Trainable Projected Gradient Detector for Massive Overloaded MIMO Channels: Data-driven Tuning Approach

Satoshi Takabe∗†, Masayuki Imanishi∗, Tadashi Wadayama∗, and Kazunori Hayashi‡
∗Nagoya Institute of Technology, Aichi, Japan,
s_takabe@nitech.ac.jp, 29414024@stn.nitech.ac.jp, wadayama@nitech.ac.jp
†RIKEN Center for Advanced Intelligence Project, Tokyo, Japan
‡Graduate School of Engineering, Osaka City University, Osaka, Japan, kazunori@eng.osaka-cu.ac.jp

Abstract—The paper presents a deep learning-aided iterative detection algorithm for massive overloaded multiple-input multiple-output (MIMO) systems where the number of transmit antennas \( n \gg 1 \) is larger than that of receive antennas \( m \). Since the proposed algorithm is based on the projected gradient descent method with trainable parameters, it is named as the trainable projected gradient-detector (TPG-detector). The trainable internal parameters like step size parameters can be optimized with standard deep learning techniques such as back propagation and stochastic gradient descent algorithms. This approach referred to as data-driven tuning brings notable advantages of the proposed scheme such as fast convergence. The main iterative process of the TPG-detector consists of matrix-vector product operations that require \( O(mn) \)-time for each iterative step. In addition, the number of trainable parameters in the TPG-detector is independent of the number of antennas \( n \) and \( m \). These features of the TPG-detector lead to its fast and stable training process and reasonable scalability to large systems. The numerical simulations show that the proposed detector achieves comparable detection performance to those of the known algorithms for massive overloaded MIMO channels, e.g., the state-of-the-art IW-SOA V detector, with lower computation cost.

Index Terms—massive MIMO, overloaded MIMO, detection algorithm, deep learning

I. INTRODUCTION

Multiple-input multiple-output (MIMO) systems have attracted great interests because they potentially achieve high spectral efficiency in wireless communications. Recently, as a consequence of high growth of mobile data traffic, massive MIMO is regarded as a key technology in the 5th generation (5G) wireless network standard [1]. In massive MIMO systems, tens or hundreds of antennas are used in the transmitter and the receiver. This fact complicates the detection problem for MIMO channels because the computational complexity of a MIMO detector, in general, increases as the numbers of antennas grow. A practical massive MIMO detection algorithm should possess low computational complexity in addition to reasonable bit error rate (BER) performance.

In a down-link massive MIMO channel with mobile terminals, a transmitter in a base station can have many antennas but a mobile terminal cannot have such a large number of receive antennas because of the restrictions on cost, space limitation, and power consumption. This scenario is known as the overloaded (or underdetermined) MIMO. The data collection by a base station from Internet of Things (IoT) nodes also can be regarded as an (up-link) overloaded MIMO system, because the number of IoT nodes is typically greater than that of antennas at the base station.

Development of an overloaded MIMO detector with computational efficiency and reasonable BER performance is a highly challenging problem because conventional naive MIMO decoders such as the zero-forcing detector or the minimum mean square error (MMSE) detector [2] exhibit poor BER performance for overloaded MIMO channels, and an optimal detection based on the maximum likelihood (ML) criterion with the exhaustive search is evidently computationally intractable.

Several search-based detection algorithms such as slab-sphere decoding [3] and enhanced reactive tabu search (ERTS) [4] have been proposed for overloaded MIMO channels. Though these schemes show excellent detection performance, they are still computationally demanding, and it may prevent us from implementing them into a practical massive overloaded MIMO system. As a computationally efficient approach based on the \( \ell_1 \)-regularized minimization, Fadlallah et al. proposed a detector using a convex optimization solver [5]. Recently, Hayakawa and Hayashi [6] proposed an iterative detection algorithm with practical computational complexity called iterative weighted sum-of-absolute value (IW-SOAV) optimization (see also [7]). The algorithm is based on the SOAV optimization [8] for sparse discrete signal recovery. In addition, the algorithm includes a re-weighting process based on the log-likelihood ratio, which significantly improves the detection performance. The IW-SOAV provides the state-of-the-art BER performance among overloaded MIMO detection algorithms with low computational complexity.

The use of deep neural networks has spread to numerous fields such as image recognition [9], [10] and speech recognition [11] with the progress of computational resources. It also gives a great impact on design of algorithms for wireless communications and signal processing [12], [13]. In [13], for
instance, it is shown that the belief propagation decoder with trainable parameters provides excellent BER performances for several short codes such as BCH codes. Gregor and LeCun first proposed the learned iterative shrinkage-thresholding algorithm (LISTA) [14], which exhibits better recovery performance than that of the original ISTA [15] for sparse signal recovery problems. Recently, a part of the authors proposed the trainable ISTA (TISTA) [16] yielding significantly faster convergence than ISTA and LISTA.

TISTA includes several trainable internal parameters and these parameters are tuned with standard deep learning techniques such as back propagation and stochastic gradient descent (SGD) algorithms. From our research work on TISTA [16] and several additional experiments (an example will be presented in Section II-B), we encountered a phenomenon that the convergence to the minimum value is accelerated with appropriate parameter embedding for several numerical optimization algorithms such as the projected gradient descent method and the proximal gradient method. We call this phenomenon data-driven acceleration of convergence.

Most of known acceleration techniques for gradient descent algorithms such as the momentum methods do not care about the statistical nature of the problems. On the other hand, the data-driven acceleration is obtained by learning the statistical nature of the problem, i.e., stochastic variations on the landscape of the objective functions. The internal parameters controlling the behavior of the algorithm are adjusted to match the typical objective function via training processes. The data-driven acceleration is especially advantageous in implementation of detection algorithms because it reduces the number of iterations without sacrificing the detection performance. This makes the algorithm faster and more power efficient.

The goal of this paper is to propose a novel detection algorithm for massive overloaded MIMO systems, which is called the Trainable Projected Gradient-Detector (TPG-detector). The proposed algorithm is based on the projected gradient descent method with trainable parameters. We have confirmed that data-driven acceleration improves the detection performance and the convergence speed.

Though deep learning architectures for MIMO systems were recently proposed, i.e., deep MIMO detectors (DMDs) in [17], [18] and a TISTA-based MIMO detection algorithm in [19], no deep learning-aided iterative detectors for massive overloaded MIMO channels have been proposed as far as the authors are aware of. Furthermore, an application of data-driven tuning to MIMO detectors has not yet been studied in the related literatures.

