

Deep Learning Aided LLR Correction Improves the Performance of Iterative MIMO Receivers

Jue Chen, Tsang-Yi Wang, *Member, IEEE*, Jwo-Yuh Wu, *Member, IEEE*, Chih-Peng Li, *Fellow, IEEE*, Soon Xin Ng, *Senior Member, IEEE*, Robert G. Maunder, *Senior Member, IEEE* and Lajos Hanzo, *Life Fellow, IEEE*

Abstract—A Deep Learning (DL) aided Logarithmic Likelihood Ratio (LLR) correction method is proposed for improving the performance of Multiple-Input Multiple-Output (MIMO) receivers, where it is typical to adopt reduced-complexity algorithms for avoiding the excessive complexity of optimal full-search algorithms. These sub-optimal techniques typically express the probabilities of the detected bits using LLRs that often have values that are not consistent with their true reliability, either expressing too much confidence or not enough confidence in the value of the corresponding bits, leading to performance degradation. To circumvent this problem, a Deep Neural Network (DNN) is trained for detecting and correcting both over-confident and under-confident LLRs. We demonstrate that the complexity of employing the DL-aided technique is relatively low compared to the popular reduced-complexity receiver detector techniques since it only depends on a small number of real-valued inputs. Furthermore, the proposed approach is applicable to a wide variety of iterative receivers as demonstrated in the context of an iterative detection and decoding aided MIMO system, which uses a low-complexity Smart Ordering and Candidate Adding (SOCA) scheme for MIMO detection and Low-Density Parity Check (LDPC) codes for channel coding. We adopt Extrinsic Information Transfer (EXIT) charts for quantifying the Mutual Information (MI) and show that our DL method significantly improves the Block Error Rate (BLER). Explicitly, we demonstrate that about 0.9 dB gain can be achieved at a BLER of 10^{-3} by employing the proposed DL-aided LLR correction method, at the modest cost of increasing the complexity by 16% compared to a benchmarker dispensing with LLR correction.

Index Terms—DL, LLR, MIMO, iterative detection and decoding, LDPC codes

I. INTRODUCTION

State-of-the-art wireless receivers rely on soft decision decoding, wherein soft information is used to express not

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/W016605/1, EP/X01228X/1 and EP/Y026721/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028). Wang's work is supported by NSTC 111-2221-E-110-019. J.-Y. Wu's work is supported in part by the National Science and Technology Council (NSTC) of Taiwan under grant MOST111-2221-E-A49-067-MY3, and in part by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education (MOE), Taiwan.

Corresponding Author: L. Hanzo, J. Chen, S.-X. Ng and R.-G. Maunder are with School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK. (E-mail: lh@ecs.soton.ac.uk; jc7m17@soton.ac.uk; sxn@ecs.soton.ac.uk; rm@ecs.soton.ac.uk).

T.-Y. Wang is with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan (E-mail: twang@mail.nsysu.edu.tw).

J.-Y. Wu is with the Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan (E-mail: jywu@cc.nctu.edu.tw).

C.-P. Li is with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, and also with the Department of Engineering and Technologies, National Science and Technology Council, Taiwan (E-mail: cpli@faculty.nsysu.edu.tw).

only what the most likely value of each bit is, but also how likely it is. This soft information is typically represented using Logarithmic Likelihood Ratios (LLRs) processed by probabilistic detection [1], [2]. However, in some practical systems, these LLRs may not accurately express true confidence in the values of the bits, owing to deficiencies in channel estimation or the adoption of reduced-complexity algorithms. This may lead to inaccurate LLRs being passed from the detector into a sensitive decoder, such as the offset min-sum Low-Density Parity Check (LDPC) decoder [3], which can result in performance loss [4]–[9]. Xu *et al.* [7] demonstrate for a tree-search algorithm-based Multiple-Input Multiple-Output (MIMO) detector that the LLRs may become less self-consistent, with some LLRs being over-confident and some LLRs being under-confident, owing to the use of a reduced-complexity Sphere Decoder (SD) technique. In this treatise, we mitigate this problem.

To elaborate, we reduce the performance degradation caused by inconsistent LLRs using a low-complexity DL technique. More specifically, a Deep Neural Network (DNN) [10], [11] is trained to learn the relationship between the inconsistent LLRs and the true LLR values, which may be then applied for correcting the LLRs. More specifically, the mapping function is learned during an offline learning process as a function of the channel SNR per bit (E_b/N_0), of the iteration index, and of the raw LLRs. During the training data processing, we employ a novel statistical method to avoid running complex Maximum-Likelihood (ML) MIMO detection algorithms. We demonstrate that the technique proposed is eminently suitable for a low-complexity MIMO system detector exchanging LLRs with an LDPC decoder. The iterative exchange of *extrinsic* LLRs between the MIMO detector and LDPC decoder has been shown to achieve beneficial performance gain over non-iterative receivers [12], [13]. While the full-search-based ML algorithm [14]–[16] achieves optimal MIMO detection, its complexity is excessive for a high number of antennas or high-order modulation schemes. To address this challenge, many reduced-complexity non-linear MIMO detectors have been proposed, which aim to select subsets of the detection candidates to be used in the numerator and denominator of the LLR calculation algorithm [7] of Eq. 11. This includes the Smart Ordering and Candidate Adding (SOCA) scheme of [17], which is particularly suited to practical implementations using parallel processing and low latency hardware implementation [18]. While the selection of subsets of the detection candidates to use the numerator and denominator of the LLR calculation algorithm can significantly reduce the complexity, this is at the cost of reducing

the self-consistency of the LLRs. In this paper, we propose a DL-aided scheme for correcting the self-consistency of the LLRs and therefore improving the performance of the iterative MIMO detection. Our technique is suitable for any reduced-complexity iterative MIMO detector and we demonstrate it for the case of using SOCA detection. Here, a tree structure is used for representing all possible candidates explored by an ML detector, but rather than traversing all tree nodes, the SOCA algorithm reduces the complexity by pruning the number of candidates considered during the MIMO detection. However, the natural consequence of this is that the self-consistency of the LLRs proposed by the SOCA detector is degraded, which can cause performance degradation.

We benchmark the proposed DL technique against a LookUp Table (LUT) based method for LLR correction as proposed in [4], [7]. This LUT-aided LLR correction scheme takes raw LLRs as its input and quantizes them to the nearest value in the LUT, which then provides the corresponding corrected LLR values. Here, different LUTs may be used depending on the values of the system parameters, such as the index of the iteration between detection and decoding, or the Signal to Noise Ratio (SNR), which again must be quantified to the nearest represented value. However, having multiple LUTs for the LLRs and SNRs limits the practical feasibility of this scheme. Furthermore, a large amount of memory may be required to store the LUTs.

We also consider an alternative benchmark, namely the Fast-convergence Sparsely connected detection Network (FS-Net) proposed by Nguyen *et al.* [19], [20] to find highly likely candidates for signal detection at low complexity. FS-Net represents the only DL technique available in the literature, which may be combined with a traditional MIMO detector for iterative detection and decoding. More specifically, FS-Net is a DL technique that has been shown to offer good performance in traditional hard-decision MIMO detection, which outputs the estimated IQ signals directly. In this work, we build a benchmarker by combining the FS-Net with a SOCA algorithm by consulting an iterative detection and decoding scheme. In this FS-Net-aided SOCA-LDPC benchmarker, we use the FS-Net to search for the most likely estimated signal candidate. Then, during the SOCA tree search, the candidates with higher costs than the most likely candidate may be pruned from the search. However, in contrast to the DL technique proposed for correcting the SOCA LLRs, our results show that the FS-Net is unable to improve the decoding performance of the reduced-complexity SOCA MIMO detector.

The main contribution of this work is that we propose a DL-aided LLR correction technique, which can be employed in different scenarios, where the LLRs are not self-consistent, such as when a reduced-complexity MIMO detector is used. We proposed the use of the histogram statistic based method for generating the training data, in order to train the DNN. Furthermore, we show that the training is based on a small number of input parameters to obtain improved performance gains with lower memory requirements compared to a LUT based benchmarker. Owing to the use of a small number of input parameters, we demonstrate that the training is simple and robust. Here, we explicitly contrast our contributions to the state-of-the-art in Table I and detail them below:

TABLE I: Contrasting our contribution to the state-of-the-art

	[4]	[5]	[6]	[7]	[8]	[9]	Proposed
LLR correction	✓	✓	✓	✓	✓	✓	✓
DL							✓
MI analysis	✓	✓			✓		✓
Clipping LLRs	✓						✓
Histogram statistic method							✓
Offline training							✓

- We conceive the first DL-aided LLR correction technique, which has a simple structure, hence it is easy to build and train.
- We propose a novel histogram-based statistical method for generating training data, which avoids the requirement for using a full-search-based ML MIMO detector, which significantly reduces the complexity of the training phase. Furthermore, we demonstrate that our DL technique can be trained offline, which reduced the complexity compared to reinforcement learning. We show that once the DNN is well-trained, it can be employed for different use cases. Furthermore, an additional benefit of offline training is the reduction of the memory required to store the trained DNN.
- We demonstrate that the proposed DL LLR correction technique may be applied in different iterative receiver schemes, which suffer from performance degradation caused by low-complexity algorithms, which generate approximate LLRs.
- Furthermore, we demonstrate the application of our DL-aided LLR correction technique in a particular SOCA-LDPC system. We employ a Mutual Information (MI) analysis-based technique for characterizing the self-consistency of the iteratively exchanged LLRs and use this to demonstrate that our proposed DL-aided LLR correction indeed improves the self-consistency of the iteratively exchanged LLRs in a quantifiable manner, and this leads to a BLER vs. SNR improvement of about 0.9 dB at the cost of only increasing the complexity by 16%. This is comparable to that of the LUT-aided LLR correction benchmarker, which can achieve about 0.7 dB gain at the cost of increasing the complexity by about 20%. However, the benchmarker has a much higher memory requirement than the proposed approach. We show that the FS-Net-aided benchmarker has the lowest complexity but at the cost of significantly degraded performance.

The rest of this paper is organized as follows. Section II introduces our novel technique for DL-aided LLR correction, including a discussion of training data generation and DNN training. Furthermore, our MI analysis shows that the proposed DL-aided LLR correction improves the self-consistency of the corrected LLRs. Following this, Section III demonstrates the application of the DL-aided LLR correction in a specific SOCA-LDPC system. The equations of the complexity derived for the proposed scheme and for the benchmarkers are formulated in Section IV. Following this, Bit Error Rate (BER), BLock Error Rate (BLER), and complexity comparison simulation results are presented in Section V. Finally, we offer our conclusions in Section VI.

II. DNN FOR DEEP LEARNING

In this section, we introduce a DNN for our DL technique in the context of improving the self-consistency of LLRs produced by low-complexity detectors. We discuss how offline processing may be used for training a DNN, and how the DNN may be applied at run time for improving the self-consistency of LLRs. Section II-A describes how the training data may be generated, while Section II-B introduces our DNN for this DL application.

A. Training data generation

TABLE II: Definition of parameters in an example scheme, which comprises an LDPC decoder and a low complexity MIMO SOCA detector.

