieve excellent performance in quantized Gaussian channels. However, obtaining accurate channel estimation in the presence of low resolution quantization is substantially more challenging compared to conventional linear channels [10], thus in practice, the channel estimates are likely to be inaccurate. In such cases, it is shown in Fig. 8 that model-based methods are highly sensitive to inaccurate CSI, and their error does not monotonically decrease with SNR. This non-monotonic behavior follows since in some scenarios involving quantized observations, the presence of noise, which causes the discrete channel outputs to change their values at some probability, can reduce the errors induced by CSI uncertainty. Unlike the model-based receivers, DeepSIC hardly exhibits any degradation under the same level of uncertainty.

Next, we consider a Poisson channel, which typically models free-space optical communications [14]. As in the quantized Gaussian case, we use $K = 4$ and $n_r = 4$. Here, the symbols are randomized from an on-off keying for which $\mathcal{S} = \{0, 1\}$. The entries of the channel output are related to the input via the conditional distribution

$$(\boldsymbol{Y}[i])_j \, | \, \boldsymbol{S}[i] \sim \mathbb{P}\left(\frac{1}{\sqrt{\sigma_w^2}} \left(\mathsf{H}\boldsymbol{S}[i]\right)_j + 1\right), \tag{13}$$

$j \in \{1, \ldots, n_r\}$, where $\mathbb{P}(\lambda)$ is the Poisson distribution with parameter $\lambda > 0$, and the entries of $\mathsf{H} \in \mathcal{R}^{n_r \times K}$ are given by (11).

The achievable SER of DeepSIC versus SNR under both perfect CSI as well as CSI uncertainty with error variance $\sigma_e^2 = 0.1$ is compared to the MAP and iterative SIC detectors in Fig. 9. Observing Fig. 9, we again note that the performance of DeepSIC is only within a small gap of the MAP performance with perfect CSI, and that the data-driven receiver is more robust to CSI uncertainty compared to the model-based
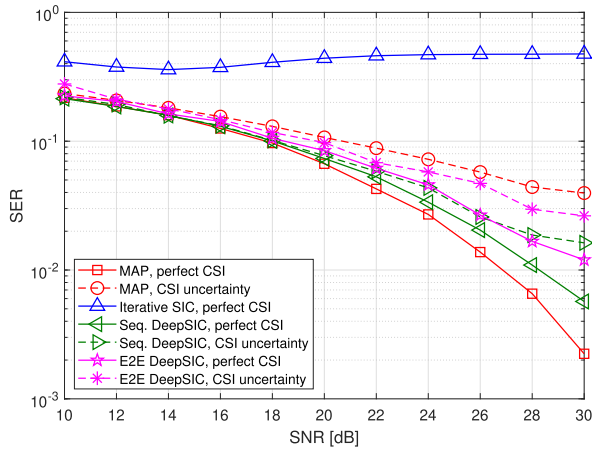
Fig. 9. SER versus SNR of DeepSIC compared to the model-based MAP rule and iterative SIC method, 4×4 Poisson channel. For DeepSIC, *Perfect CSI* implies that the receiver is trained and tested using samples from the same channel, while under *CSI uncertainty* it is trained using samples from a set of different channels.

Fig. 10. SER versus training set size of DeepSIC with sequential training and end-to-end training, 6×6 linear channel with AWGN.

MAP. In particular, DeepSIC with sequential training, which utilizes a deeper network architecture for each building block, outperforms here end-to-end training with basic two-layer structures for the conditional distribution estimation components. We conclude that under such non-Gaussian channels, more complex DNN models are required to learn to cancel interference and carry out soft detection accurately. This further emphasizes the gain of our proposed sequential approach for training each building block separately, thus allowing to train an overall deep architecture using a limited training set based on the understanding of the role of each of its components. Furthermore, it is noted that iterative SIC, which is designed for linear Gaussian channels (2) in which the interference is additive, achieves very poor performance when the channel input-output relationship is substantially different from (2). Since iterative SIC is shown to be unreliable in this setup with accurate knowledge of H, we do not include in Fig. 9 its SER performance with CSI uncertainty.