This paper is organized as follows. In Section II, we introduce the concept of the data-driven tuning for iterative algorithms and demonstrate it by a simple example. In Section III, we describe settings of massive overloaded MIMO systems. Section IV is the main part of this paper which proposes the TPG-detector for massive overloaded MIMO systems and presents its detection performance comparing with other detection algorithms such as the IW-SOAV. The last section is devoted to a summary of this paper. The Appendix presents a brief review of the IW-SOAV.

II. DATA-DRIVEN TUNING

In this section, we first introduce our key design principle called data-driven tuning for numerical optimization algorithms. A simple example of data-driven tuning is then presented. In the numerical results, we will observe phenomena of data-driven acceleration of convergence for a projected gradient descent algorithm. The trainable algorithm shown in the example will be used as the base of the TPG-detector in Section IV.

A. Basic concept

We here introduce the concept of the data-driven tuning of numerical optimization algorithms, whose origin dates back to the work by Gregor and LeCun [14]. For improving the performance of a numerical iterative optimization algorithm, several trainable parameters can be embedded in the algorithm. By unfolding the iterative process of the numerical optimization algorithm, we have a multilayer signal-flow graph that is similar to a deep neural network. If each component of the signal-flow graph is differentiable, these trainable parameters can be adjusted by standard deep learning techniques. It is crucial to have sufficient training data for this approach. Note that the training data can be randomly generated according to a channel model in the case of the communication problem.

Figure 1 (a) illustrates a signal-flow diagram of an iterative numerical optimization algorithm where Processes A, B, and C are processes whose input/output relationships are expressed with differentiable functions. By unfolding the signal-flow diagram, we obtain a signal-flow graph similar to a multilayer neural network (Fig. 1(b)).

Each process contains trainable parameters that are represented by the black circles in Fig. 1(b). The trainable parameters can control behavior of the processes A, B, and C. Appending a loss function, e.g., the squared loss function, at the end of the unfolded signal-flow graph, we are ready to feed randomly generated training data to the graph. We can apply back propagation and a SGD type parameter update (SGD, RMSprop, Adam, etc.) to optimize the parameters.

B. Example of data-driven tuning

We now discuss a simple example of the data-driven tuning in more detail. As a toy model closely related to the MIMO...
channel, we consider a quadratic programming problem with binary variables.

1) Problem Setting: Let us consider a simple quadratic optimization problem

\[
\text{minimize}_{x \in \{-1, +1\}^n} \frac{1}{2} \|Ax - y\|^2_2, \tag{1}
\]

where \(A \in \mathbb{R}^{n \times n}\) is a given matrix and \(\| \cdot \|_2\) represents the Euclidean norm. We assume that \(y\) is stochastically generated as \(y = Ax + w \in \mathbb{R}^n\) where \(x\) is a vector sampled from \(\{-1, +1\}^n\) uniformly at random and \(w \in \mathbb{R}^n\) consists of i.i.d. Gaussian random variables with zero mean and variance \(\sigma^2\). The optimization problem is essentially the same as the ML estimation rule for the Gaussian linear vector channel.

Since solving this problem is known as an NP-hard problem in general, we need to solve the problem approximately. We here exploit a variant of the projected gradient (PG) algorithm to solve (1) approximately for large systems. The PG algorithm can be described by the recursive formula:

\[
\begin{align*}
    r_t &= s_t + \gamma A^T(y - As_t), \quad t = 1, \ldots, T \text{ and } \tanh(\cdot) \text{ is calculated element-wisely.} \tag{2}
    s_{t+1} &= \tanh(r_t), \quad \text{where } t = 1, \ldots, T \text{ and } \tanh(\cdot) \text{ is calculated element-wisely.} \tag{3}
\end{align*}
\]

The initial value is set to \(s_1 = \mathbf{0}\).

The PG algorithm consists of two computational steps for each iteration. In the gradient descent step (2), a search point moves to the opposite direction to the gradient vector of the objective function, i.e., \(\nabla \frac{1}{2} \|Ax - y\|^2_2 = -A^T(y - Ax)\). The parameter \(\gamma\) controls the step size causing critical influence on the convergence behavior. In the projection step (3), soft-projection based on the hyperbolic tangent function is applied to the search point to obtain a new search point nearly rounded to binary values. Precisely speaking, the projection step is not the projection onto the binary symbols \(\{-1, +1\}\). This is because the true projection to discrete values results in insufficient convergence behavior in a minimization process (see also the discussion in Section II-B4). The parameter \(\xi\) controls the softness of the soft projection. Note that this type of nonlinear projection has been commonly used in several iterative multiuser detection algorithms such as the soft parallel interference canceller [20].

2) Trainable PG algorithm: According to the data-driven tuning framework, we can embed trainable parameters into the PG algorithm. The trainable PG (TPG) algorithm is based on the recursion

\[
\begin{align*}
    r_t &= s_t + \gamma_t A^T(y - As_t), \quad t = 1, \ldots, T \text{ and } \tanh(\cdot) \text{ is calculated element-wisely.} \tag{4}
    s_{t+1} &= \tanh(\xi r_t), \quad \text{with } s_1 = 0. \tag{5}
\end{align*}
\]

The trainable parameters \(\{\gamma_t\}_{t=1}^T\) play a key role in the gradient descent step by adjusting its step size adaptively. In the following discussion, the parameter \(\xi\) is treated as a fixed hyper parameter in Section II-B3 and it is treated as a trainable parameter in Section II-B4.

As described above, the parameters \(\{\gamma_t\}_{t=1}^T\) are optimized by the standard mini-batch training. The \(i\)th training data \(d^i \triangleq (x^i, y^i)\) is generated randomly, i.e., \(x^i \in \{-1, +1\}^n\) is generated uniformly at random and the corresponding \(y^i\) is then generated according to \(y^i = Ax^i + w^i\) with a given \(A\). A mini-batch consists of \(D\) training data \(D \triangleq \{d^1, d^2, \ldots, d^D\}\). In the following experiment, a matrix \(A\) is randomly generated for each mini-batch. Each element of \(A\) follows the Gaussian distribution with mean 0 and variance 1.