K'	Number of information bits in each frame
R	LDPC coding rate
N_t	Number of transmit antennas
N_r	Number of receive antennas
$\mathbf{M} = [M_1, \dots, M_{N_t}]$	M_n : number of surviving nodes in the n^{th} layer of the SOCA tree search
$\mathbf{K} = [K_1, \dots, K_{N_t}]$	K_n : number of nodes to extend from each parent node in the n^{th} layer of the SOCA tree search
L_{\max}	LLR clipping value
I_{LDPC}	Number of LDPC decoding iterations
$I_{\text{SOCA-LDPC}}$	number of iterations performed between SOCA detection and LDPC decoding
$i_{\text{SOCA-LDPC}}$	Index of the iteration $i_{\text{SOCA-LDPC}}^{\text{th}}$ between SOCA detection and LDPC decoding

Figure 1(a) portrays our turbo detection aided scheme, wherein a reduced-complexity inner decoder iteratively exchanges LLRs with an outer decoder. Our novel scheme introduces a DL and an LLR clipping function for improving the self-consistency of the LLRs provided by the reduced-complexity inner decoder. A binary vector \mathbf{a} , comprising K' bits are encoded by an outer encoder in order to obtain the encoded bit vector \mathbf{b} , which comprises $E > K'$ bits. Then, the order of the bits in the encoded vector \mathbf{b} is rearranged by the turbo interleaver Π , in order to obtain the bit vector \mathbf{c} , which also comprises E bits. This is entered into an inner encoder in order to construct the transmitted signal vector \mathbf{s} , comprising F number of symbols. The symbols of the vector \mathbf{s} are then transmitted through the channel of Figure 1(a), in order to obtain the received signal vector \mathbf{r} , which also comprises F symbols. This is detected using a reduced-complexity inner decoder in order to obtain the *extrinsic* LLR vector $\tilde{\mathbf{c}}^e$, which pertains to the bit vector \mathbf{c} and hence comprises E number of LLRs. Next, the values of the *extrinsic* LLRs in the vector $\tilde{\mathbf{c}}^e$ are clipped to L_{\max} by the clipping function to obtain the *extrinsic* LLR vector $\tilde{\mathbf{c}}'^e$. These clipped *extrinsic* LLRs of $\tilde{\mathbf{c}}'^e$ are input to the DL scheme to obtain the vector of E corrected *extrinsic* LLRs $\tilde{\mathbf{c}}^e$. The order of LLRs in the vector of $\tilde{\mathbf{c}}^e$ is rearranged by the de-interleaver Π^{-1} in order to obtain the vector of E *a priori* LLRs $\tilde{\mathbf{b}}^a$, which are then passed to the outer decoder of Figure 1(a). The vector of E *extrinsic* LLRs $\tilde{\mathbf{b}}^e$ output by the outer decoder are in turn interleaved by the interleaver Π to obtain the vector of E *a priori* LLRs $\tilde{\mathbf{c}}^a$, which are then clipped to obtain the vector $\tilde{\mathbf{c}}'^a$. Having now completed the first iteration, the second iteration may be performed, in which the reduced-complexity inner decoder accesses both the received signal vector \mathbf{r} and the *a priori*

LLR vector $\tilde{\mathbf{c}}'^a$, as its inputs. The process is repeated until the affordable number of iterations is completed, whereupon the estimated binary vector $\hat{\mathbf{a}}$, comprising K' bits is output by the outer decoder of Figure 1(a).

Figure 1(b) illustrates the signal flow of generating training data for the proposed DL-aided LLR correction scheme. As shown by the dashed lines in Figure 1(b), it differs from Figure 1(a), because it replaces the DL scheme by a block counting and correcting the LLRs, which requires *a priori* knowledge of the encoded bit vector \mathbf{c} . This is reasonable in this offline training environment, but naturally it is impossible in a realistic system in a real deployment.

As described in Section I, the use of a reduced-complexity inner decoder namely a SOCA may lead to relatively poor self-consistency for the *extrinsic* LLRs in the vector $\tilde{\mathbf{c}}'^e$, compared to the LLRs produced by a full-complexity ML decoder. Hence, the objective of the proposed DL technique of Figure 1(a) is to replace these LLRs by a vector of corrected *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ having improved self-consistency. Here, the self-consistency of the raw *extrinsic* LLR vector $\tilde{\mathbf{c}}'^e$ may be quantified by its Mutual Information (MI). In particular, we may adopt two different techniques for measuring MI. The true measured MI of the raw *extrinsic* LLR vector $\tilde{\mathbf{c}}'^e$ may be quantified by the histogram-based method of [21], which makes a comparison with the bit vector \mathbf{c} . By contrast, the claimed MI of the raw *extrinsic* LLR vector $\tilde{\mathbf{c}}'^e$ may be quantified by the averaging method, which does not consider the bit vector \mathbf{c} . Instead, intuitively trusts the LLRs by assuming that the high-magnitude LLRs resultant a high MI, even though sometimes the values or signs of the LLRs may be inaccurate. In our solution, the measured value is calculated using the histogram-based method by considering the corresponding bit sequence of the LLRs, which represents the true MI. In cases where the claimed MI obtained by the averaging method differs from the true measured MI obtained by the histogram-based method, we can deduce that the LLRs are not self-consistent.

Figure 2 plots the quality of the *extrinsic* LLRs $\tilde{\mathbf{c}}'^e$ output by the SOCA detector for the case of an example scheme, where it is concatenated with an LDPC decoder. The parameters of this scheme are captured in Table II and are further elaborated on in Section III. More explicitly, Figure 2 characterizes the self-consistency of the raw *extrinsic* LLRs $\tilde{\mathbf{c}}'^e$ as a function of channel SNR per bit (E_b/N_0), after performing $I = 3$ iterations between the LDPC decoder and SOCA detector. Here, the solid lines characterize the MI of the raw *extrinsic* LLRs $\tilde{\mathbf{c}}'^e$, which are obtained using the averaging method of [22]. Furthermore, the dashed lines represent the measured MI of the raw *extrinsic* LLRs $\tilde{\mathbf{c}}'^e$, which are obtained using the histogram based method of [21]. Here, the match between the claimed MIs and measured MIs in the case of the LLR is self-consistent, where a smaller gap between the claimed MIs and measured MIs represents improved self-consistency between the raw *extrinsic* LLRs $\tilde{\mathbf{c}}'^e$. Figure 2 characterizes the claimed and measured MIs when employing clipping for different clipping values L_{\max} . Explicitly, clipping mitigates the LLRs that have unwarranted high values, which express over-confidence, as mentioned before. As shown in the figure, the scheme that does not use clipping suffers from the substantial discrepancy between

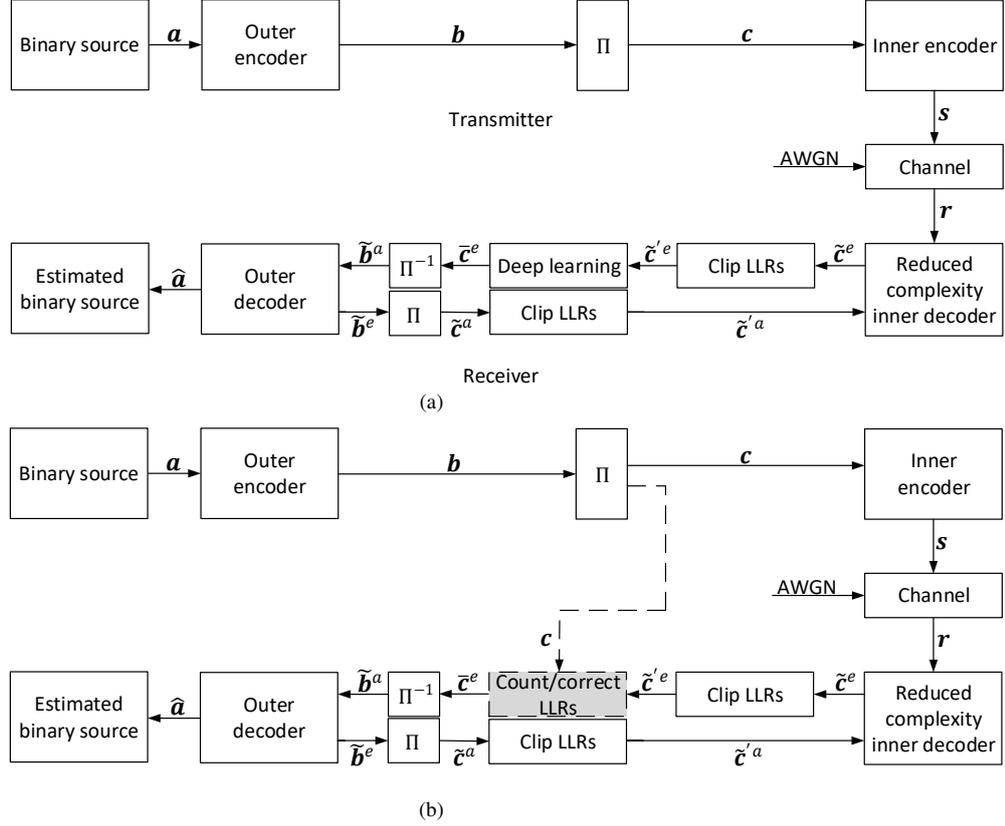


Fig. 1: 1(a) schematic of the proposed DL-aided LLR correction in an iterative receiver scheme. 1(b) schematic of generating training data for the proposed DL-aided LLR correction scheme.

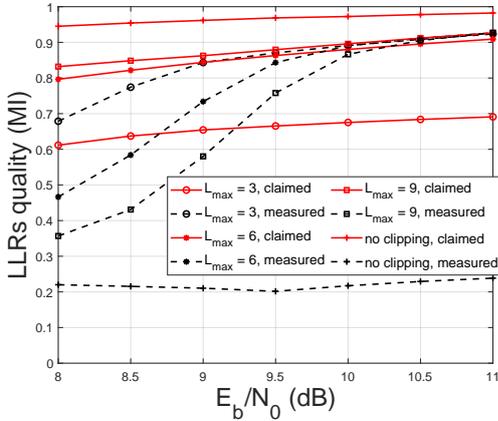


Fig. 2: Quality of the *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ qualified by claimed MI using the averaging method and measured MI obtained using the histogram method, for a SOCA-LDPC scheme having $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, $\mathbf{M} = [8 \ 1 \ 1]$, $\mathbf{K} = [8 \ 8 \ 8]$, $I_{LDPC} = 20$ iterations inside the LDPC decoding, $I = 3$ iterations between SOCA detector and LDPC decoder and 16-QAM modulation for communication over Rayleigh fading channel.

the claimed and measured MIs, because the LLRs are not self-consistent. By contrast, the introduction of clipping can be seen to significantly improve the self-consistency of the raw *extrinsic* LLRs $\tilde{\mathbf{c}}^e$, although some discrepancy remains, which is mitigated by DL as discussed in Section II. Here, the results show that the clipping value L_{\max} should be chosen carefully since a low clipping value L_{\max} may make the LLRs insufficiently confident, while a high clipping value L_{\max} may retain those LLRs that are over-confident. As shown in Figure 2, $L_{\max} = 6$ is a good choice for improving the self-

consistency of the raw *extrinsic* LLRs $\tilde{\mathbf{c}}^e$, like with [4].

Figure 3 characterizes the MIs for different M_1 values in the conventional SOCA-LDPC scheme. It may be observed that the solid lines and the dashed lines converge as the E_b/N_0 is increased, indicating that the *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ become self-consistent at high E_b/N_0 values, but suffer from inconsistency at low E_b/N_0 values. As the number of extended tree nodes M_1 is increased, the complexity of the SOCA detector is increased and hence a higher LLR self-consistency is achieved. But as shown in Figure 3, when M_1 is low, large MI discrepancies are observed between the claimed and measured MI, which indicates that the *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ are not self-consistent. These discrepancies may be explained by the large fraction of signal candidates that are not considered in the LLR calculation of the low-complexity SOCA detector when M_1 is small. This leads to sub-optimal LLR values. This degrades the performance of a conventional SOCA-LDPC scheme, which deinterleaves the *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ and directly passes the resultant *a priori* LLRs $\tilde{\mathbf{b}}^a$ to the outer decoder without employing LLR correction.

In order to correct the inconsistent *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ provided by a reduced-complexity inner decoder, a DNN may be trained to find the relationship between the raw value of the *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ and the optimal values of corrected *extrinsic* LLRs $\tilde{\mathbf{c}}^e$. More specifically, supervised learning may be employed using a large amount of training data, comprising pairs of the raw values of the *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ with the optimal values of the corrected *extrinsic* LLRs $\tilde{\mathbf{c}}^e$. Hence, it is necessary to generate training data based on corrected *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ before building a DNN. Figure 4 shows the flow of generating the corrected *extrinsic* LLRs $\tilde{\mathbf{c}}^e$ for the

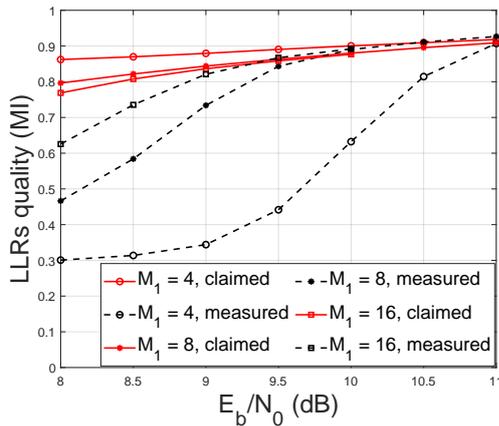


Fig. 3: Quality of the *extrinsic* LLRs \tilde{c}^e qualified by claimed MI using the averaging method and measured MI obtained using the histogram method, for a SOCA-LDPC scheme having $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K^l = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$, $M = [M_1 \ 1 \ 1 \ 1]$, $\mathbf{K} = [M_1 \ M_1 \ M_1 \ M_1]$, $I_{\text{LDPC}}=20$ iterations inside the LDPC decoding, $I = 3$ iterations between SOCA detector and LDPC decoder, and 16-QAM modulation for communication over Rayleigh fading channel.