The results presented in this section demonstrate the ability of DeepSIC to achieve excellent performance and learn to implement interference cancellation from training, under statistical models for which conventional model-based interference cancellation is effectively inapplicable.

C. Training Methods Comparison

In the numerical studies reported in the previous subsections, we used different network architectures for each training method of DeepSIC: For E2E DeepSIC we used two fully-connected layers in implementing the machine learning based building blocks of Fig. 1, while Seq. DeepSIC used three layers. The rationale behind this setting was that for a given training set, Seq. DeepSIC updates distinct subsets of its parameters gradually and can thus adapt larger networks using the same training set compared to end-to-end training. However, it is also demonstrated in the previous subsections that E2E DeepSIC often achieves improved performance compared to Seq. DeepSIC, despite the fact that it uses smaller networks, as the joint training process results in a more accurate detector.
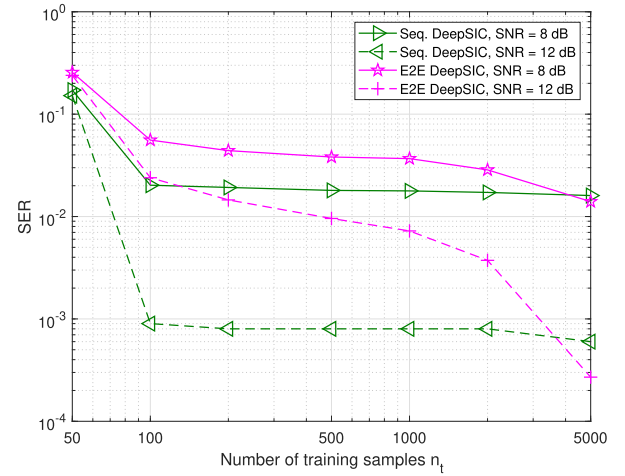
To understand which training method is preferable for a given network architecture and training set, in the following we numerically evaluate the performance of DeepSIC versus the training set size n_t when using the same number of parameters under both training methods. In particular, we fix the building blocks to consist of the architecture used under end-to-end training in the previous subsections, namely, a $(n_r + (K - 1)(M - 1)) \times 60 \times M$ fully-connected two layers. The achievable SER of the training methods versus the training set size n_t for the 6×6 linear Gaussian channel detailed in Subsection IV-A with perfect CSI for SNR values of 8 and 12 dB is depicted in Fig. 10. Observing Fig. 10, we note that the accuracy of end-to-end training is substantially degraded as the number of labeled sampled decreases, since the available training is used to jointly adapt the complete set of parameters of the network. For comparison, sequential training, which exploits the understanding of the role of each building block in the model-based iterative SIC method to train each subsystem individually, is capable of operating reliably with as few as $n_t = 100$ training samples. In particular, it is observed in Fig. 10 that the fact that sequential training uses the same training set to adapt sub-sets of the overall architecture sequentially, facilitates convergence with much smaller training sets compared to end-to-end training. However, the fact that end-to-end training jointly adapts the complete DeepSIC architecture allows it to converge to improved configurations when a sufficient number of training samples is provided. These results demonstrate how converting a model-based algorithm into a data-driven system not only yields a suitable DNN architecture, but can also facilitate its training.

D. Online Tracking of Channel Variations

The ability of DeepSIC with sequential training to tune its parameters using small labeled data sets facilitates its application in dynamic time-varying channels. One of the major challenges of machine learning aided receivers stems from the fact that DNNs require a large volume of training data in order to learn their mapping, and once trained they are

applicable to inputs obeying the same (or a similar) distribution as the one used during training. Since communication channels are typically dynamic and conditions may change significantly over time, DNN-based receivers should re-train periodically in order to track channel variations, without degrading the spectral efficiency. In such cases, the fact that DeepSIC can be efficiently trained with small data sets allows it to track time-varying channels without inducing additional communication overhead by exploiting the inherent structure of digital communication protocols.