For each round of a training process, we feed these mini-batches to the TPG algorithm to minimize the squared loss function

\[
L(\Theta_t) \triangleq D^{-1} \sum_{d \in D} \|x^i - \hat{x}_t(y^i)\|^2_2, \tag{6}
\]

where \(\hat{x}_t(y) \triangleq s_t \cup \gamma_t\) is the output of the TPG algorithm with \(t\) iterations and \(\Theta_t \triangleq \{\gamma_1, \ldots, \gamma_t\}\) (or \(\Theta_t \triangleq \{\gamma_1, \ldots, \gamma_t \cup \xi\}\)) is a set of trainable parameters up to the \(t\)th round. A back propagation process evaluates the gradient \(\nabla L(\Theta_t)\) and it is used for updating the set of parameters \(\Theta_t\) by a SGD type algorithm such as the Adam optimizer [21].

It should be remarked that a simple single-shot training for a whole process by letting \(t = D\) does not work well because the vanishing gradient phenomenon prevents appropriate parameter updates due to the fact that the derivative of the soft projection function (5) becomes nearly zero almost everywhere. Figure 2 shows the gradient of trainable parameters \(\{\gamma_t\}_{t=1}^T\) in the TPG algorithm. We find that the gradient vanishes as the iteration index \(t\) becomes small. It results in insufficient training by the single-shot training (see Fig. 3). In order to avoid the vanishing gradient phenomenon, we use an alternative approach, i.e., incremental training as in TISTA [16]. In the incremental training, the parameters \(\{\gamma_t\}_{t=1}^T\) are sequentially trained from \(\Theta_1\) to \(\Theta_T\) in an incremental manner.

The details of the incremental training is as follows. At first, \(\Theta_1\) is trained by minimizing \(L(\Theta_1)\). After finishing the training of \(\Theta_1\), the values of trainable parameters in \(\Theta_1\) are copied to the corresponding parameters in \(\Theta_2\). In other words, the results of the training for \(\Theta_1\) are taken over to \(\Theta_2\) as the initial values. For each round of the incremental training which is called a generation, \(K\) mini-batches are processed.

3) Data-driven acceleration: We show the numerical demonstration of the TPG algorithm. Here, we treat the step...
size parameters \( \{\gamma_t\}_{t=1}^T \) as trainable parameters but \( \xi \) is treated as a hyper parameter. In the experiment, the dimension of matrix \( A \) is set to \( n = 1000 \) and the noise variance is fixed to \( \sigma^2 = 4.0 \). The number of iterations of the TPG algorithm is \( T = 20 \). We performed two types of training processes of the TPG algorithm to measure the effect of incremental training. In the training process of TPG with incremental training, we used \( K = 100 \) mini-batches per generation. The mini-batch size was set to \( D = 200 \) and Adam optimizer with learning rate 0.0002 was used for the parameter updates. In the training process of TPG without incremental training (named “TPG-noINC” in Fig. 3), we used \( K = 2000 \), \( D = 200 \) and Adam optimizer with learning rate 0.002. The initial values of the trainable parameters were given by \( \gamma_t = 1.0 \times 10^{-4} \) \( (t = 1, \ldots, T) \). In this experiment, the softness parameter \( \xi \) was fixed to 8.0 for the TPG algorithm.

Figure 3 shows the mean squared error (MSE) as a function of iteration steps of the plain PG algorithm based on (2), (3) \( (\xi = 6.0, \gamma = 6.5 \times 10^{-4}) \) and the TPG algorithms based on (4), (5) (with/without incremental training). The MSE after \( t \) iterations is defined by \( 10 \log_{10} (\mathbb{E}[\|\tilde{x} - \hat{x}(y)\|^2]/n) \) (dB) and it was estimated from \( 10^4 \) random samples of \( A, y \) and \( x \). The parameter \( \gamma = 6.5 \times 10^{-4} \) in the plain PG algorithm is the optimal value for \( T = 20 \) (see also Fig. 4).

From Fig. 3, we can observe that the TPG algorithm provides much smaller MSEs than those of the plain PG algorithm. The MSE of TPG achieves \(-80\) dB at \( t = 8 \) but the PG yields the smaller MSE after \( t = 19 \). Namely, TPG shows much faster convergence and it implies that the parameter tuning drastically improves the convergence speed. This is an example of the data-driven acceleration of convergence by data-driving tuning. The effect of the incremental training can be confirmed by comparing the MSEs of the TPG algorithms with/without incremental training. The MSE curve of TPG-noINC is almost flat, indicating that the parameter tuning is not successful due to the vanishing gradient in the case of the TPG algorithm without incremental training.

In Fig. 4, we show the relationships between the step size parameter \( \gamma \) and the MSE performance of the plain PG algorithm with \( \xi = 6.0 \). It can be observed that the parameter \( \gamma \) must be selected carefully to obtain appropriate convergence. In other words, the sweet spot of \( \gamma \) is relatively narrow, i.e., close neighborhood of \( 6.5 \times 10^{-4} \) is only allowable choice for achieving \(-100\) dB at \( T = 200 \). This means that optimization of the step size is critical even for the plain PG algorithm. In addition, the TPG algorithm achieves the lower MSE performance (around \(-130\) dB) which cannot be achieved by the plain PG algorithm. This fact implies that embedding an independent step size parameter into each iteration step provides substantial improvement on the quality of the solution.

4) Effect of softness parameter: We further discuss the effect of the softness parameter \( \xi \) in (5) on the MSE performance. Figure 5 shows the MSE curves of TPG algorithms with different \( \xi \)’s and the TPG algorithm with trainable \( \xi \). The setting of the experiment is the same as the previous one. As described above, the projection of the TPG algorithm is the soft-projection function instead of the hard-projection function corresponding to the \( \xi \rightarrow \infty \) limit. The results show that the large fixed \( \xi \) is not appropriate in terms of the MSE performance. On the other hand, the TPG algorithm with small fixed \( \xi \) also shows poor MSE performance. It is thus crucial to tune not only the step size parameter \( \{\gamma_t\}_{t=1}^T \) but also the softness parameter \( \xi \) to pull out the full performance of the TPG algorithm. The curve with the label “TPG (\( \xi \) trained)” represents the MSE of TPG with trainable \( \xi \). In the experiment, we used \( K = 10000 \) due to the slow convergence of \( \xi \). In Fig. 5, we can see that it outperforms other TPG algorithms with fixed \( \xi \).