DL training data, which was generated by analyzing the raw values of the *extrinsic* LLRs \tilde{c}^e with the aid of the correct bit sequence \mathbf{c} , representing the function of the ‘Count/correct’ block of Figure 1(b). Naturally, the correct bit sequence \mathbf{c} is not known in the real system, but in our offline process of generating training data, the correct bit sequence \mathbf{c} may be assumed as *a priori* knowledge. In the first step of the flow of generating the training data, each *extrinsic* LLR in the training vector \tilde{c}^e is matched to its corresponding bit in the sequence \mathbf{c} , and when the corresponding bit has the value 0, the LLR is placed into the sequence \tilde{c}_0^e . Otherwise, it is placed into the sequence \tilde{c}_1^e . Following this, the mean of the LLRs in \tilde{c}_0^e is calculated, which is used for calculating the variance σ_0^2 of the LLRs in \tilde{c}_0^e . Likewise, the mean of the LLRs in \tilde{c}_1^e is calculated and used for calculating the variance σ_1^2 of the LLRs in \tilde{c}_1^e . As shown in Figure 4, these two variances of \tilde{c}_0^e and \tilde{c}_1^e are used for calculating a bin width β , which will be used to perform histogram analysis. This is achieved by using the following equation:

$$\beta = \frac{1}{2} \left(3.49\sqrt{\sigma_0^2} \times (V_0)^{-\frac{1}{3}} + 3.49\sqrt{\sigma_1^2} \times (V_1)^{-\frac{1}{3}} \right). \quad (1)$$

where Eq. 1 is adopted from the technique of [23] for automatically choosing the appropriate bin width of a histogram for Gaussian distribution data. Here, β is the width of each histogram bin, V_0 is the number of elements in \tilde{c}_0^e , and V_1 is the number of elements in \tilde{c}_1^e .

Following this, the range spanning from the smallest value to the largest value among the *extrinsic* LLRs of \tilde{c}^e separated into a certain number of bins, each having a width of β . Next, the number of LLRs V_0^t and V_1^t in each of \tilde{c}_0^e and \tilde{c}_1^e corresponding to the bin having each index $\in [1, T]$ is counted, in order to obtain a pair of histograms, as shown in Fig 4. Following this, we may quantify the probability of a 0-valued bit corresponding to an LLR value in the t^{th} bin as $P_0^t = \frac{V_0^t}{V_0}$. Similarly, the probability of a 1-valued bit corresponding to an LLR value in the t^{th} bin is given by $P_1^t = \frac{V_1^t}{V_1}$. Then, the optimal value for an LLR falling into the

t^{th} bin may be calculated as

$$L^t = \ln \left(\frac{P_0^t}{P_1^t} \right). \quad (2)$$

A LUT may be used to store both a raw LLR value and the corresponding corrected LLR for each bin. More specifically, the LUT comprises two rows, where the first row contains T raw LLR values, which are taken from the middle value of each bin. By contrast, the second row contains the corresponding corrected LLR value for each bin, obtained using Eq. 2. Following this, linear interpolation may be used for selecting a corrected LLR value of \tilde{c}^e for each of the raw LLRs in \tilde{c}^e . Then, the sequences of raw *extrinsic* LLRs \tilde{c}^e and the corrected *extrinsic* LLRs \tilde{c}^e may be employed for training the DNN. In order to train the DNN more accurately, the process may be repeated over the number of frames denoted using the notation O . In this way, O frames bit sequence \mathbf{c} are simulated to obtain O frames raw *extrinsic* LLRs \tilde{c}^e and O frames corrected *extrinsic* LLRs \tilde{c}^e . The training of a DNN is discussed in the next section.

B. DNN for deep learning

In this section, we introduce the DNN proposed for LLR correction. In Section II-B1, the training phase of the DL is introduced. Furthermore, Section II-B2 introduces the method proposed for correcting the raw *extrinsic* LLRs \tilde{c}^e .

1) *Offline training*: In this work, a DNN is trained using the generated training data discussed in Section II-A, which characterizes the relationship between the raw *extrinsic* LLRs \tilde{c}^e and the corresponding corrected *extrinsic* LLRs \tilde{c}^e . Figure 5 shows a typical fully connected neural network having $L = 5$ layers, including an input layer having three input elements, three hidden layers each having six neurons, and an output layer having a single output element. In this DNN, $(L - 1)$ non-linear transformations are performed to map the input $\mathbf{x}^1 = [E_b/N_0 \ i_{\text{inner-outer}} \ \tilde{c}_i^e]^T$ to the output y^L according to

$$y^L = \Phi(\mathbf{x}; \mathbf{\Delta}) = f^{L-1} \left(\dots (f^1(\mathbf{x}^1; \mathbf{\Delta}^1); \dots); \mathbf{\Delta}^{L-1} \right). \quad (3)$$

Here, the function $f^l(\cdot)$ of the l^{th} layer is expressed as

$$f^l(\mathbf{x}^l; \mathbf{\Delta}^l) = \alpha^l \left(\mathbf{W}^l \mathbf{x}^l + \boldsymbol{\theta}^l \right), \quad (4)$$

where α^l represents a Rectified Linear Unit (ReLU) activation function for the l^{th} layer. We employ d^l to represent the number of nodes in the l^{th} layer. Here, we have $\mathbf{\Delta}^l \triangleq \left\{ \mathbf{W}^l; \boldsymbol{\theta}^l \right\}$, whose $\mathbf{W}^l \in \mathbb{R}^{d^l \times d^{l-1}}$ is the weight-matrix, $\mathbf{x} \in \mathbb{R}^{d^l \times 1}$ is the input of the l^{th} layer and $\boldsymbol{\theta}^l \in \mathbb{R}^{d^l \times 1}$ is the bias vector. Furthermore, $\mathbf{\Delta}$ contains all weight matrices and bias vectors for all layers.

During the offline training phase, the input of the DNN includes three parameters of the system, namely E_b/N_0 , the index of the iteration $i_{\text{inner-outer}}$ being performed between the inner and outer decoder, as well as the raw *extrinsic* LLRs \tilde{c}_i^e being corrected. In this way, the DNN is trained with independent input vectors, given by:

$$\mathbf{x}_i^1 = [E_b/N_0 \ i_{\text{inner-outer}} \ \tilde{c}_i^e]^T. \quad (5)$$

Each raw *extrinsic* LLR \tilde{c}_i^e has an index $\in [1, E \times O]$, because there are E bits of LLR in one frame and in order to obtain

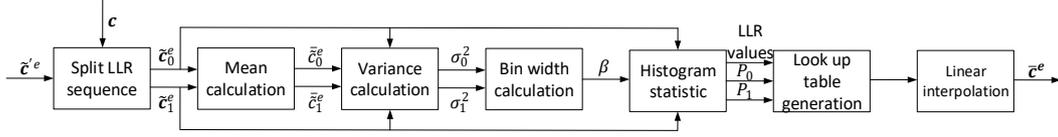


Fig. 4: The flow of generating DNN training data, comprising pairs of raw values for the *extrinsic* LLRs \tilde{c}^e and the optimal values of the corrected *extrinsic* LLRs \bar{c}^e .

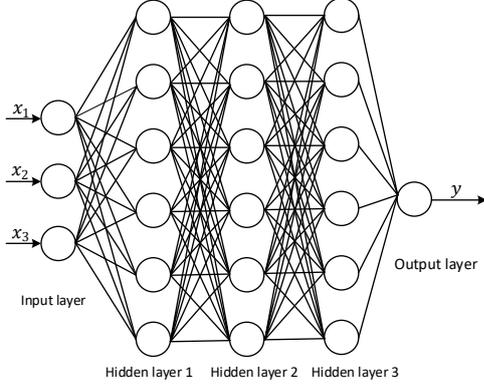


Fig. 5: A fully connected neural network with three inputs, three hidden layers, and one output.

a well-trained DNN, we generate training data comprising O number of LLR vectors \tilde{c}^e . We merge these O vectors into a single vector and then marry each of the LLRs $\tilde{c}_i^e \in [1, E \times O]$ with the corresponding E_b/N_0 and the iteration index $i_{\text{inner-outer}}$, in order to generate each input vector. The DL can flexibly apply different corrections to the raw LLRs across a range of E_b/N_0 and $i_{\text{inner-outer}}$ values, owing to the use of different E_b/N_0 and $i_{\text{inner-outer}}$ values for training the DNN during the offline training phase. The weights and biases of the DL are obtained by minimizing the following Mean-Squared Error (MSE) based loss function

$$\text{Loss}(\mathbf{M}) = \frac{1}{E \times O \times U \times I_{\text{SOCA-LDPC}}} \times \sum_{i=1}^{E \times O \times U \times I_{\text{SOCA-LDPC}}} |\bar{c}_i^e - \Phi(\mathbf{x}_i; \mathbf{\Delta})|^2. \quad (6)$$

Here, \bar{c}_i^e represents the corrected *extrinsic* LLRs corresponding to the raw *extrinsic* LLR \tilde{c}_i^e , as calculated during the training data generation phase using Figure 4. Moreover, U is the number of different E_b/N_0 values considered and $I_{\text{SOCA-LDPC}}$ is the number of iterations calculated between SOCA detection and LDPC decoding. A batch of inputs obtained using Eq. (5) is fed to the DNN, in order to obtain the output $y^L = \Phi(\mathbf{x}; \mathbf{\Delta})$, which represents the corrected LLR estimate. The DNN is trained by minimizing the MSE loss function of Eq. (6), where \bar{c}_i^e represents the true value that the LLR should adopt as obtained using the histogram statistic based method introduced in Section II-A.

2) *Online LLR correction*: During online LLR correction, the DNN's $(L-1)$ weight matrices and $(L-1)$ bias vectors stored during the offline training are employed to correct the raw *extrinsic* LLRs \tilde{c}^e . We consider the training phase required for the proposed scheme to be practical, since it only relies on $O = 500$ frames, each comprising 1000 LLRs, in order to complete the training for each combination of our 10 selected E_b/N_0 values and 3 iteration indices. More

specifically, the training is completed by merging the O frames into a matrix, where each row comprises the pairing of an input LLR value and the desired corrected output LLR value. We use 15 epochs during the training, which takes about 10 to 20 minutes. We repeated the training three times for each of the three different M_1 values that are considered in this paper. During the training phase, the E_b/N_0 and the iteration index are set at the input of the DNN according to $\mathbf{x}_i^1 = [E_b/N_0 \ i_{\text{inner-outer}} \ \tilde{c}_i^e]^T$. Furthermore, in accordance with the training phase, the 'ReLU' activation function [20] is employed, and the corrected *extrinsic* LLR \bar{c}_i^e is formulated as

$$\Phi(\mathbf{x}_i; \mathbf{M}) = \prod_{l=1}^{L-1} \max \left\{ \mathbf{0}, \left(\mathbf{W}^l \mathbf{x}_i^l + \boldsymbol{\theta}^l \right) \right\}, \quad i = 1, \dots, E. \quad (7)$$

The complexity of utilizing the trained DNN to correct the LLRs may be expressed as follows [24]

$$C_{DL} = \sum_{l=2}^L (2d^{l-1} + 1) \times d^l. \quad (8)$$

As shown in Eq. 8, the complexity associated with employing the trained DNN depends both on the number of layers and the number of neurons employed in each layer, which in turn depend on the number of input elements. Typically, the more input elements, the more layers, and neurons are required, where the size of a DNN is usually selected by simulations [20]. Hence, an advantage of our proposed scheme is that the input vector comprises only three elements, which are not influenced by the choice of the specific inner decoder.