We next demonstrate this advantage by applying the self-supervised online training method proposed in [45, Sec. IV] for tracking channel variations by utilizing the presence of coded communications. Here, each user transmits a set of codewords protected by a forward error correction (FEC) code, and each codeword is transmitted as a block of symbols. The receiver applies DeepSIC to detect the transmitted symbols from the channel output and decodes the message using its FEC decoder. If the messages are successfully decoded, they are re-encoded and re-modulated into their corresponding channel input, which is used along with the observed channel output to re-train DeepSIC. Since the FEC decoder can successfully recover the message even when some symbol errors are present, this approach allows an initially trained deep receiver which is capable of re-training using small data sets of the order of a single codeword, to track block-fading time-varying channel conditions.

In particular, we simulate two 4×4 channels: a linear AWGN channel (2) and a Poisson channel (13). Here, each user transmits 50 codewords encoded using a $[255, 239]$ Reed-Solomon FEC code, i.e., each codeword consists of 2040 symbols embedding a message of 1784 bits. The channel observed by the bth codeword is generated with the channel matrix $\mathsf{H}(b)$ whose entries are given by

$$(\mathsf{H}(b))_{i,j} = e^{-|i-j|} \cos((\boldsymbol{\phi})_i \cdot b), \tag{14}$$

$i \in \{1, \ldots, n_r\}$, $j \in \{1, \ldots, K\}$, where we used $\boldsymbol{\phi} = [51, 39, 33, 21]$. The difference between the initial channel matrix and that used during the bth block, given by the squared Frobenius distance $\|\mathsf{H}(b) - \mathsf{H}(0)\|^2$, is depicted in Fig. 11c. For each of the considered channel models, i.e., the linear AWGN channel and the Poisson channel, we evaluate the instantaneous bit error rate (BER) in decoding each codeword of DeepSIC using the method of [45, Sec. IV] for self-supervised channel tracking, where before the first block is transmitted, DeepSIC is trained using 5000 samples corresponding to the channel with index $b = 0$. These results are compared to the model-based MAP detector which knows the channel at each time instance, as well as when it has only knowledge of the initial channel $\mathsf{H}(0)$. Furthermore, we also evaluate the instantaneous BER of DeepSIC without online training when trained only once using 5000 samples corresponding to $\mathsf{H}(0)$, as well as when the set of 5000 training samples constitutes of 10 subsets corresponding to $\mathsf{H}(b)$ with $b \in \{1, \ldots, 10\}$, representing joint learning for multiple channels [56]. The results are depicted in Figs. 11a-11b for the linear AWGN channel and the Poisson channel, respectively. We specifically consider relatively high SNRs of 14 db and 32 dB for the
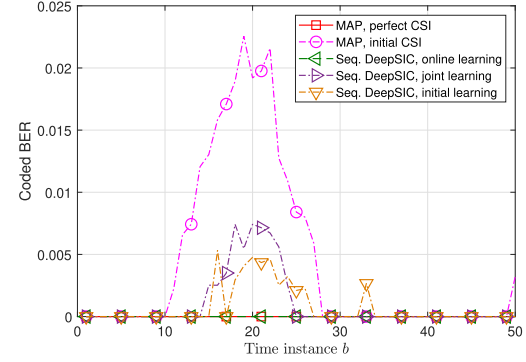
(a) Coded BER for each codeword versus block index, linear channel with AWGN, SNR = 14 dB.
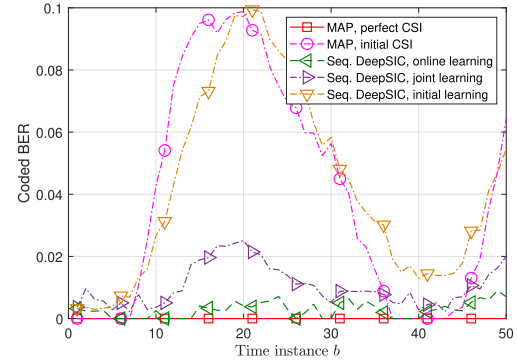
(b) Coded BER for each codeword versus block index, Poisson channel, SNR = 32 dB.
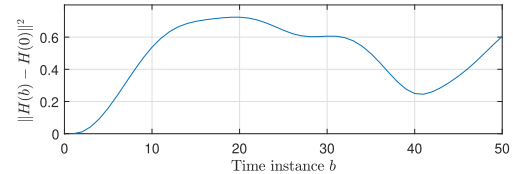
(c) Channel variations versus block index.