III. OVERLOADED MIMO CHANNELS

The section describes the MIMO channel model and introduces several definitions and notations. The numbers of transmit and receive antennas are denoted by \( m \) and \( n \), respectively. We mainly consider the overloaded MIMO scenario in this paper where \( m < n \) holds. It is also assumed that the transmitter does not use precoding and that the receiver perfectly knows the channel state information, i.e., the channel matrix.

Let \( \hat{x} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]^T \in \tilde{S}^n \) be a vector which consists of transmitted symbols \( \hat{x}_j \) (\( j = 1, \ldots, n \)) from the \( j \)-th antenna.
The symbol $\tilde{S} \subset \mathbb{C}$ represents a symbol alphabet, i.e., a signal constellation. Similarly, $\tilde{y} \triangleq [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m]^T \in \mathbb{C}^m$ denotes a vector composed of received symbols $\tilde{y}_i$ $(i = 1, \ldots, m)$ by the $i$th antenna. A flat Rayleigh fading channel is assumed here and the received symbol vector $\tilde{y}$ is then given by

$$\tilde{y} = \tilde{H}\tilde{x} + \tilde{w}, \quad (7)$$

where $\tilde{w} \in \mathbb{C}^m$ consists of complex Gaussian random variables with zero mean and covariance $\sigma_w^2 I$. The matrix $\tilde{H} \in \mathbb{C}^{m \times n}$ is a channel matrix whose $(i,j)$ entry $\tilde{h}_{i,j}$ represents a path gain from the $j$th transmit antenna to the $i$th receive antenna. Each entry of $\tilde{H}$ independently follows the complex circular Gaussian distribution with zero mean and unit variance. For the following discussion, it is convenient to derive an equivalent channel model defined over $\mathbb{R}$, i.e.,

$$y = Hx + w, \quad (8)$$

where

$$\begin{bmatrix} y \\ w \end{bmatrix} \triangleq \begin{bmatrix} \text{Re}(\tilde{y}) \\ \text{Im}(\tilde{y}) \end{bmatrix} \in \mathbb{R}^M, \quad H \triangleq \begin{bmatrix} \text{Re}(\tilde{H}) & -\text{Im}(\tilde{H}) \\ \text{Im}(\tilde{H}) & \text{Re}(\tilde{H}) \end{bmatrix}, \quad (9)$$

$$\begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} \text{Re}(\tilde{x}) \\ \text{Im}(\tilde{x}) \end{bmatrix} \in \mathbb{R}^N, \quad \begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} \text{Re}(\tilde{w}) \\ \text{Im}(\tilde{w}) \end{bmatrix} \in \mathbb{R}^M,$$

and $(N, M) \triangleq (2n, 2m)$. The signal set $\tilde{S}$ is the real counter part of $\tilde{S}$. The matrix $\tilde{H} \in \mathbb{R}^{M \times N}$ is converted from $H$. Similarly, the noise vector $w$ consists of i.i.d. random variables following the Gaussian distribution with zero mean and variance $\sigma_w^2/2$. Signal-to-noise ratio (SNR) per receive antenna is then represented by

$$\text{SNR} \triangleq \frac{E_s}{N_0} = \frac{2n}{\sigma_w^2}, \quad (10)$$

where $E_s \triangleq \mathbb{E}[|\tilde{H}\tilde{x}|^2]/m$ stands for the signal power per receive antenna and $N_0 \triangleq \sigma_w^2$ stands for the noise power per receive antenna. Throughout the paper, we assume the QPSK modulation format, i.e., $\tilde{S} \triangleq \{1 + j, -1 + j, -1 - j, 1 - j\}$, which is equivalent to the BPSK modulation $\tilde{S} \triangleq \{-1, +1\}$.

### IV. Trainable Projected Gradient (TPG)-Detector

The proposed algorithm called TPG-detector is based on the TPG algorithm introduced in Section II-B. We first describe the detail of the TPG-algorithm and discuss its time complexity. Then, we will show some numerical results which mainly compare the detection performance of the TPG-detector with that of other baseline algorithms for massive overloaded MIMO systems. In addition, the detection performance for non-massive non-overloaded MIMO systems will be presented, which indicates that the TPG-detector is a promising MIMO detector not only for massive overloaded systems but also for non-massive non-overloaded systems with reasonably low computational complexity.

#### A. Details of TPG-detector

The ML estimation rule for the MIMO channel defined above is given by

$$\hat{x} \triangleq \arg\min_{x \in \{-1, +1\}^N} \|Hx - y\|_2^2. \quad (11)$$

This problem is a non-convex problem and finding the global minimum is computationally intractable for a large scale problem. The TPG-detector is based on the TPG algorithm to solve the above non-convex problem approximately. The process of the TPG-detector is described by the following recursive formulas:

$$r_t = s_t + \gamma_t W(y - Hs_t), \quad (12)$$

$$s_{t+1} = \tanh \left( \frac{r_t}{\|\theta_t\|} \right), \quad (13)$$

where $t(=1, \ldots, T)$ represents the index of an iteration step (or layer) and we set $s_1 = 0$ as the initial value. This algorithm estimates a transmitted signal $x$ from the received signal $y$ and outputs the estimate $\hat{x} = s_{T+1}$ after $T$ iteration steps.

The steps (12) and (13) correspond to the gradient descent step and to the projection step, respectively, as described in Section II-B. The matrix $W$ in the gradient step (12) is the linear MMSE (LMMSE)-like matrix defined by

$$W \triangleq HH^T(\alpha I + \beta H H^T)^{-1}, \quad (14)$$

where $\alpha \in \mathbb{R}$ is a trainable parameter. The matrix (14) also appears in the solution of a linear regression problem with a quadratic regularization term. Precisely speaking, the matrix $W$ should be $H^T$ as in (4) for achieving the gradient descent process for the quadratic objective function in (11). However, we adopt the modification inspired by [22] because this modification improves the BER performance of the proposed scheme (an experimental evidence will be shown in Section V-B1). This modification is especially effective when the matrix $H$ is ill-conditioned, i.e., the condition number of $H$ is large. The LMMSE-like matrix (14) changes an ill-conditioned matrix to a reasonably well-conditioned matrix. Since it is critical to optimize $\alpha$ to achieve a reasonable detection performance, the parameter $\alpha$ is optimized in a training process. As in the case in (5), we use the hyperbolic tangent function for the soft projection.
B. Trainable parameters