Figure 6 characterizes the MI of the corrected *extrinsic* LLRs \bar{c}^e using the proposed DL-aided LLR correction approach in a particular iterative SOCA-LDPC scheme. The same parameters are used in this figure as in Figure 3. Again, the solid lines represent the claimed MI values and the dashed lines represent the measured MI values. As seen from Figure 6, the discrepancies between the claimed and measured MI values of the corrected *extrinsic* LLRs \bar{c}^e are much smaller compared to those associated with the raw *extrinsic* LLRs \tilde{c}^e in Figure 3. In particular, when M_1 is small, the discrepancy reduction is particularly significant. The discrepancy reduction indicates that the corrected *extrinsic* LLRs \bar{c}^e have become more self-consistent, owing to the adoption of the proposed DL-aided LLR correction scheme. As a result, the *a priori* LLRs $\tilde{\mathbf{b}}^a$ of the LDPC decoder become more self-consistent than the *a priori* LLRs $\tilde{\mathbf{b}}^a$ of the conventional SOCA-LDPC scheme, leading to improved performance, as it will be characterized in Section V. The MI analysis presented here demonstrates that our proposed DL-aided LLR correction approach successfully corrects the inconsistent LLRs.

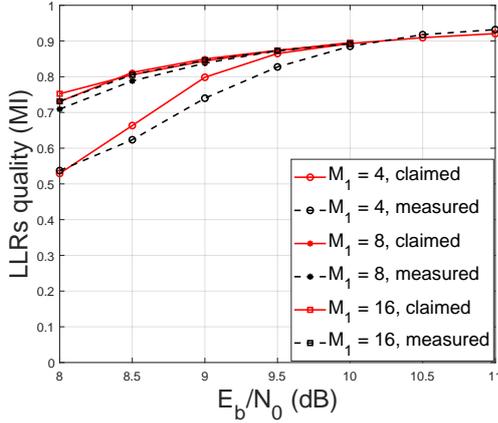


Fig. 6: Quality of the *extrinsic* LLRs \tilde{c}^e qualified by claimed MI using the averaging method and measured MI obtained using the histogram method, for a SOCA-LDPC scheme having $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$, $\mathbf{M} = [M_1 \ 1 \ 1 \ 1]$, $\mathbf{K} = [M_1 \ M_1 \ M_1 \ M_1]$, $I_{\text{LDPC}} = 20$ iterations inside the LDPC decoding, $I = 3$ iterations between SOCA detector and LDPC decoder, and 16-QAM modulation for communication over Rayleigh fading channel.

III. SYSTEM MODEL

In this section, we introduce a specific example of the serially concatenated iterative receiver of Figure 7, which employs a reduced-complexity MIMO detector to show that the proposed DL technique is beneficial for improving the self-consistency of the iteratively exchanged LLRs. Furthermore, the transmitter, channel model and receiver of LDPC-coded MIMO system are detailed in Sections III-A, III-B and III-C, respectively.

A. Transmitter

As shown in Figure 7, the input of the proposed transmitter is a random binary vector \mathbf{a} having a length K' , which is encoded by an LDPC encoder, in order to obtain the binary vector \mathbf{b} comprising $E > K'$ number of bits. The resultant coding rate is $R = K'/E$. The order of the encoded bits in the vector \mathbf{b} is randomly rearranged by the interleaver Π in order to obtain the bit vector \mathbf{c} , whose bits are QAM modulated and distributed among the N_t number of transmit antennas. Here, $G = 2^\omega$ -ary QAM modulation is employed, i.e. ω bits are transmitted per QAM symbol. More specifically, each antenna transmits a sequence comprising $F = E/(\omega N_t)$ QAM symbols, over a series of F time instants. The signal transmitted in each time instant is represented by the vector $\mathbf{s} = [s_1, \dots, s_{N_t}]^T$, where each component of \mathbf{s} is a QAM symbol having an average power E_s .

B. Channel model

We assume that N_r receive antennas are employed. The symbol vector \mathbf{s} of each time instant is transmitted through a $(N_r \times N_t)$ MIMO channel, hence the received signals may be represented as

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{n}. \quad (9)$$

We assume an uncorrelated narrow-band Rayleigh fading channel, where $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$. Here, $\mathbf{n} \in \mathbb{C}^{N_r \times 1}$ is a zero-mean complex Gaussian distributed random noise vector, where each element has a variance of N_0 . The SNR at each receive antenna is given by E_s/N_0 , which may be expressed as the SNR per bit of $E_b/N_0 = E_s/(N_0 \cdot R \cdot \omega)$.

C. Receiver

In this section, we detail the proposed receiver. We commence by describing the iteratively exchanged *extrinsic* LLRs between an LDPC decoder and a reduced-complexity MIMO detector in Section III-C1. Here, the MIMO detector adopts the SOCA algorithm. Following this, we detail the operations of the reduced-complexity SOCA detector in Section III-C2.

1) *Iterative detection and decoding*: For the receiver, we assume perfect channel estimation. The received signals are processed by the SOCA detector applied to the *a priori* LLRs \tilde{c}^a provided by the LDPC decoder, in order to obtain the *extrinsic* LLRs \tilde{c}^e . In the first iteration, the *a priori* LLR vector \tilde{c}^a is initialized with zero-valued LLRs. Following SOCA detection, the *extrinsic* LLRs \tilde{c}^e are clipped in order to improve the system performance [25]–[27]. More specifically, the *extrinsic* LLRs having a magnitude of $|\tilde{c}^e| > L_{\max}$ are set to $\pm L_{\max}$ in order to obtain the *extrinsic* LLRs \tilde{c}'^e . Owing to the reduced-complexity of the SOCA detector, the raw *extrinsic* LLRs \tilde{c}'^e may suffer from a lack of self-consistency, as discussed in Section II-A. In order to address this, we pass the raw *extrinsic* LLRs \tilde{c}'^e to the trained DNN to obtain the corrected *extrinsic* LLRs \tilde{c}^e . During the DL-aided correction of each raw *extrinsic* LLR in the sequence \tilde{c}'^e , the inputs of the trained DNN are the SNR per bit $\frac{E_b}{N_0}$, the index of the current iteration $i_{\text{SOCA-LDPC}}$ and the corresponding LLR gleaned from the sequence of all raw *extrinsic* LLRs \tilde{c}'^e , as shown Eq. (5). The DL technique employs the weight matrices \mathbf{W}^l , bias vectors $\boldsymbol{\theta}^l$, and activation function α^l trained during the offline training process, where $l = 1, \dots, L-1$ is the index of each layer in the DNN.

The order of the corrected *extrinsic* LLRs \tilde{c}^e is rearranged by a de-interleaver employing the same interleaving pattern as the transmitters' interleaver, in order to create the *a priori* LLRs \tilde{b}^a . These LLRs in \tilde{b}^a are processed by the LDPC decoder in order to obtain the *extrinsic* LLRs \tilde{b}^e is discussed in [28]. Correspondingly, the order of the *extrinsic* LLRs \tilde{b}^e are rearranged by a corresponding interleaver in order to obtain the *a priori* LLRs \tilde{c}^a , which are clipped as discussed above, in order to obtain the *a priori* LLRs \tilde{c}'^a for the SOCA detector.

2) *SOCA detection*: In each time instant, the task of the SOCA detector is to derive *extrinsic* LLRs pertaining to the transmitted signal \mathbf{s} , based on the channel input information (\mathbf{H}, N_0) , on the received signals \mathbf{r} and on the *a priori* LLRs \tilde{c}'^a . More specifically, the SOCA detector has to find a list of the most likely candidates of transmitted signals \mathbf{s} , and then map these onto values for the encoded bits \mathbf{c} , in order to calculate the *extrinsic* LLRs \tilde{c}^e . Then *A Posteriori* LLRs $\tilde{c}^p(\hat{c}_i)$ pertaining to each bit in the sequence \mathbf{c} may be calculated as

$$\tilde{c}^p(\hat{c}_i) \triangleq \ln \left(\frac{P(\hat{c}_i = 0|\mathbf{r})}{P(\hat{c}_i = 1|\mathbf{r})} \right), \quad (10)$$

where $i = 1, \dots, \omega \times N_t$. More specifically, in each time instant, there are $\omega \times N_t$ *extrinsic* LLRs $\tilde{c}^e(\hat{c}_i)$, $i = 1, \dots, \omega \times N_t$, which may be obtained by detecting the transmitted signal \mathbf{r} . Furthermore, the *extrinsic* LLRs $\tilde{c}^e(\hat{c}_i)$, $i = 1, \dots, \omega \times N_t$ obtained from each of the F time instances may be concatenated in order to obtain the *extrinsic* LLR vector \tilde{c}^e as the output of the SOCA detector.

After the application of Bayes' rules and the max-log ap-

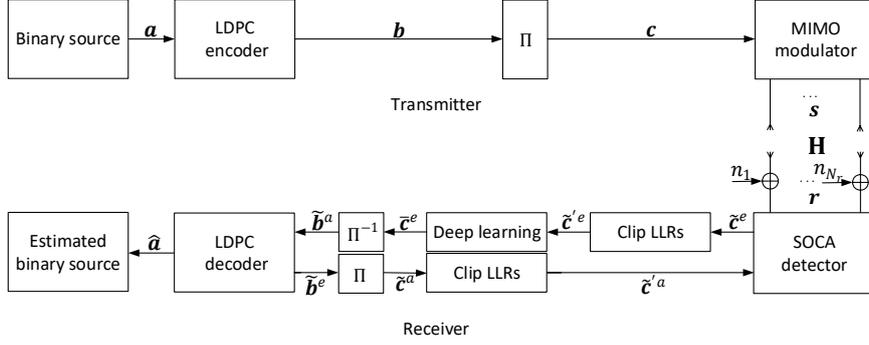


Fig. 7: Schematic of the proposed DL-aided LLR correction LDPC-coded MIMO system.

proximation [29], the corresponding *extrinsic* LLRs $\tilde{c}^e(\hat{c}_i)$ pertaining to the i^{th} bits \hat{c}_i may be calculated as

$$\begin{aligned} \tilde{c}^e(\hat{c}_i) &= \tilde{c}^p(\hat{c}_i) - \tilde{c}'^a(\hat{c}_i) \\ &= \min_{\hat{s} \in \mathcal{X}_i^1} \left(\frac{\|\mathbf{r} - \mathbf{H}\hat{s}\|^2}{N_0} - \frac{1}{2} \sum_{i=1}^{\omega \times N_t} (2\hat{c}_i - 1) \times \tilde{c}'^a(\hat{c}_i) \right) \\ &\quad - \min_{\hat{s} \in \mathcal{X}_i^0} \left(\frac{\|\mathbf{r} - \mathbf{H}\hat{s}\|^2}{N_0} - \frac{1}{2} \sum_{i=1}^{\omega \times N_t} (2\hat{c}_i - 1) \times \tilde{c}'^a(\hat{c}_i) \right) - \tilde{c}'^a(\hat{c}_i), \end{aligned} \quad (11)$$

where $\tilde{c}'^a(\hat{c}_i)$ is the corresponding *a priori* LLR. Moreover, \mathcal{X}_i^1 denotes the set that includes all encoded bit sequence candidates \hat{c} , in which the corresponding bit \hat{c}_i adopts a value of 1. Similarly, \mathcal{X}_i^0 denotes the set that comprises all encoded bit sequence candidates, in which the corresponding \hat{c}_i adopts a value of 0. Furthermore, the candidate signal vector \hat{s} having a length of N_t is the hypothetical signal corresponding to \hat{c} according to a specific bit to constellation mapping.

In order to calculate the *extrinsic* LLRs \tilde{c}^e , the sets \mathcal{X}^1 and \mathcal{X}^0 should be explored, as shown in Eq. (11). When using a high number of antennas N_t and/or a modulation scheme having a number of constellation points G , the cardinality $N = G^{N_t}$ of the superset of \mathcal{X}^1 and \mathcal{X}^0 is very large. Hence, the traditional Maximum A Posteriori (MAP) algorithm suffers from excessive complexity, since it considers all candidates of the encoded bits \hat{c} , when calculating the *extrinsic* LLRs $\tilde{c}^e(\hat{c}_i)$, $i = 1, \dots, \omega \times N_t$. In order to reduce the complexity, the SOCA algorithm [17] considers only subsets of \mathcal{X}^1 and \mathcal{X}^0 when considering each of the encoded bits in \hat{c} . More specifically, these subsets are designed to comprise the most likely candidate and its counter-hypothesis, which are found using a tree search. Figure 8 presents a simple example of a SOCA tree search for the case of $N_t = 2$ transmit antennas, using QPSK modulation, which has $G = 4$ constellation points. While this toy example has only $N = G^{N_t} = 4^2 = 16$ candidates, the modernist case using for example $N_t = 4$ transmit antennas and $G = 16$ -QAM has $N = 16^4 = 65\,536$ candidates, leading to much higher complexity and motivating the SOCA algorithm.