Fig. 11.   Channel tracking via online self-supervised training.

Gaussian and Poisson channels, respectively, to focus on scenarios where detection errors are expected to occur mostly due to channel variations, rather than due to the presence of a dominant noise, which is the case in lower SNRs.

Observing Figs. 11a-11b, we note that as the channel parameters begin to deviate considerably compared to their initial value, around the tenth codeword. Receivers which do not track channel variations, i.e., DeepSIC without retraining and the MAP detector operating with the initial CSI, start exhibiting errors. However, DeepSIC which trains online in a self-supervised manner based on its FEC decoder outputs, successfully tracks the variations of the channel, demonstrating low error rates for each codeword, and in fact achieving zero errors over all considered blocks for the linear AWGN channel. This demonstrates how the ability of DeepSIC to adapt with a small number of samples using the sequential training method allows it to track time-varying channels without requiring additional pilots, by merely exploiting the inherent structure of digital communication protocols. Furthermore, we also observe that DeepSIC trained once over various channel conditions via joint learning exhibits reduced error rates compared to the MAP detector with initial CSI as well as DeepSIC

which trains only for the initial channel, particularly for the Poisson channel in Fig. 11b. This illustrates the potential of such joint training to improve the robustness of DeepSIC, as also observed for the CSI uncertainty case in the previous subsections.

V. CONCLUSION

In this work we proposed DeepSIC, a data-driven multiuser MIMO receiver architecture. To derive DeepSIC, we relied on the iterative SIC algorithm, which is an accurate and computationally feasible model-based MIMO detection scheme that can be naturally extended to incorporate machine learning methods. We obtained the data-driven DeepSIC receiver by replacing the model-based building blocks of iterative SIC with dedicated compact DNNs. Unlike its model-based counterpart, DeepSIC is channel-model-independent and can learn to implement interference cancellation in non-linear setups with non-additive interference. We proposed two methods for training DeepSIC: an end-to-end approach and a sequential scheme, where the latter is more suitable for small training sets and can thus be used to quickly adapt in dynamic environments. Our numerical results demonstrate that for conventional linear channels, DeepSIC approaches the MAP performance, outperforming previously proposed DNN-based receivers while demonstrating improved robustness to CSI uncertainty. Finally, we showed that the MAP-comparable performance of deepSIC and its resiliency to uncertainty hold in the presence of non-linear channels, where the applicability of iterative SIC as well as previously proposed machine learning based receivers is limited.

REFERENCES

[1] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deep soft interference cancellation for MIMO detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 8881–8885.

[2] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, no. 3, pp. 311–335, Mar. 1998.

[3] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antenna," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, Nov. 2010, pp. 3600–3950.

[4] A. El Gamal and Y. H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[5] N. Shlezinger and Y. C. Eldar, "On the spectral efficiency of noncooperative uplink massive MIMO systems," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1956–1971, Mar. 2019.

[6] J. G. Andrews, "Interference cancellation for cellular systems: A contemporary overview," *IEEE Wireless Commun*, vol. 12, no. 2, pp. 19–29, Apr. 2005.

[7] W.-J. Choi, K.-W. Cheong, and J. M. Cioffi, "Iterative soft interference cancellation for multiple antenna systems," in *Proc. IEEE Wireless Commun. Netw. Conf. Rec.*, Chicago, IL, USA, Sep. 2000, pp. 304–309.

[8] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[9] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. B. Schlegel, "Iterative multiuser interference reduction: Turbo CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1008–1014, Jul. 1999.

[10] N. Shlezinger, Y. C. Eldar, and M. R. D. Rodrigues, "Asymptotic task-based quantization with application to massive MIMO," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 3995–4012, Aug. 2019.

[11] C. Studer and G. Durisi, "Quantized massive MU-MIMO-OFDM uplink," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2387–2399, Jun. 2016.

[12] I. Iofedov and D. Wulich, "MIMO–OFDM with nonlinear power amplifiers," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4894–4904, Dec. 2015.