The trainable parameters of the TPG-detector are $2T + 1$ real scalar variables $\alpha$, $\mathcal{\{\gamma_i\}}_{i=1}^{T}$, and $\mathcal{\{\theta_i\}}_{i=1}^{T}$. The parameters $\mathcal{\{\gamma_i\}}_{i=1}^{T}$ in the gradient step control the step size of a move of the search point. In order to achieve fast convergence, appropriate setting of these step size parameters is of critical importance as described in Section II-B. It should be remarked that similar trainable parameters are also introduced in the structure of TISTA [16]. The parameters $\mathcal{\{\theta_i\}}_{i=1}^{T}$ control the softness of the soft projection in (13). Different from the TPG algorithm, the trainable parameters depend on the iteration index $t$ to increase the degree of freedom of trainable parameters in the soft-projection functions. The parameter $\alpha$ adjusts the degree of compensation for an ill-conditioned matrix $H$. Apart from $\mathcal{\{\gamma_i\}}_{i=1}^{T}$ and $\mathcal{\{\theta_i\}}_{i=1}^{T}$, the TPG-detector uses the same parameter $\alpha$ through all the iteration steps. It is crucial to reduce the computational cost to execute the TPG-detector, which will be discussed in the next subsection.

One of the advantages of the TPG-detector is that the number of trainable parameters is small, i.e., $O(T)$, and it leads to fast and stable training processes. The number of trainable parameters of the TPG-detector is constant to the number of antennas $n$ and $m$ though the DMD [17] contains $O(n^2T)$ parameters in $T$ layers.

C. Time complexity

The computational complexity of the TPG-detector per iteration is $O(mn)$ because calculating the vector-matrix products $Hs_t$ and $W(y - Hs_t)$ takes $O(mn)$ computational steps. We need to calculate the LMMSE-like matrix $W$ with $O(m^3)$ computational steps because the calculation involves an matrix inversion. The evaluation of $W$ is required only when $H$ changes, i.e., the matrix inversion is not needed for each iteration of a TPG-detector if $H$ is constant during the process. A TISTA-based MIMO detection algorithm proposed in [19] also uses the LMMSE matrix as a linear estimator. It should be remarked that, in the TISTA-based MIMO detector [19], the LMMSE matrix must be recalculated for each iteration and it thus requires $O(m^3)$ computational steps for execution. This is one of the critical differences between our algorithm and the TISTA-based MIMO detector.

D. Training process

The TPG-detector is trained based on the incremental training described in Section II-B2. The training data is generated randomly according to the channel model (8) with fixed variance $\sigma_w^2$ corresponding to a given SNR. As described in Section III, we assume a practical situation in which a channel matrix $H$ is a random variable. According to this assumption, a matrix $H$ is randomly generated for each mini-batch in a training process of the TPG-detector.

V. NUMERICAL RESULTS

In this section, we show the detection performance of the TPG-detector and compare it to that of other algorithms such as the IW-SOAV which is known as one of the most efficient iterative algorithms for massive overloaded MIMO systems.

A. Experimental setup

A transmitted vector $x$ is generated uniformly at random. The BER is then evaluated for a given SNR. We use randomly generated channel matrices for BER estimation.

The TPG-detector was implemented by PyTorch 0.4.0 [23]. In this paper, a training process is executed with $T = 50$ rounds using the Adam optimizer [21]. To calculate the BER of the TPG-detector, a sign function $\text{sign}(z)$ which takes $-1$ if $z \leq 0$ and $1$ otherwise is applied to the final estimate $s_{T+1}$ in an element-wise manner.

As the baselines of detection performance, we use the ERTS [4], IW-SOAV [6], and the standard MMSE detector. The ERTS is a heuristic algorithm based on a tabu search for overloaded MIMO systems. The parameters of ERTS is based on [4]. The IW-SOAV is a double loop algorithm for massive overloaded MIMO systems (see Appendix for the brief review). Its inner loop is the W-SOAV optimization recovering a signal using a proximal operator. Each round of the W-SOAV takes $O(mn)$ computational steps, which is comparable to that of the TPG-detector. After finishing an execution of the inner loop with $K_{it}$ iterations, several parameters are then updated in a re-weighting process based on a tentative recovered signal. This procedure is repeated $L$ times in the outer loop. The total number of steps of the IW-SOAV is thus $K_{it}L$. In the following, we use the simulation results in [6] with $K_{it} = 50$.

B. Main Results

1) Selection of matrix $W$ in gradient step: As described above, we can choose matrix $W$ in the gradient step (12). The selection of the matrix will affect the detection performance of algorithms as shown in the orthogonal AMP [22]. Before we show the detection performance of the proposed TPG-detector, we numerically test the effect of the selection of $W$. Figure 7 shows the detection performance of the TPG-detector with a different choice of $W$. We examined three types of matrices: the matched filter matrix (MF), pseudo-inverse matrix (PINV), and LMMSE-like matrix (LMMSE). We find that the LMMSE-like matrix outperforms other choices of $W$ in a wide range of SNR. We thus select the LMMSE-like matrix as $W$ to treat an ill-conditioned matrix $H$ in the proposed TPG-detector. This selection is also effective for the non-massive non-overloaded case as shown in Section V-C.
detection algorithms in a large margin when SNR is larger than 10 dB. The label “MF” stands for the matched filter which corresponds to the gradient of the original problem (11), i.e., $W = H^T (H H^T)^{-1}$, and “PINV” represents the pseudo-inverse matrix $W \triangleq H^T (H H^T)^{-1}$, and “LMMSE” is the LMMSE-like matrix (14).

For $(n, m) = (50, 64)$ in Fig. 8, ERTS outperforms other detection algorithms in a large margin when SNR is larger than 10 dB. It should be remarked that ERTS needs much more time complexity (around several orders of magnitudes) than that of the IW-SOAV (see Fig. 7 in [6]). Comparing the TPG-detector with the IW-SOAV, we find that the TPG-detector overwhelms the IW-SOAV ($L = 1$) and shows BER performance close to the IW-SOAV ($L = 5$) when SNR is below 20 dB. Note that the computational cost for executing the TPG-detector with $T = 50$ is almost comparable to that of the IW-SOAV ($L = 1$). The IW-SOAV ($L = 5$) requires 250 iterations that is 5 times as large as the number of required iterations for the TPG-detector with $T = 50$. It implies that the TPG-detector achieves considerably good detection performance with relatively small computational cost.