In the tree search, each tree layer considers the signal transmitted by a particular one of the transmit antennas. However, we have the freedom to order the transmit antennas as we please, when constructing the tree. Furthermore, the number of child nodes derived from each parent node is given by the modulation order G and hence the total number of child nodes in the n^{th} , $n \in [1, N_t]$ layer is equal to G^n . Furthermore, each child node in the tree corresponds to

a binary code word, according to the specific constellation points that it represents, as shown in Figure 8. In the SOCA algorithm, there are only $M_n \leq G$ child nodes derived from each parent node, which will be extended in the n^{th} layer, where the particular set that is extended is selected according to the node metrics ψ quantified by the squared Euclidean distance [17]. In the case where we have $M_n = G$ for all layers, the SOCA algorithm becomes identical to the ML algorithm, which traverses all possible nodes in the tree. In the example of Figure 8, we adopt $M_1 = 3$ and hence only the 3 child nodes having the smallest node metrics are extended from the root node, with the remaining child nodes being pruned. Once 3 nodes in the first layer have been traversed, these extended child nodes are then traversed as parent nodes in the second layer. Note that the pruned child nodes in the first layer will not be traversed as parent nodes in the second layer. The example of Figure 8 adopts $M_2 = 2$ and hence the 2 child nodes of each parent node having the smallest metrics are extended in the second layer, with all other child nodes being pruned. Furthermore, when traversing all layers besides the first layer, the SOCA algorithm additionally extends the counter-hypothesis of the child node having the lowest node metric across the layer. For example, the child node in the second layer of Figure 8 identified as having the lowest node metric is associated with the codeword ‘0 0’, which has the counter-hypotheses of ‘0 1’ and ‘1 0’. Hence the child nodes corresponding to these codewords and extending from the same parent node as the child nodes having the lowest metrics are also extended in Figure 8. As an additional constraint, K_n child nodes having lower node metrics in the n^{th} layer will survive, while the child nodes having high node metrics are pruned. The example of Figure 8 adopts $K_1 = \infty$ and hence all surviving child nodes are retained in the first layer. Furthermore, our example in Figure 8 adopts $K_2 = 6$ and hence some of the child nodes identified for extension are pruned. Owing to this complexity reduction mechanism, the counter-hypothesis may not be extended for some layers and hence \mathcal{X}_i^1 or \mathcal{X}_i^0 may be empty in some cases, leading to $\pm\infty$ valued LLRs from Eq. 11. In these cases, we clip the corresponding LLRs to L_{\max} . In the case of Figure 8, the $K_2 = 6$ surviving child nodes are associated with the corresponding candidates:

$$\mathbf{x} = \begin{array}{l} \text{first layer} \\ \text{first layer} \\ \text{second layer} \\ \text{second layer} \end{array} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad (12)$$

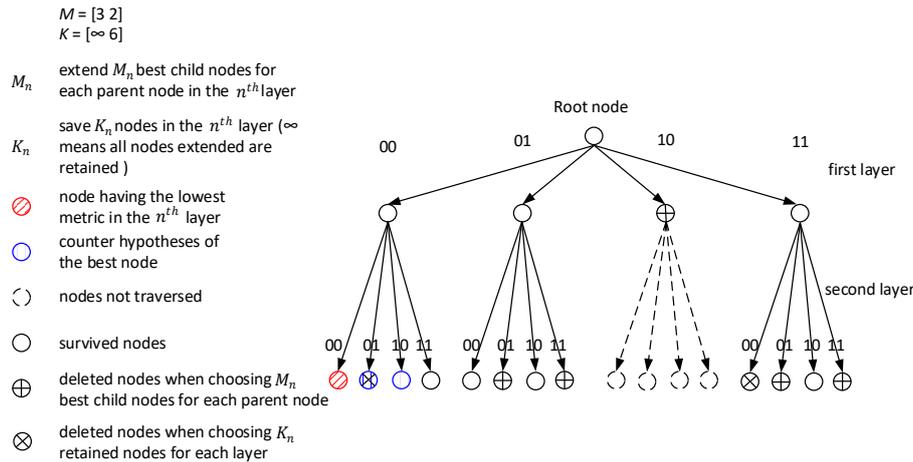


Fig. 8: An example of SOCA tree search with $N_t = 2$ and employing QPSK modulation.

Reduced-complexity detection may be achieved by reordering the spatial streams arriving from the transmit antennas according to the SNR [4]. In this case, the Smart Ordered QR Decomposition (SOQRD) technique of [17] is an attractive method for reordering spatial streams. More specifically, if the layer of the tree emanating from the root node corresponds to a spatial stream having a high effective SNR, then a complexity reduction may be attained. To elaborate further, it may be assumed that the spatial streams having the highest SNR will rarely suffer demodulation errors and so we may reduce the number of child nodes that are explored in this layer and prune a significant portion of the tree at an early stage in the algorithm, hence reducing its complexity. Let us consider the example where the first layer of the tree search in Figure 8 corresponds to the spatial stream transmitted from the second transmit antenna. Correspondingly, in this case, the second layer of the tree search corresponds to the spatial stream transmitted from the first transmit antenna. In this case, the candidates obtained would be ordered according to

$$\mathcal{X} = \begin{matrix} \text{second layer} \\ \text{second layer} \\ \text{first layer} \\ \text{first layer} \end{matrix} \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (13)$$

Furthermore, the authors of [30] have proven that performance vs. complexity trade-off can be achieved by using Minimum Mean Square Error (MMSE) based pre-processing, which takes the noise in the received signal into account. This may be achieved using the QR decomposition (QRD) of the extended channel matrix $\overline{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \sigma \mathbf{I}_{N_t} \end{bmatrix}$, where $\sigma^2 = N_t N_0 / E_s$.

By employing the MMSE-SOQRD in the SOCA tree search, the input-output relationship of Eq. (9) becomes

$$\hat{\mathbf{r}} = \hat{\mathbf{H}} \tilde{\mathbf{s}} + \hat{\mathbf{n}}, \quad (14)$$

where the inputs of the SOQRD are $\overline{\mathbf{H}}$ and K_1 , and the outputs are \mathbf{Q} , $\begin{bmatrix} \hat{\mathbf{H}} \\ \mathbf{0}_{N_t \times N_t} \end{bmatrix}$ and \mathbf{P} . Here, \mathbf{P} is a permutation matrix, which represents the reordering of the spatial stream as they are mapped to the layers of the tree. Furthermore, these outputs of the SOQRD are employed to transform the input-output relationship of Eq. (9) to Eq. (14), where

$$\hat{\mathbf{r}} = \mathbf{Q}(1 : N_t, 1 : N_t)^H \times \mathbf{r}, \quad \tilde{\mathbf{s}} = \mathbf{P} \mathbf{s}.$$

In addition to the MMSE pre-processing of the received signal \mathbf{r} , the tree search of the SOCA detection benefits from the *a priori* LLRs provided by the LDPC decoder, enabling iterative detection and decoding. More specifically, when extending the g^{th} ($g' \in [1, G]$) child node in the $(n+1)^{\text{th}}$ layer from the $g^{\text{th}} \in [1, G]$ parent node in the n^{th} layer, the node metric may be calculated as

$$\psi_{n+1}^{g,g'} = \frac{\psi_n^g + \left| \hat{\mathbf{r}}_{n+1} - \sum_{1 \leq v \leq n+1} \hat{h}_{n+1,v} \tilde{s}_v^{g'} \right|^2}{8(n+1)+3} - \prod_{i=1}^{\omega} \frac{\exp((1 - \hat{c}_i) \tilde{c}'^a(\hat{c}_i))}{1 + \exp(\tilde{c}'^a(\hat{c}_i))} - \sigma^2 \left\| \tilde{s}_v^{g'} \right\|^2, \quad (15)$$

where, $n \in [0, N_t - 1]$ and $\psi_0^g = 0$ in the case of the root node. Note that Eq. (15) is annotated with the associated complexity matrix, which will be discussed in Section IV.

The node metrics quantified by the squared Euclidean distance of Eq. (15) are employed in the SOCA tree search to evaluate the child nodes in each layer and to obtain the candidate set \mathcal{X} for the encoded bits $\hat{\mathbf{c}}$, including the most likely candidate and its counter-hypothesis, as discussed above. Following this, the *extrinsic* LLR vector $\tilde{\mathbf{c}}^e$ can be calculated using Eq. (11) according to the inputs of the SOCA detector, namely the channel information (\mathbf{H}, N_0) , the received signal \mathbf{r} and the *a priori* LLR vector $\tilde{\mathbf{c}}^a$ provided by the LDPC decoder. Furthermore, the corrected *extrinsic* LLR vector $\tilde{\mathbf{c}}^e$ is obtained according to the online LLR correction discussed in Section II-B2.

IV. COMPLEXITY ANALYSIS

In this section, we quantify the complexity of the MIMO detector for the case of SOCA detection without DL-aided LLR correction and for the case, where the DL-aided LLR correction is employed. The LDPC complexity is not considered here, because we employ the same LDPC decoder in all schemes.

The complexity of each part in the SOCA detector algorithm may be quantified by the number of additions, multiplications, divisions, and exponential calculations performed.

Here, each complex-valued multiplication may be represented by four real-valued multiplications and two real-valued additions. Moreover, each complex addition may be considered to comprise two real-valued additions.

The SOCA detection algorithm comprises two parts, namely the tree search and the LLR calculation. In the first part, the complexity of tree search is incurred when calculating the node metric. More specifically, the complexity associated with the g^{th} path in the $(n+1)^{th}$ ($n \in [0, N_t - 1]$) layer in Eq. (15) may be expressed as

$$C_{n+1} = 8n + 7\omega + 17, \quad (16)$$

where ω is the number of bits per symbol.

The complexity of the second part of the SOCA detection, which calculates an *extrinsic* LLR using Eq. (11), may be expressed as

$$C_{LLR} = K_{N_t} \times (6N_r N_t + 6N_r + \omega N_t + 1), \quad (17)$$

where K_{N_t} is the number of surviving nodes in the final layer of the tree search.

Based on the results of Eq. 16 and Eq. 17, the overall complexity of performing SOCA detection for a conventional SOCA-LDPC scheme, which does not employ our DL technique for correcting the LLRs may be calculated as

$$\begin{aligned} c_{OS} = & \left\{ 2^\omega \times c_1 + \frac{K_1 \times (2^{\omega+1} - K_1 - 1)}{2} \right\} \\ & + \left[K_2 \times 2^\omega \times c_2 + K_2 \times (2^\omega - 1) + \frac{K_2 \times (K_2 + 2\omega - 1)}{2} \right] \\ & \vdots \\ & + \left[K_{n+1} \times 2^\omega \times c_{n+1} + K_{n+1} \times (2^\omega - 1) + \frac{K_{n+1} \times (K_{n+1} + 2\omega - 1)}{2} \right] \\ & \vdots \\ & + \left[K_{N_t} \times 2^\omega \times c_{N_t} + K_{N_t} \times (2^\omega - 1) + \frac{K_{N_t} \times (K_{N_t} + 2\omega - 1)}{2} \right] \times \left(\frac{K'}{2^\omega R} \right) \\ & + C_{LLR} \times \frac{K'}{R}. \end{aligned} \quad (18)$$

Here, each row of Eq. 18 apart from the last one corresponds to the calculation complexity incurred in each tree layer, while the last row corresponds to the complexity of calculating the LLRs.

Furthermore, the complexity incurred in each iteration when employing DL-aided SOCA detection may be obtained as the sum of the complexity of employing the SOCA detection from Eq. 18 and the complexity of employing the DL from Eq. (8), which is expressed as:

$$C_{DL-SOCA} = C_{OS} + C_{DL}. \quad (19)$$

By way of comparison, the complexity associated with employing the FS-Net-aided scheme proposed in [20] may be expressed as

$$C_{FS-Net} = 2N_t(4N_r - 1) + 4N_t^2(2N_t - 1) + L_F(8N_r^2 + 10N_r), \quad (20)$$

where L_F is the number of layers in the FS-Net, which is set to 2 in this paper.

Hence, the overall complexity of employing the FS-Net SOCA scheme in this work is given by

$$C_{FS-Net-SOCA} = C_{OS} + C_{FS-Net}. \quad (21)$$

Table III characterizes the complexity of the four schemes for the case of $M_1 = 4$, $M_1 = 8$ and $M_1 = 16$, respectively.