[13] L. Zheng, M. Lops, Y. C. Eldar, and X. Wang, "Radar and communication coexistence: An overview: A review of recent methods," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 85–99, Sep. 2019.

[14] M. A. Khalighi and M. Uysal, "Survey on free space optical communication: A communication theory perspective," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2231–2258, 2014.

[15] N. Shlezinger, R. Shaked, and R. Dabora, "On the capacity of MIMO broadband power line communications channels," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4795–4810, Oct. 2018.

[16] N. Farsad, Y. Murin, W. Guo, C.-B. Chae, A. W. Eckford, and A. Goldsmith, "Communication system design and analysis for asynchronous molecular timing channels," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 3, no. 4, pp. 239–253, Dec. 2017.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[18] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[19] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.

[20] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[21] D. Gunduz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. Murthy, and M. van der Schaar, "Machine learning in the air," 2019, *arXiv:1904.12385*. [Online]. Available: http://arxiv.org/abs/1904.12385

[22] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," 2019, *arXiv:1906.05774*. [Online]. Available: http://arxiv.org/abs/1906.05774

[23] H. He, S. Jin, C.-K. Wen, F. Gao, G. Ye Li, and Z. Xu, "Model-driven deep learning for physical layer communications," 2018, *arXiv:1809.06059*. [Online]. Available: http://arxiv.org/abs/1809.06059

[24] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[25] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," 2018, *arXiv:1805.09317*. [Online]. Available: http://arxiv.org/abs/1805.09317

[26] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018.

[27] Y. Liao, N. Farsad, N. Shlezinger, Y. C. Eldar, and A. J. Goldsmith, "Deep neural network symbol detection for millimeter wave communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikola, HI, USA, Dec. 2019, pp. 1–6.

[28] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4694–4708, Oct. 2018.

[29] A. Caciularu and D. Burshtein, "Blind channel equalization using variational autoencoders," 2018, *arXiv:1803.01526*. [Online]. Available: http://arxiv.org/abs/1803.01526

[30] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[31] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014, *arXiv:1409.2574*. [Online]. Available: http://arxiv.org/abs/1409.2574

[32] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. ICML*, Haifa, Israel, Jun. 2010, pp. 399–406.

[33] O. Solomon *et al.*, "Deep unfolded robust PCA with application to clutter suppression in ultrasound," 2018, *arXiv:1811.08252*. [Online]. Available: http://arxiv.org/abs/1811.08252

[34] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Deep algorithm unrolling for blind image deblurring," 2019, *arXiv:1902.03493*. [Online]. Available: http://arxiv.org/abs/1902.03493

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

[36] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO detection with deep learning," 2018, *arXiv:1812.01571*. [Online]. Available: http://arxiv.org/abs/1812.01571

[37] S. Takabe, M. Imanishi, T. Wadayama, and K. Hayashi, "Deep learning-aided projected gradient detector for massive overloaded MIMO channels," 2018, *arXiv:1806.10827*. [Online]. Available: http://arxiv.org/abs/1806.10827

[38] S. Khobahi, N. Naimipour, M. Soltanalian, and Y. C. Eldar, "Deep signal recovery with one-bit quantization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 2987–2991.

[39] L. Liu, C. Yuen, Y. L. Guan, Y. Li, and Y. Su, "Convergence analysis and assurance for Gaussian message passing iterative detector in massive MU-MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6487–6501, Sep. 2016.

[40] L. Liu, Y. Chi, C. Yuen, Y. L. Guan, and Y. Li, "Capacity-achieving MIMO-NOMA: Iterative LMMSE detection," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1758–1773, Apr. 2019.

[41] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 584–588.

[42] H. He, C.-K. Wen, S. Jin, and G. Ye Li, "Model-driven deep learning for joint MIMO channel estimation and signal detection," 2019, *arXiv:1907.09439*. [Online]. Available: http://arxiv.org/abs/1907.09439

[43] J. Guo *et al.*, "Deep neural network-aided Gaussian message passing detection for ultra-reliable low-latency communications," *Future Gener. Comput. Syst.*, vol. 95, pp. 629–638, Jun. 2019.