For $(n, m) = (100, 64)$ in Fig. 9, ERTS detector shows the best BER performance in a middle SNR region where SNR is between 10 dB and 18 dB but the BER curve of ERTS is saturated after 20 dB. The TPG-detector and the IW-SOAV ($L = 1$ and $L = 5$) outperform ERTS in a high SNR regime. In such a regime, the TPG-detector exhibits the BER performance superior to that of the IW-SOAV ($L = 1$) for the entire range of SNR, i.e., the TPG-detector achieves approximately 5 dB gain at BER $= 10^{-4}$ over the IW-SOAV ($L = 1$). More interestingly, the BER performance of the TPG-detector is fairly close to that of the IW-SOAV ($L = 5$). For example, with SNR = 20 dB, the BER estimate of the TPG-detector is $6.8 \times 10^{-5}$ whereas that of the IW-SOAV ($L = 5$) is $4.3 \times 10^{-5}$.

Figure 10 shows the BER performance for $(n, m) = (150, 96)$. In this case, ERTS shows poor BER performance that cannot achieve BER smaller than $10^{-3}$ for any SNR. The TPG-detector successfully recovers transmitted signals with lower BER than that of the IW-SOAV ($L = 1$). It again achieves about 5 dB gain against the IW-SOAV ($L = 1$) at BER $= 10^{-5}$. Although the IW-SOAV ($L = 5$) shows considerable performance improvements in this case, the gap between the curves of the TPG-detector and the IW-SOAV ($L = 5$) is only 2 dB at BER $= 10^{-5}$.

**2) BER performance:** We present the BER performance of each detector as a function of SNR for $(n, m) = (50, 32), (100, 64),$ and $(150, 96)$ in Figs. 8-10, respectively. All the results show that the MMSE detector fails to detect transmitted signals reliably (BER $\approx 10^{-1}$) because the system is underdetermined.

For $(n, m) = (50, 32)$ in Fig. 8, ERTS outperforms other detection algorithms in a large margin when SNR is larger than 10 dB. It should be remarked that ERTS needs much more time complexity (around several orders of magnitudes) than that of the IW-SOAV (see Fig. 7 in [6]). Comparing the TPG-detector with the IW-SOAV, we find that the TPG-detector overwhelms the IW-SOAV ($L = 1$) and shows BER performance close to the IW-SOAV ($L = 5$) when SNR is below 20 dB. Note that the computational cost for executing the TPG-detector with $T = 50$ is almost comparable to that of the IW-SOAV ($L = 1$). The IW-SOAV ($L = 5$) requires 250 iterations that is 5 times as large as the number of required iterations for the TPG-detector with $T = 50$. It implies that the TPG-detector achieves considerably good detection performance with relatively small computational cost.

For $(n, m) = (100, 64)$ in Fig. 9, ERTS detector shows the best BER performance in a middle SNR region where SNR is between 10 dB and 18 dB but the BER curve of ERTS is saturated after 20 dB. The TPG-detector and the IW-SOAV ($L = 1$ and $L = 5$) outperform ERTS in a high SNR regime. In such a regime, the TPG-detector exhibits the BER performance superior to that of the IW-SOAV ($L = 1$) for the entire range of SNR, i.e., the TPG-detector achieves approximately 5 dB gain at BER $= 10^{-4}$ over the IW-SOAV ($L = 1$). More interestingly, the BER performance of the TPG-detector is fairly close to that of the IW-SOAV ($L = 5$). For example, with SNR = 20 dB, the BER estimate of the TPG-detector is $6.8 \times 10^{-5}$ whereas that of the IW-SOAV ($L = 5$) is $4.3 \times 10^{-5}$.

Figure 10 shows the BER performance for $(n, m) = (150, 96)$. In this case, ERTS shows poor BER performance that cannot achieve BER smaller than $10^{-3}$ for any SNR. The TPG-detector successfully recovers transmitted signals with lower BER than that of the IW-SOAV ($L = 1$). It again achieves about 5 dB gain against the IW-SOAV ($L = 1$) at BER $= 10^{-5}$. Although the IW-SOAV ($L = 5$) shows considerable performance improvements in this case, the gap between the curves of the TPG-detector and the IW-SOAV ($L = 5$) is only 2 dB at BER $= 10^{-5}$.

**3) System-size dependency:** In Fig. 11, we show the BER performance of the TPG-detector and IW-SOAV ($L = 1$) as a function of the number of antennas $n$ with the rate $m/n = 0.6$ fixed. The gap of their BER performances is especially large for SNR= 20 dB. We also find that the gain of the TPG-detector increases as $n$ grows though these algorithms have
Theoretical treatments for providing reasonable interpretation expected to accelerate the convergence of the signal recovery.

The parameter $damping$ amplitude similar to that observed in TISTA [16].

when SNR $\gamma$ is non-monotonic unlike the IW-SOAV, i.e, it increases when SNR $\gamma < 7.5$ dB. The parameter corresponding to the parameter $\alpha$ in the IW-SOAV should be chosen in advance by numerical simulations. The learning process in the TPG-detector easily tunes the parameter $\alpha$ in addition to other trainable parameters.

the same computational costs. It is confirmed that the TPG-detector outperforms low-complexity algorithms especially in the massive overloaded MIMO channels.

4) Trained parameters: Figure 12 displays the learned parameters $\{\gamma_t\}_{t=1}^{T}$, and $\{\theta_i\}_{t=1}^{T}$ of the TPG-detector after a training process as a function of the iteration index $t (= 1, \ldots, T)$. We find that they exhibit a zigzag shape with damping amplitude similar to that observed in TISTA [16].

The parameter $\gamma_t$, the step size of a linear estimator, is expected to accelerate the convergence of the signal recovery. Theoretical treatments for providing reasonable interpretation on these characteristic shapes of the learned parameters are interesting open problems.

The trained values of $\alpha$ for different SNR values are shown in Fig. 13. We find that the parameter $\alpha$ is tuned depending on the value of SNR. In particular, the trained value decreases when SNR $\leq 7.5$ dB. This tendency is similar to that of a parameter related to the parameter $\alpha$ in the IW-SOAV [6] using the LMMSE-like matrix. On the other hand, the trained value is non-monotonic unlike the IW-SOAV, i.e, it increases when SNR $\leq 7.5$ dB. The parameter corresponding to the parameter $\alpha$ in the IW-SOAV should be chosen in advance by numerical simulations. The learning process in the TPG-detector easily tunes the parameter $\alpha$ in addition to other trainable parameters.