TABLE III: Average number of operations of different schemes when $E_b/N_0 = 10$ dB (in millions).

	$M_1 = 4$	$M_1 = 8$	$M_1 = 16$
conventional scheme	8.26	16.25	32.22
FS-Net scheme	8.53	16.14	30.14
LUT scheme	9.90	17.92	34.00
DL-aided scheme	9.55	17.54	33.52

In all cases, it may be seen that doubling the value of M_1 roughly doubles the associated complexity. Note that, the addition of the DL-aided and LUT-aided LLR correction does not significantly increase the complexity of the associated schemes compared to the conventional SOCA-LDPC scheme, regardless of M_1 value.

When $M_1 = 4$, employing DL-aided LLR correction increases the complexity by about 16%, while the complexity increase associated with the LUT-aided LLR correction is about 20%. However, LUT-aided LLR correction has the additional cost of requiring LUTs for all the different numbers of iterations and different E_b/N_0 to be stored, which necessitates a large amount of memory. By contrast, the DL-aided LLR correction scheme only requires the storage of a small number of weight matrices and bias vectors. When the LUT stores the LLR corrections for each E_b/N_0 value in the range of $\{8, 8.5, 9, \dots, 12\}$ dB, as well as in the number of iterations in the set $\{1, 2, 3\}$ between the SOCA detector and LDPC decoder, a total of 47 920 bytes of memory is required. By contrast, the DL-aided LLR correction scheme requires only 920 bytes of memory to store the trained weight matrices and bias vectors for the case where $M_1 = 4$. While the FS-Net-aided SOCA-LDPC scheme benefits from a reduced SOCA tree complexity, there is some extra complexity incurred by the FS-Net computation. Hence, the FS-Net-aided SOCA-LDPC scheme has a slightly high complexity than the conventional SOCA-LDPC scheme, which is about 3%.

In the case of $M_1 = 8$, the complexity of the proposed DL-aided LLR correction assisted SOCA-LDPC scheme is about 8% higher than that of the conventional SOCA-LDPC scheme, and it requires 920 bytes of memory for storing the trained weight matrices and bias vectors. By contrast, the LUT-aided LLR correction assisted SOCA-LDPC scheme requires 35 744 bytes of memory to store the LUTs in the case of $M_1 = 8$. The reduced memory compared to $M_1 = 4$ is a benefit of only requiring E_b/N_0 values in a smaller set of $\{8, 8.5, 9, \dots, 10.5\}$ dB. The complexity of the LUT-aided LLR correction assisted SOCA-LDPC scheme is about 10% higher than that of the conventional SOCA-LDPC scheme. Then, in the case of $M_1 = 8$, the complexity of the FS-Net-aided SOCA-LDPC scheme is about 1% lower than that of the conventional SOCA-LDPC scheme.

In the case of $M_1 = 16$, the complexity of the proposed DL-aided LLR correction SOCA-LDPC is about 4% higher than that of the conventional SOCA-LDPC scheme and it has the additional requirement of 920 bytes of memory, as discussed above. By contrast, the LUT-aided LLR correction SOCA-LDPC scheme requires 29 600 bytes of memory to store the LUTs for each E_b/N_0 value in the set of $\{8, 8.5, 9, \dots, 10\}$ dB, as well as in the number of iterations in the set $\{1, 2, 3\}$. The complexity of the LUT-aided scheme is about 6% higher compared to the conventional SOCA-LDPC scheme. As seen in Table III, the complexity of the FS-Net-aided SOCA-LDPC scheme is about 7% lower than that of

the conventional SOCA-LDPC scheme when $M_1 = 16$, but the performance is degraded by about 1.2 dB at BLER of 10^{-3} .

V. SIMULATION RESULTS

In this section, we characterized both the performance and complexity of the proposed DL-aided LLR correction assisted SOCA-LDPC scheme and compared it to various benchmarkers. The density ($d^l, l = 2, 3, 4$) of each hidden layer in the neural network of the proposed scheme and the optimal number of iterations to use between the SOCA detector and the LDPC decoder will be selected according to the simulation results. Here, we employ the 3GPP 5G NR LDPC code, which we decode using the log-sum-product algorithm. Moreover, we will compare the decoding performance of the four different iterative detection and decoding schemes described in Section III-C. More specifically, the proposed DL-aided LLR correction SOCA-LDPC scheme employs the trained DNN to correct the raw *extrinsic* LLRs \tilde{c}^e , as described in Section II. Furthermore, there are three benchmarkers, the first of which is a conventional SOCA-LDPC scheme dispensing with LLR correction. The second benchmarker is the LUT-aided LLR correction assisted SOCA-LDPC scheme, which corrects the raw *extrinsic* LLRs \tilde{c}^e using LUTs that are generated using Figure 4. Finally, the third benchmarker is the FS-Net-aided SOCA-LDPC scheme, which has been described in Section I. The use of FS-Net for MIMO detection to estimate the most likely candidate of transmitted MIMO signals was proposed in [19], [20]. We augment this work by combining it with a SOCA detector, in which we may directly delete all tree nodes in each layer of the SOCA tree search having higher node metrics than that of the most likely candidate identified by FS-Net. Hence, the number of surviving tree nodes in the $n^{\text{th}}, 1 \leq n \leq N_t$ layer of the tree may be lower than K_n in the FS-Net-aided SOCA-LDPC scheme.

Throughout this section, we adopt $N_r = 4$ receive antennas, $N_t = 4$ transmit antennas, $K' = 1024$ information bits, a coding rate of $R = 1/2$, an LLR clipping level of $L_{\max} = 6$, $I_{\text{LDPC}} = 20$ iterations inside the LDPC decoder, and 16-QAM for communication over uncorrelated narrow-band Rayleigh fading channels. Here, the number of inner iterations in the LDPC decoder can be determined with the aid of Figure 9. Here, an EXIT chart is a graphical representation used for analyzing the iterative decoding process, specifically focusing on the exchange of *extrinsic* information between the inner and the outer decoder in a concatenated coding scheme [31]. Since the *extrinsic* LLRs produced by the inner decoder become the *a priori* LLRs of the outer decoder and vice versa, it is conventional to swap over the axes of the outer decoder, so that the *a priori* information is plotted on the same axis as the inner decoder's *extrinsic* information and vice versa. Explicitly, Figure 9 is produced by generating \tilde{b}^a artificially and then measure the MI of \tilde{b}^e . The details of generating the LDPC decoder's EXIT function are similar to the processing introduced in [32], [33]. Figure 9 shows that upon increasing the number of LDPC inner iterations I_{LDPC} , the LDPC EXIT function moves downwards, which results in improved decoding performance. Furthermore, diminishing returns are attained beyond $I_{\text{LDPC}} = 20$, which proves that the choice of $I_{\text{LDPC}} = 20$ is reasonable.

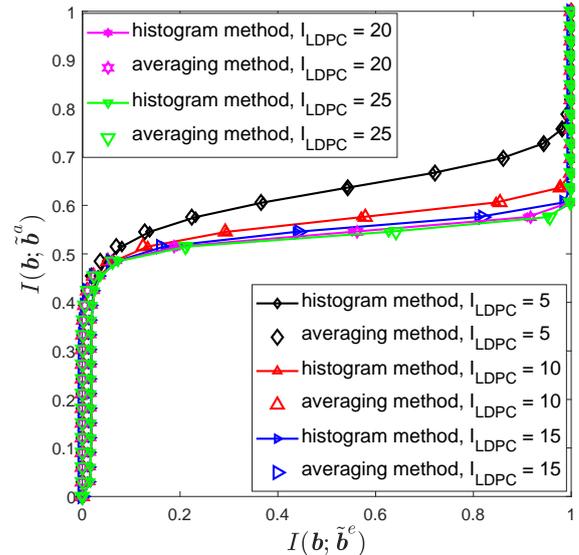


Fig. 9: LDPC EXIT functions, for the case of $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$, $\mathbf{M} = [4, 1, 1, 1]$, $\mathbf{K} = [4, 4, 4, 4]$ and 16-QAM modulation for communication over Rayleigh fading channel.

Figure 10 characterizes the BER performance of the proposed DL-aided LLR correction SOCA-LDPC scheme, when employing different numbers $d^l, (l = 2, 3, 4)$ of neural nodes in each hidden layer. Moreover, the parameters for the SOCA tree search are set as $\mathbf{M} = [4, 1, 1, 1]$ and $\mathbf{K} = [4, 4, 4, 4]$, which indicate that in the first layer, the $M_1 = 4$ child nodes of the root node are extended and all $K_1 = 4$ child nodes are retained. Then for other layers, $M_n = 1, (n = 2, 3, 4)$ child node of each parent node is extended and $K_n = 4$ child nodes are retained after the corresponding counter-hypotheses have been introduced in each layer. In Figure 10, $I = 3$ iterations are performed between the SOCA detector and the LDPC decoder. As characterized in Eq. 8, the complexity of the DL algorithm is influenced by the number of layers L and the density $d^l (l = 1, \dots, L)$ of each layer in the DNN. The performance of the DL may be affected by these two parameters, where it is typically expected that a higher performance is achieved when the density and the number of layers is increased. Therefore, the performance vs. complexity trade-off may be investigated when using different numbers of layers L and different densities in each layer. As shown in Figure 10, increasing the number of hidden layers and the density of each hidden layer may improve the performance, as expected. However, diminishing returns may be observed when having more than 9 hidden layers and more than $d^l = 3$ neuron nodes in each hidden layer. In the case of $L - 2 = 9$ hidden layers and $d^l = 3$ neuron nodes in each hidden layer, each LLR correction requires a total of 210 addition and multiplication operations. By contrast, when using $L - 2 = 3$ hidden layers and $d^l = 6$ neuron nodes in each hidden layer, the total number of addition and multiplication operations for each LLR correction amounts to 211. Both of these cases have very similar complexity, but we observe that having $d^l = 6$ for each hidden layer and $L - 2 = 3$ hidden layers is associated with slightly better performance. Given this performance vs. complexity trade-off, we select a density of $d^l = 6$ for each of the $L - 2 = 3$ hidden layers throughout the following investigations.

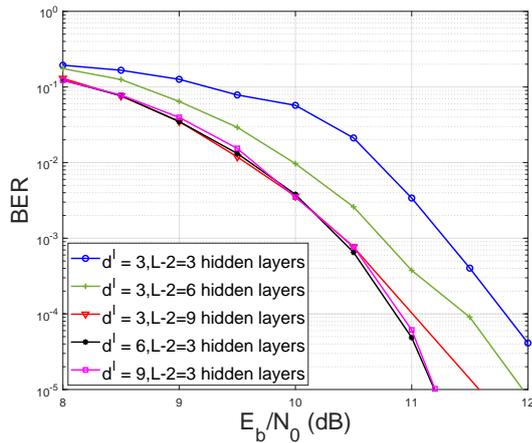
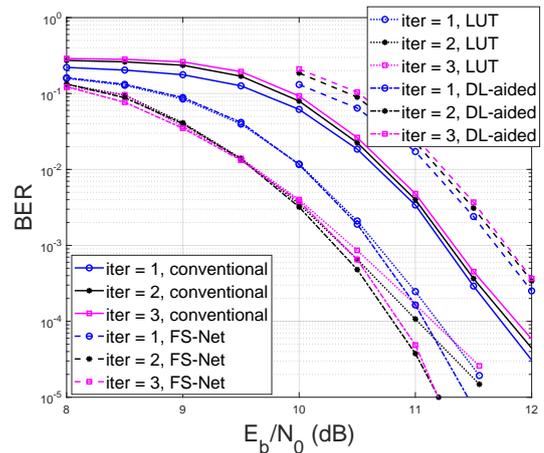
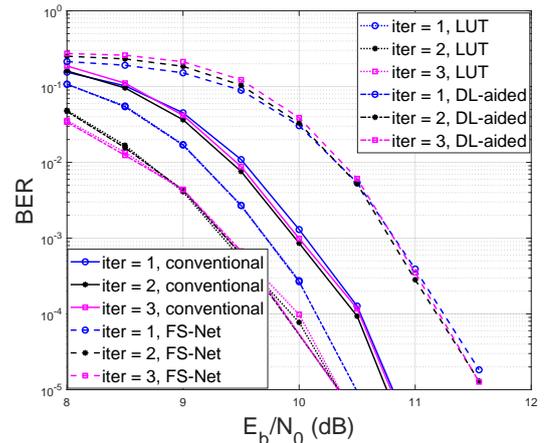


Fig. 10: BER performance of the proposed DL-aided LLR correction SOCA-LDPC scheme having various densities ($d^l, l = 2, 3, 4$) in each of 3 hidden layers, for the case of $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$, $\mathbf{M} = [4, 1, 1, 1]$, $\mathbf{K} = [4, 4, 4, 4]$, $I_{\text{LDPC}} = 20$ iterations inside the LDPC decoding, $I = 3$ iterations between the SOCA detector and the LDPC decoder, and 16-QAM modulation for communication over Rayleigh fading channel.