[44] Z. Zhang, Y. Li, C. Huang, Q. Guo, C. Yuen, and Y. L. Guan, "DNN-aided block sparse Bayesian learning for user activity detection and channel estimation in grant-free non-orthogonal random access," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, Dec. 2019, pp. 12000–12011.

[45] 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, May 2020.

[46] N. Shlezinger, D. Zahavi, Y. Murin, and R. Dabora, "The secrecy capacity of Gaussian MIMO channels with finite memory," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1874–1897, Mar. 2017.

[47] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Data-driven factor graphs for deep symbol detection," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 1–15.

[48] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Inference from stationary time sequences via learned factor graphs," 2020, *arXiv:2006.03258*. [Online]. Available: http://arxiv.org/abs/2006.03258

[49] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code-division multiple-access communications," *IEEE Trans. Commun.*, vol. 38, no. 4, pp. 505–519, Apr. 1990.

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

[51] N. Shlezinger and Y. C. Eldar, "Deep task-based quantization," 2019, *arXiv:1908.06845*. [Online]. Available: http://arxiv.org/abs/1908.06845

[52] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4320–4328.

[53] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, Dec. 2013, pp. 945–948.

[54] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*. New York, NY, USA: Academic, 2010.

[55] H. Wang, C.-K. Wen, and S. Jin, "Bayesian optimal data detector for mmWave OFDM system with low-resolution ADC," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1962–1979, Sep. 2017.

[56] O. Simeone, S. Park, and J. Kang, "From learning to meta-learning: Reduced training overhead and complexity for communication systems," 2020, *arXiv:2001.01227*. [Online]. Available: http://arxiv.org/abs/2001.01227

Rong Fu (Graduate Student Member, IEEE) received the B.S. degree from Beihang University, Beijing, China, in 2012. She is currently pursuing the Ph.D. degree with the Intelligence Sensing Laboratory, Department of Electronic Engineering, Tsinghua University. Her current research interests include radar signal processing, model-based deep learning, and compressive sensing.

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degree in physics and the B.Sc. degree in electrical engineering from Tel-Aviv University (TAU), Tel-Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2002.

She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. She was previously a Professor with the Department of Electrical Engineering, Technion, where she held the Edwards Chair in Engineering. She is also a Visiting Professor at MIT, a Visiting Scientist at the Broad Institute, an Adjunct Professor at Duke University, and was a Visiting Professor at Stanford. Her research interests are in the broad areas of statistical signal processing, sampling theory and compressed sensing, learning and optimization methods, and their applications to biology and optics.

Dr. Eldar was a Horev Fellow of the Leaders in Science and Technology Program at the Technion and an Alon Fellow. She is a member of the Israel Academy of Sciences and Humanities (elected 2017) and a fellow of the EURASIP. She was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is the Editor-in-Chief of the *Foundations and Trends in Signal Processing*, a member of the IEEE Sensor Array and Multichannel Technical Committee, and serves on several other IEEE committees. She has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award (2013), the IEEE/AESS Fred Nathanson Memorial Radar Award (2014), and the IEEE Kiyo Tomiyasu Award (2016). She received the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel and David Jacknow Award for Excellence in Teaching, and the Technion's Award for Excellence in Teaching (two times). She received several best paper awards and best demo awards together with her research students and colleagues, including the SIAM Outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award and the IET Circuits, Devices and Systems Premium Award, and was selected as one of the 50 most influential women in Israel. She was the Co-Chair and Technical Co-Chair of several international conferences and workshops. In the past, she was a Signal Processing Society Distinguished Lecturer, a member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees, and has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, *EURASIP Journal of Signal Processing*, *SIAM Journal on Matrix Analysis and Applications*, and *SIAM Journal on Imaging Sciences*.

Nir Shlezinger (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Ben-Gurion University, Israel, in 2011, 2013, and 2017, respectively. From 2017 to 2019, he was a Post-Doctoral Researcher with Technion. From 2019 to 2020, he was a Post-Doctoral Researcher with the Weizmann Institute of Science. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Ben-Gurion University, Israel. His research interests include communications, information theory, signal processing, and machine learning.