5) Computation time: We finally discuss the scalability of the TPG-detector to show the required computation time for training. The empirical execution time of training process of the TPG-detector is measured by using a PC with GPU NVIDIA GerForce GTX 1080 and Intel Core i7-6700K CPU 4.0 GHz with 8 cores. Table I presents the execution time of training processes with different $n$. Even for the case $(n, m) = (150, 96)$, we need only 20 minutes for training the TPG-detector and this results indicates that the training process of the TPG-detector is reasonably practical for fairly large systems.

| $(n, m)$ | $(50, 32)$ | $(100, 64)$ | $(150, 96)$ |
|---------|------------|-------------|-------------|
| Exec. time (sec) | 638.7 | 853.2 | 1203 |

C. Performance for non-massive non-overloaded MIMO systems

Although the main target of the TPG-detector is massive overloaded MIMO systems, the proposed algorithm can be applied to a non-massive non-overloaded MIMO systems as well. In this subsection, we will study the detection performance of the TPG-detector for such a MIMO channel with $(n, m) = (5, 5)$.

Figure 14 presents BER performance curves for several detection algorithms. In Fig.14, the label “ML” stands for the ML detection algorithm that provides the smallest BER. The label “MMSE” and “GIGD” represents the MMSE detector and a belief propagation-based detection algorithm [24], respectively. In addition to these detectors, the BER curves of the IW-SOAV ($L = 1$ and $L = 5$) are also included in Fig.14.

The MMSE detector requires the least computational complexity but it gives the largest BER from SNR = 0 to 20 dB. Although the BER performance of GIGD is competitive with that of the TPG-detector and the IW-SOAV ($L = 5$) up to 12.5 dB, the BER of GIGD is saturated to a constant value when SNR $> 12.5$ dB. Comparing the TPG-detector with the IW-SOAV ($L = 1$), we find that the TPG-detector shows superior BER performance as observed in the massive overloaded MIMO systems. The gaps between the BER curves.
of the TPG-detector and the IW-SOAV \((L = 1)\) is about 6 dB at BER \(10^{-3}\). It can be also observed that the TPG-detector and IW-SOAV \((L = 5)\) shows almost the same BER performance which is the best except for the ML detection. Note that the IW-SOAV \((L = 5)\) requires much larger computational complexity compared with the TPG-detector. This experimental result indicates that the TPG-detector is also a promising iterative algorithm for non-massive non-overloaded MIMO systems.

\section*{VI. CONCLUSION}

In this paper, we proposed the TPG-detector, a deep learning-aided iterative decoder for massive overloaded MIMO channels. It is based on the concept of the data-driven tuning with standard deep-learning techniques. The TPG-detector contains two trainable parameters for each layer: \(\gamma_i\) controlling a step size of the gradient descent step and \(\theta_i\) controlling the softness of the soft projection. In the soft projection, the parameter \(\alpha\) in the LMMSE-like matrix \(W\) (14) is also optimized in a training process. The total number of the trainable parameters in \(T\) layers is thus \(2T + 1\), which is significantly smaller than those used in the previous studies such as [17], [18]. This fact promotes fast and stable training processes for the TPG-detector.

The computational complexity of the TPG-detector is \(O(mn)\) per iteration because a matrix inversion is not required for each iteration. A TISTA-based MIMO detector [19] needs a matrix inversion for each iteration and it needs \(O(m^2)\) time complexity. In terms of the time complexity, it could be said that the TPG-detector is more scalable for massive MIMO systems.

The numerical simulations show that the TPG-detector outperforms the state-of-the-art IW-SOAV \((L = 1)\) by a large margin and achieves a comparable detection performance to the IW-SOAV \((L = 5)\). The TPG-detector therefore can be seen as a promising iterative detector for massive overloaded MIMO channels providing an excellent balance between a low computational cost and a reasonable detection performance.

\section*{APPENDIX}

\textbf{BRIEF REVIEW OF IW-SOAV}

Here, we give a brief review of the IW-SOAV detector. The IW-SOAV is an effective iterative detection algorithm for massive overloaded MIMO systems proposed in [6], [7]. It is based on a valiant of the Douglas-Rachford algorithm [25] which solves the following weighted SOAV (W-SOAV) optimization problem:

\[
\hat{s} \triangleq \arg\min_{z \in \mathbb{R}^n} \left( \frac{1}{2} \sum_{j=1}^{2n} w_j^+ |z_j - 1| + \frac{1}{2} \sum_{j=1}^{2n} w_j^- |z_j + 1| \right. \\
\left. + \frac{\alpha}{2} \|y - Hz\|_2^2 \right),
\]

where \(z_j (j = 1, \ldots, 2n)\) is the \(j\)th element of \(z\) and \(\alpha (\geq 0)\) is a constant. Here, we assume that each symbol \(x_j\) in the transmitted signal \(x\) is an independent random variable which takes \(1\) w.p. \(w_j^+\) and \(-1\) w.p. \(w_j^- \triangleq 1 - w_j^+\).

The IW-SOAV repeats the following procedures: (i) estimation of \(w_j^+\) based on the detected signal and (ii) detection of transmitted signal by solving the W-SOAV optimization (15). The IW-SOAV is thus a double-loop algorithm.