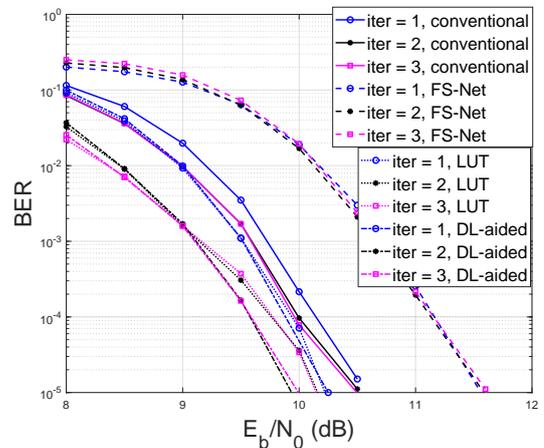
Figure 11(a) characterizes the BER performance of the proposed DL-aided LLR correction SOCA-LDPC scheme and compares it to that of the three benchmarks, for the case of using $\mathbf{M} = [4, 1, 1, 1]$ and $\mathbf{K} = [4, 4, 4, 4]$ during the SOCA tree search in all schemes. Note that, in the FS-Net-aided SOCA-LDPC scheme, the actual number of the surviving tree nodes may be smaller than $K_n, 1 \leq n \leq N_t$, as mentioned above. As detailed in Section II, the selection of a low M_1 value leads to less self-consistent raw *extrinsic* LLRs \tilde{c}'^e at the output of the SOCA detector, which may degrade the decoding performance. As these inconsistent raw *extrinsic* LLRs \tilde{c}'^e are exchanged between the SOCA detector and LDPC decoder, the decoding performance can actually degrade in each successive iteration, as demonstrated by the BER performance of the conventional SOCA-LDPC scheme in Figure 11(a). More specifically, the performance degradation observed for the SOCA detector may be attributed to its use of complexity-reducing mechanisms, such as the early termination of the tree search [34]. As a result, the SOCA detector may generate sub-optimal extrinsic LLRs \tilde{c}'^e , which do not satisfy the self-consistency condition [7] that channel decoders rely upon for maximising their performance [4]. As shown in Figure 3, the self-consistency of the raw extrinsic LLRs \tilde{c}'^e may be improved by increasing the value of the SOCA detector's M_1 parameter, leading to the elimination of performance degradation in successive iterations, as shown in Figure 11(c). However, the proposed DL-aided LLR correction eliminates the performance degradation caused by the inconsistent LLRs, accordingly, the BER performance improves after each successive iteration, as shown in Figure 11(a). Furthermore, the LUT-aided LLR correction assisted SOCA-LDPC scheme can also be seen to eliminate performance degradation. However, the decoding performance of the FS-Net-aided SOCA-LDPC scheme shows about 0.5 dB BER degradation compared to the conventional SOCA-LDPC scheme. This may be attributed to the FS-Net failing to correctly identify the most likely decoding candidate and the SOCA detector being unable to remedy this owing to its reduced-complexity. Furthermore, the FS-Net-aided benchmark does not benefit from any LLR



(a) $M_1 = 4$



(b) $M_1 = 8$



(c) $M_1 = 16$

Fig. 11: BER performance of the proposed DL-aided LLR correction SOCA-LDPC scheme and the various benchmarks, for the case of $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$, $\mathbf{M} = [M_1, 1, 1, 1]$, $\mathbf{K} = [M_1, M_1, M_1, M_1]$, $I_{\text{LDPC}} = 20$ iterations inside the LDPC decoding, and 16-QAM modulation for communication over Rayleigh fading channel.

correction.

Figure 11(b) characterizes the BER performance of the four schemes, for the case of using $\mathbf{M} = [8, 1, 1, 1]$ and $\mathbf{K} = [8, 8, 8, 8]$ during the SOCA tree search. When M_1 is increased to $M_1 = 8$, more candidates are considered during the SOCA tree to search and when calculating the LLRs, this requests in improved self-consistency. As a result, the performance of the

conventional SOCA-LDPC scheme observed after deleterious iterations between the SOCA detector and the LDPC decoder may be eliminated. However, no significant performance gains are achieved after successive iterations of the conventional SOCA-LDPC scheme and a performance gap of about 0.15 dB may be observed in Figure 11(b), relative to the proposed DL-aided LLR correction assisted SOCA-LDPC scheme and to the LUT-aided benchmark. To elaborate further, the DL-aided LLR correction assisted SOCA-LDPC scheme and the LUT-aided benchmark can achieve performance gains in successive iterations between the SOCA detector and the LDPC decoder, as a benefit of using the corrected *extrinsic* LLRs \tilde{c}^e . In the case of $M_1 = 8$, the FS-Net-aided SOCA-LDPC benchmark still has worse performance than the conventional SOCA-LDPC scheme, where a performance gap can be observed in Figure 11(b).

Figure 11(c) shows the BER performance of the four schemes, for the case of using $\mathbf{M} = [16 \ 1 \ 1 \ 1]$ and $\mathbf{K} = [16 \ 16 \ 16 \ 16]$ during the SOCA tree search. When employing $M_1 = 16$, the performance gain offered by the proposed DL-aided LLR correction assisted SOCA-LDPC scheme is about 0.25 dB when employing three iterations instead of one at a BER of 10^{-5} . While the conventional SOCA-LDPC scheme achieves some iteration gains in the case of $M_1 = 16$, only limited gains of less than 0.1 dB are attained after the second iteration. The LUT-aided LLR correction scheme can be seen to offer a similar iteration gain to the conventional SOCA-LDPC scheme. As shown in Figure 11(c), the FS-Net-aided SOCA-LDPC scheme still offers no BER performance BER improvement upon employing iterations between the SOCA detector and LDPC decoder for a larger M_1 . In fact, it suffers from about 1.6 dB performance loss compared to the proposed DL-aided LLR correction assisted SOCA-LDPC scheme.

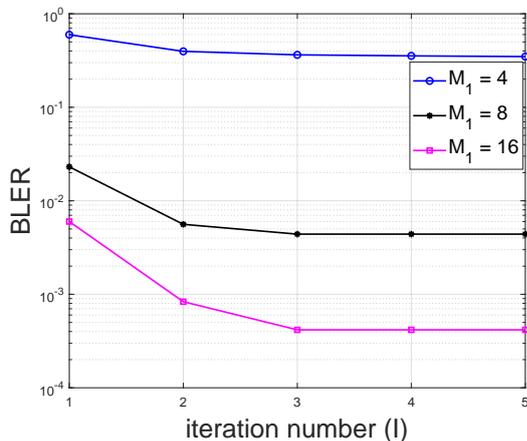


Fig. 12: BLER performance of the conventional SOCA-LDPC employing the various number of iterations between the SOCA detector and LDPC decoder, for the case of $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$ and $E_b/N_0 = 10$, $\mathbf{K} = [M_1 \ M_1 \ M_1 \ M_1]$, $I_{\text{LDPC}} = 20$ iteration inside the LDPC decoding, and 16-QAM modulation for communication over uncorrelated narrow-band Rayleigh fading channels.

In contrast to the BER performance, the BLER performance of the conventional SOCA-LDPC scheme does improve in successive iterations between the SOCA detector and the LDPC decoder for both small and large M_1 values, as shown Figure 12. This demonstrates that while the prevalence of

block errors is reduced upon increasing the number of decoding iterations, the number of bit errors within those erroneous blocks increases in successive iterations. This phenomenon may be attributed to the LDPC decoder's convergence towards an incorrect solution in the case of these erroneous blocks, with successive iterations increasingly deviating from the original block. Significant iteration gains may be observed in Figure 12 up to 3 iterations. However, diminishing returns are observed beyond 3 iterations. Given the trade-off between the performance and complexity, we offer $I = 3$ iterations in the following investigation for all the schemes considered.

Figure 13(a) characterizes the BLER performance of the four schemes for the case of $\mathbf{M} = [4 \ 1 \ 1 \ 1]$ and $\mathbf{K} = [4 \ 4 \ 4 \ 4]$. It may be observed that the proposed DL-aided LLR correction assisted SOCA-LDPC scheme offers a 0.9 dB performance gain at a BLER of 10^{-3} , compared to the conventional SOCA-LDPC scheme. This BLER vs. SNR gain may be attributed to the proposed DL-aided LLR correction of the *extrinsic* LLRs \tilde{c}^e . By way of comparison, the LUT-aided LLR correction scheme offers about 0.7 dB of gain at a BLER of 10^{-3} , compared to the conventional SOCA-LDPC scheme. Here, the proposed DL-aided LLR correction scheme achieves 0.2 dB better performance than the LUT-aided LLR correction scheme because of deviations from the ideal correction factor during the linear interpolation between the values stored in the LUT. Although linear interpolation is employed in the training data generation of the DL-aided LLR correction scheme, the relationship between the raw *extrinsic* LLRs \tilde{c}'^e and the corrected *extrinsic* LLRs \tilde{c}^e are learned in the DNN. This allows the near-ideal correction of each raw *extrinsic* LLR \tilde{c}'^e during the operation of the proposed DL-aided LLR correction SOCA-LDPC scheme. The FS-Net-aided SOCA-LDPC scheme suffers from a performance degradation of about 0.4 dB at a BLER of 10^{-3} compared to the conventional SOCA-LDPC scheme.

Figure 13(b) characterizes the BLER performance of the four schemes for the case of $\mathbf{M} = [8 \ 1 \ 1 \ 1]$ and $\mathbf{K} = [8 \ 8 \ 8 \ 8]$. The proposed DL-aided LLR correction assisted SOCA-LDPC scheme can be seen to offer about 0.6 dB gain compared to the conventional SOCA-LDPC scheme at a BLER of 10^{-3} , while the LUT-aided LLR correction assisted SOCA-LDPC scheme offers about 0.55 dB gain. Figure 13(b) shows that the performance degradation of the FS-Net-aided SOCA-LDPC scheme becomes more pronounced at $M_1 = 8$, compared to $M_1 = 4$. This may be attributed to having high node metrics compared to the most likely candidate and being deleted in the SOCA tree search when $M_1 = 8$. More specifically, when more leaf nodes are deleted in the SOCA tree search, fewer candidates are available for the LLR calculation, which degrades the FS-Net-aided scheme's ability to generate self-consistent LLR values. As seen in Figure 13(a), Figure 13(b), and Table III, the proposed DL-aided scheme offers about 0.9 dB gain compared to the conventional SOCA scheme when $M_1 = 4$, at the cost of a complexity increase of around 16%. This may be compared to the observation that increasing M_1 in the conventional SOCA scheme to 8 offers around 1.3 dB of gain compared to the conventional SOCA scheme using $M_1 = 4$, at the cost of roughly doubling its complexity. Furthermore, from a practical hardware implementation perspective, in order to improve the

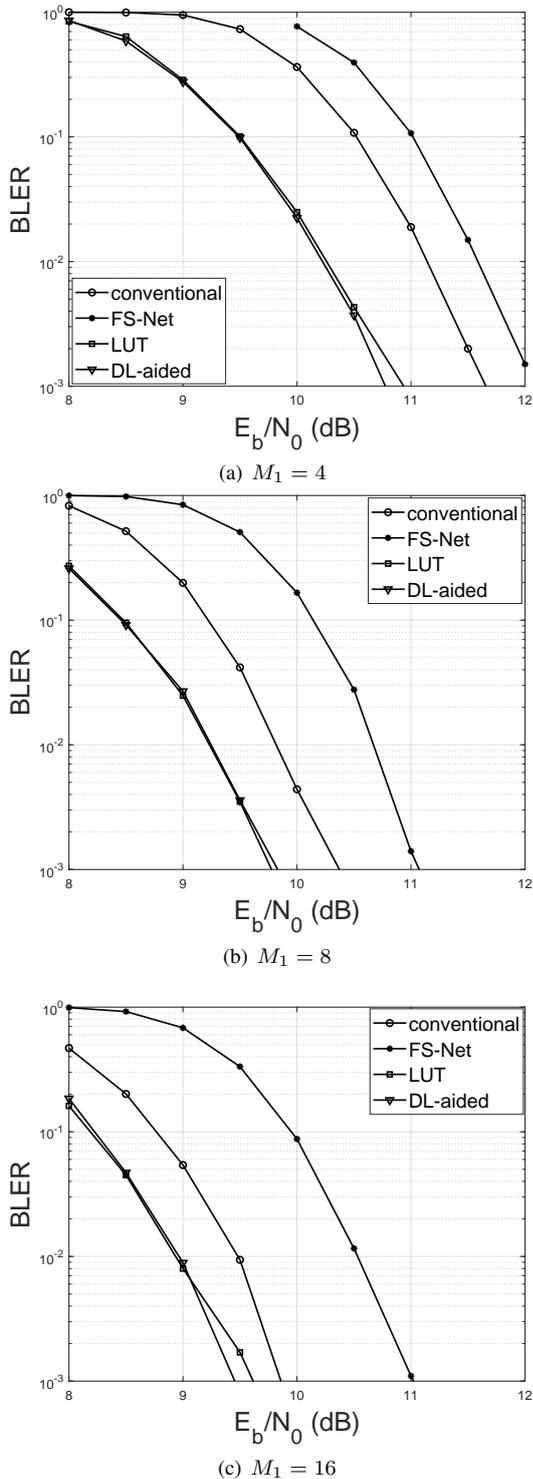


Fig. 13: BLER performance of the proposed DL-aided LLR correction SOCA-LDPC scheme and the various benchmarks, for the case of $N_t = 4$ transmit antennas, $N_r = 4$ receive antennas, $K' = 1024$ information bits, a coding rate of $R = \frac{1}{2}$, an LLR clipping level of $L_{\max} = 6$, $\mathbf{M} = [M_1, 1, 1, 1]$, $\mathbf{K} = [M_1 \ M_1 \ M_1 \ M_1]$, $I_{\text{LDPC}} = 20$ iterations inside the LDPC decoding, $I = 3$ iterations between the SOCA detector and the LDPC decoder, and 16-QAM modulation for communication over Rayleigh fading channel.

performance of the conventional SOCA detector by changing the M_1 parameter, an entirely new SOCA detector implementation would be required, which represents a substantial design effort compared to training the proposed DL scheme.

Figure 13(c) characterizes the BLER performance of the four schemes for the case of $\mathbf{M} = [16 \ 1 \ 1 \ 1]$ and $\mathbf{K} = [16 \ 16 \ 16 \ 16]$. When $M_1 = 16$, the proposed DL-aided and

LUT-aided LLR correction assisted SOCA-LDPC schemes to offer gains of about 0.4 and 0.25 dB, respectively, compared to the conventional SOCA-LDPC scheme at a BLER of 10^{-3} . As the M_1 value is increased to 16, more candidates are considered for calculating the LLRs, and hence more self-consistent raw *extrinsic* LLRs \tilde{c}^e are produced. As a result, the LLR correction has a lower influence when $M_1 = 16$. This may also be demonstrated by comparing the gaps between the claimed and measured MI values in Figure 3 and Figure 6. In the absence of LLR correction, Figure 3 shows that as the value of M_1 is increased, the gap between the claimed and measured MI values becomes small, which represents an improved self-consistency of the LLRs. Figure 6 shows that in the presence of LLR correction, the gap is also reduced as the value of M_1 is increased, but much less dramatically than in Figure 3. Hence, LLR correction has less benefit, when the value of M_1 is high. As shown in Figure 13(c), the FS-Net-aided SOCA-LDPC scheme still suffers from a performance degradation compared to the conventional SOCA-LDPC scheme, when $M_1 = 16$.

VI. SUMMARY AND CONCLUSION

A DL-aided LLR correction scheme was proposed, which is capable of correcting any inconsistency in the LLRs that are generated by reduced-complexity decoders in an iterative receiver. We have presented MI results, which show that the proposed DL-aided LLR correction significantly improves the self-consistency of the LLRs. The proposed technique is characterized in the context of an iterative SOCA-LDPC scheme, which is compared to three benchmarks. The first benchmarker is provided by a conventional SOCA-LDPC scheme, which does not employ any LLR correction. The second benchmarker is a LUT-aided LLR correction assisted SOCA-LDPC scheme, which uses LUTs to correct the LLR values. The final benchmarker is provided by a FS-Net-aided SOCA-LDPC scheme, which employs the FS-Net of [20] to estimate the most likely candidate during signal detection using a trained DNN. More specifically, we adopt the most likely candidate identified by the FS-Net in order to prune the SOCA tree and reduce the complexity.

We have presented both BER and BLER results, which demonstrate that improved performance is attained by adopting the proposed DL-aided LLR correction and by the LUT-aided LLR correction assisted SOCA-LDPC schemes. The corrected *extrinsic* LLRs \tilde{c}^e are shown to significantly improve the decoding performance using both the proposed DL-aided LLR correction scheme and the LUT-aided LLR correction assisted SOCA-LDPC schemes. In the case, where the low complexity SOCA detector is parameterized by $M_1 = 4$, the proposed DL-aided LLR correction assisted SOCA-LDPC scheme achieves about 0.9 dB gain over the conventional SOCA-LDPC scheme, at the cost of 16% increased complexity. When the complexity of the SOCA detector is increased using $M_1 = 8$ and $M_1 = 16$, the proposed DL-aided LLR correction assisted SOCA-LDPC scheme achieves about 0.5 dB and 0.4 dB of gain, at the cost of increasing the complexity by 8% and by 4%, respectively. The results show that the proposed DL-aided LLR correction assisted SOCA-LDPC scheme always outperforms the LUT-aided SOCA-LDPC LLR correction assisted scheme. Furthermore, the memory requirement of the proposed DL-aided LLR correction SOCA-LDPC scheme is about 97%

lower than that of the LUT-aided LLR correction assisted SOCA-LDPC scheme, which requires excessive memory to store its LUTs. The complexity of the FS-Net-aided SOCA-LDPC benchmarker is about 1% and 7% lower than that of the conventional SOCA-LDPC scheme for $M_1 = 8$ and $M_1 = 16$, respectively. However, our results show that the FS-Net-aided scheme suffers from significant BER and BLER performance degradation compared to the conventional SOCA-LDPC scheme. In future work, the proposed DL-aided LLR correction scheme may be combined with a soft-input and soft-output DNN network, as a replacement for the SOCA detector.

REFERENCES

- [1] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.
- [2] Z. B. K. Egilmez, L. Xiang, R. G. Maunder, and L. Hanzo, "A soft-input soft-output polar decoding algorithm for turbo-detection in MIMO-aided 5G New Radio," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6454–6468, 2022.
- [3] X. Wu, Y. Song, M. Jiang, and C. Zhao, "Adaptive-normalized/offset min-sum algorithm," *IEEE Communications Letters*, vol. 14, no. 7, pp. 667–669, 2010.
- [4] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4827–4842, 2010.
- [5] L. Szczecinski, "Correction of mismatched L-values in BICM receivers," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3198–3208, 2012.
- [6] S. Ning, "Improving resistive RAM hard and soft decision correctable BERs by using improved-LLR and reset-check-reverse-flag concatenating LDPC code," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 10, pp. 2164–2168, 2020.
- [7] C. Xu, X. Zuo, S. X. Ng, R. G. Maunder, and L. Hanzo, "Reduced-complexity soft-decision multiple-symbol differential sphere detection," *IEEE Transactions on Communications*, vol. 63, no. 9, pp. 3275–3289, 2015.
- [8] F. Alberge, "On some properties of the mutual information between extrinsics with application to iterative decoding," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1541–1553, 2015.
- [9] J. Tan, Y. Xiao, C. Wu, and W. Tang, "Accurate log-likelihood ratio calculation for vector perturbation precoding," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6272–6276, 2021.
- [10] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.
- [11] L. Liu, L. Cai, L. Ma, and G. Qiao, "Channel state information prediction for adaptive underwater acoustic downlink OFDMA system: Deep neural networks based approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9063–9076, 2021.
- [12] J. Dai, K. Niu, and J. Lin, "Iterative Gaussian-approximated message passing receiver for MIMO-SCMA system," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 3, pp. 753–765, 2019.
- [13] K. Takeuchi and S. Horio, "Iterative multiuser detection and decoding with spatially coupled interleaving," *IEEE Wireless Communications Letters*, vol. 2, no. 6, pp. 619–622, 2013.
- [14] M. Torlak and G. Xu, "Maximum likelihood detection of co-channel communication signals exploiting the spatio-temporal diversity," in *Conference Record of The Thirtieth Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1996, pp. 728–732 vol.1.
- [15] L. Xiao, P. Yang, Y. Xiao, S. Fan, M. Di Renzo, W. Xiang, and S. Li, "Efficient compressive sensing detectors for generalized spatial modulation systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1284–1298, 2017.
- [16] L. Dai, X. Gao, X. Su, S. Han, C.-L. I, and Z. Wang, "Low-complexity soft-output signal detection based on Gauss–Seidel method for uplink multiuser large-scale mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4839–4845, 2015.
- [17] D. L. Milliner, E. Zimmermann, J. R. Barry, and G. Fettweis, "A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 6, pp. 1016–1025, 2009.
- [18] D. L. Milliner, *Low-complexity list detection algorithms for the multiple-input multiple-output channel*. Georgia Institute of Technology, 2009.
- [19] N. T. Nguyen and K. Lee, "Deep learning-aided tabu search detection for large MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4262–4275, 2020.
- [20] N. T. Nguyen, K. Lee, and H. Dai, "Application of deep learning to sphere decoding for large MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6787–6803, 2021.
- [21] A. Movahed, M. C. Reed, N. Aboutorab, and S. E. Tajbakhsh, "EXIT chart analysis of turbo compressed sensing using message passing dequantization," *IEEE Transactions on Signal Processing*, vol. 64, no. 24, pp. 6600–6612, 2016.
- [22] Z. Kong, S. Yang, Z. Cai, and D. Guan, "A clustering-ensemble approach based on average mutual information," in *2008 IEEE International Conference on Networking, Sensing and Control*, 2008, pp. 1692–1695.
- [23] D. W. SCOTT, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 12 1979. [Online]. Available: <https://doi.org/10.1093/biomet/66.3.605>
- [24] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4368–4378, 2019.
- [25] S. Roger, A. Gonzalez, V. Almenar, and G. Matz, "An efficient fixed-complexity sphere decoder with quantized soft outputs," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1828–1831, 2012.
- [26] G. Papa, D. Ciuonzo, G. Romano, and P. Salvo Rossi, "A dominance-based soft-input soft-output MIMO detector with near-optimal performance," *IEEE Transactions on Communications*, vol. 62, no. 12, pp. 4320–4335, 2014.
- [27] R. H. Gohary and T. J. Willink, "On LLR clipping in BICM-IDD non-coherent MIMO communications," *IEEE Communications Letters*, vol. 15, no. 6, pp. 650–652, 2011.
- [28] S. Papaharalabos and F. Lazarakis, "Approximated Box-Plus decoding of LDPC codes," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2074–2077, 2015.
- [29] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389–399, 2003.
- [30] E. Zimmermann and G. Fettweis, "Unbiased MMSE tree search detection for multiple antenna systems," in *Proceedings of the International symposium on wireless personal multimedia communications (WPMC'06)*. San Diego, USA, 2006.
- [31] M. El-Hajjar and L. Hanzo, "EXIT charts for system design and analysis," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 127–153, 2014.
- [32] J. Chen, S. Shao, T.-Y. Wang, J.-Y. Wu, C.-P. Li, S. X. Ng, R. G. Maunder, and L. Hanzo, "LDPC coded compressive sensing for joint source-channel coding in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022.
- [33] J. Chen, T.-Y. Wang, J.-Y. Wu, C.-P. Li, S. X. Ng, R. G. Maunder, and L. Hanzo, "Tree-search techniques for joint iterative compressive sensing and LDPC decoding in wireless sensor networks," *IEEE Sensors Journal*, vol. 23, no. 1, pp. 434–451, 2023.
- [34] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and vlsi implementation," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008.