In the outer loop corresponding to procedure (i), the algorithm approximately estimates \(\{w_j^+\}\) of each transmitted symbol. The estimation is based on the approximate log likelihood ratio which reads

\[
\hat{\Lambda}_j = \sum_{i=1}^{2m} \frac{2h_{i,j} (y_i - (\hat{\mu}_i - h_{i,j} \hat{s}_j'))}{\hat{\sigma}_j^2 - h_{i,j}^2 (1 - \hat{s}_j'^2)},
\]

where \(h_{i,j}\) is the \((i, j)\) element of matrix \(H\). \(\hat{s}'\) represents a clipped signal of \(\hat{s}\), i.e., \(\hat{s}'_j (j = 1, \ldots, 2n)\) takes \(-1\) if \(\hat{s}_j < -1\), \(1\) if \(\hat{s}_j > 1\), and \(s_j\) otherwise. In addition, we define

\[
\hat{\mu}_i \triangleq \sum_{k=1}^{2n} h_{i,k} \hat{s}_k', \\
\hat{\sigma}_j^2 \triangleq \sum_{k=1}^{2n} h_{i,k}^2 (1 - \hat{s}_j'^2) + \frac{\sigma_0^2}{2},
\]

for \(i = 1, \ldots, 2m\). Then, the weight \(w_j^+\) is calculated by

\[
w_j^+ = \frac{e^{\hat{\Lambda}_j}}{1 + e^{\hat{\Lambda}_j}}.
\]

In the inner loop corresponding to procedure (ii), the algorithm solves the W-SOAV optimization problem with an iterative process defined by the following recursive formula:

\[
z_t = (I + \alpha \gamma H^T H)^{-1} (r_t + \alpha \gamma H^T y) \\
r_{t+1} = r_t + \theta_t (\phi_\gamma (2z_t - r_t) - z_t),
\]

where \(t = 1, \ldots, K_{itr}\) is an index of the iteration step, \(\theta_t \in [\epsilon, 2 - \epsilon]\) is a constant, and \(\phi_\gamma : \mathbb{R}^N \rightarrow \mathbb{R}^N\) is a component-wise function whose \(j\)th element \([\phi_\gamma (z)]_j\) is defined by

\[
[\phi_\gamma (z)]_j \triangleq \begin{cases} 
  z_j + \gamma & (z_j < -1 - \gamma) \\
  -1 & (\neg \gamma < z_j < -1 - d_j \gamma) \\
  z_j + d_j \gamma & (-1 - d_j \gamma \leq z_j < -1 - d_j \gamma), \\
  1 & (1 - d_j \gamma < z_j < 1 + \gamma) \\
  z_j - \gamma & (1 + \gamma \leq z_j) \end{cases}
\]

Fig. 14. BER performance for \((n, m) = (5, 5)\). The number of iterations of the TPG-detector is \(T = 50\).
with $d_j \triangleq w_j^+ - w_j^-$. The parameters $\gamma > 0$, $\epsilon \in (0, 1)$, and the initial value $r_0 \in \mathbb{R}^{2n}$ can be set arbitrary. In this W-SOAV optimizer, the transmitted symbol is detected as $\hat{x} = z_{K_{itr} + 1}$ after $K_{itr}$ iteration steps. The IW-SOAV starts with $\hat{\mathbf{s}} = \mathbf{0}$ and repeats $L$ outer loops with $K_{itr}$ inner loops. When all loops are finished, the sign function $\text{sgn}(\cdot)$ is applied to the output $\hat{x}$ in an element-wise manner. The parameter $\alpha$ is to be fixed appropriately depending on SNR. In numerical experiments in Section V, we used $r_0 = 0$, $\epsilon = 0$, $\gamma = 1$, and $\theta_1 = 1.9$ ($t = 1, \ldots, K_{itr}$), and chose $\alpha$ as in [6].

The computational cost of each iteration of the IW-SOAV is $O(mn)$. Although it contains a matrix inversion which takes $O(m^3)$ computational steps, it can be computed in advance. Since the total number of inner and outer loops is $K_{itr}L$, the total computational cost of the IW-SOAV is $O(K_{itr}Lmn)$.

ACKNOWLEDGEMENT

The authors are very grateful to Mr. Ryo Hayakawa for providing numerical data and simulation programs in [6]. This work was partly supported by JSPS Grant-in-Aid for Scientific Research (A) Grant Number 17H01280, Grant-in-Aid for Scientific Research (B) Grant Number 16H02878 (TW), and Grant-in-Aid for Young Scientists (Start-up) Grant Number 17H06758 (ST).

REFERENCES

[1] S. Yang and L. Hanzo, “Fifty years of MIMO detection: the road to large-scale MIMOs,” IEEE Comm. Surveys and Tutorials, vol. 17, no. 4, pp. 1941-1988, Fourthquarter 2015.

[2] D. A. Shnidman, “A generalized Nyquist criterion and an optimum linear receiver for a pulse modulation system,” The Bell System Technical Journal, vol. 46, no. 9, pp. 2163-2177, Nov. 1967.

[3] K. K. Wong, A. Paulraj, and R. D. Murch, “Efficient high-performance decoding for overloaded MIMO antenna systems,” IEEE Trans. Wireless Commun., vol. 6, no. 5, pp. 1833-1843, May 2007.

[4] T. Datta, N. Srinidhi, A. Chockalingam, and B. S. Rajan, “Low-complexity near-optimal signal detection in underdetermined large-MIMO systems,” in Proc. National Conf. Comm., Feb. 2012, pp. 1-5.

[5] Y. Fadlallah, A. Alissa-El-Bey, K. Amis, D. Pastor, and R. Pyndiah, “New decoding strategy for underdetermined MIMO transmission using sparse decomposition,” in Proc. in Proc. 2013 21st Eur. Signal Proc. Conf., Sep. 2013, pp. 1-5.

[6] R. Hayakawa and K. Hayashi, “Convex optimization-based signal detection for massive overloaded MIMO systems,” in IEEE Trans. Wireless Commun., vol. 16, no. 11, pp. 7080-7091, Nov. 2017.

[7] R. Hayakawa, K. Hayashi, H. Sasa hara, and M. Nagahara, “Massive overloaded MIMO signal detection via convex optimization with proximal splitting,” in Proc. 2016 24th Eur. Signal Proc. Conf., Aug./Sep. 2016, pp. 1383-1387.

[8] M. Nagahara, “Discrete signal reconstruction by sum of absolute values” IEEE Signal Process. Lett., vol. 22, no. 10, pp. 1575-1579, Oct. 2015.

[9] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks.” Science, vol. 313, no. 5876, pp. 504-507, Jun. 2006.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks” in Proc. Advances in Neural Inf. Proc. Sys. 2012, pp. 1097-1105, 2012.

[11] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97, Nov. 2012.

[12] B. Aazhang, B. P. Paris, and G. C. Orsak, “Neural networks for multiuser detection in code-division multiple-access communications,” IEEE Trans. Commun., vol. 40, no. 7, pp. 1212-1222, Jul. 1992.

[13] E. Nachmani, Y. Beéty, and O. Burshtein, “Learning to decode linear codes using deep learning,” 2016 54th Annual Allerton Conf. Comm., Control, and Computing, 2016, pp. 341-346.