

Graph Neural Network Aided Expectation Propagation Detector for MU-MIMO Systems

Alva Kosasih*, Vincent Onasis*, Wibowo Hardjawana*, Vera Miloslavskaya*,
Victor Andrean†, Jenq-Shiou Leu†, Branka Vucetic*

*Centre of Excellence in Telecommunications, The University of Sydney, Sydney, Australia.

†Mobilizing Information Technology Lab., National Taiwan University of Science and Technology, Taipei, Taiwan.

Email: {alva.kosasih, wibowo.hardjawana, vera.miloslavskaya, branka.vucetic}@sydney.edu.au
vona0880@uni.sydney.edu.au, andreanvictor6374@gmail.com, jsleu@mail.ntust.edu.tw.

Abstract—Multiuser massive multiple-input multiple-output (MU-MIMO) systems can be used to meet high throughput requirements of 5G and beyond networks. In an uplink MU-MIMO system, a base station is serving a large number of users, leading to a strong multi-user interference (MUI). Designing a high performance detector in the presence of a strong MUI is a challenging problem. This work proposes a novel detector based on the concepts of expectation propagation (EP) and graph neural network, referred to as the GEPNet detector, addressing the limitation of the independent Gaussian approximation in EP. The simulation results show that the proposed GEPNet detector significantly outperforms the state-of-the-art MU-MIMO detectors in strong MUI scenarios with equal number of transmit and receive antennas.

Index Terms—MU-MIMO detector, graph neural network, expectation propagation, beyond 5G

I. INTRODUCTION

Multiuser massive multiple-input multiple-output (MU-MIMO) technique is one of the key technologies to enable a high throughput in 5G and beyond networks [1]. The usage of multiple transmit and receive antennas ensures a high spectral efficiency [2], and therefore a high throughput. One of the challenging problems in uplink MU-MIMO systems is to design a practical base station detector that can achieve a high reliability performance in the presence of a strong multi-user interference (MUI). The MUI is caused by multiple user antennas simultaneously sending information to multiple base station antennas. The state-of-the-art practical MU-MIMO detectors can be classified as classical and neural network (NN)-based detectors.

The classical detectors [3]–[6] use Gaussian distributions to approximate the posterior probability of the transmitted symbol estimates conditioned on the received signal. They were shown to achieve a near maximum likelihood (ML) performance [7] only when the number of receive antennas is much higher than the number of transmit antennas (users). The approximate message passing (AMP) detector [3] performs poorly in the case of ill-conditioned channel matrices. The problem of ill-conditioned channel matrices has been partially resolved by the orthogonal AMP (OAMP) detector [5] by integrating the linear minimum mean square error (MMSE) filtering. The expectation propagation (EP) detectors [4], [6]

outperform the OAMP detector by introducing regularization parameters in the MMSE filter that are adjusted iteratively according to the channel matrix and MUI level. However, there is still a significant performance gap between the EP and ML detectors when the number of base station receive antennas is equal to the number of user transmit antennas, referred to as a high MUI scenario.

The NN-based detectors have been proposed in [8]–[12] to address the performance limitation of the mentioned classical detectors in the case of ill-conditioned channel matrices and/or high MUI. This is done by unfolding their iterations into NN layers and optimizing their parameters. The OAMPNet detector [11] combines the OAMP and NN that has a small number of trainable parameters to deal with the ill-conditioned channel matrices. This results in a significant performance improvement compared to the conventional OAMP detector. A high performance recurrent equivariant (RE)-MIMO detector was proposed in [12]. The RE-MIMO detector unfolds the AMP detector and integrates it with a transformer self-attention network to cancel the high MUI and ensure equivariance under permutations of the user transmit antennas. The addition of the transformer based MUI canceller results in a significant performance improvement compared to the conventional AMP detector. Nevertheless, a significant performance gap remains when comparing the performance of the NN-based and ML detectors in a high MUI scenario [11], [12]. Our analysis shows that the reason is the inaccuracy of the Gaussian approximation. To the best of the authors' knowledge, none of the state-of-the-art detectors address this issue.

In this paper, we propose a novel unfolded NN-based detector for high MUI scenarios, referred to as graph EP network (GEPNet) detector. The proposed detector integrates the EP [6] and graph neural network (GNN) [10] as follows. The EP can be divided into three modules: (1) an observation module, calculating the likelihood function of the transmitted symbols based on the received signal; (2) a Gaussian approximation module, approximating the posterior probability distribution of each transmitted symbol estimate using a Gaussian distribution; and (3) an estimation module, calculating the transmitted symbol estimates. The second module assumes that the joint posterior probability distribution of the transmitted symbol

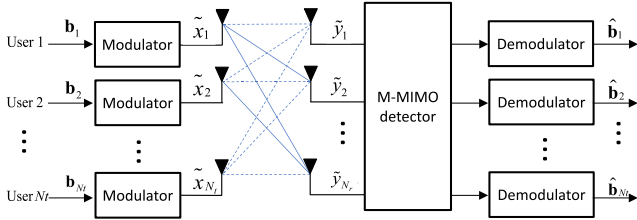


Fig. 1: The MU-MIMO system

estimates is approximated by the product of K independent Gaussian distributions, where K is the number of users. As a consequence, the EP loses some MUI information. In MU-MIMO systems with high MUI, this approximation is therefore inaccurate and induces a severe performance degradation. Instead of using this approximation, the proposed detector uses the GNN to produce the posterior probability distribution parameterized according to a Markov random field (MRF). Specifically, we adopt a factor graph representation [13]. The MUI between each pair of users is characterized by a pair potential. Thus, the GNN captures the MUI information using the MRF. The main contributions of this paper are unfolding EP into NN layers and integrating it with the GNN to address the limitation of the independent Gaussian approximation in EP. This contribution results in the first offline NN-based detector. In contrast to all existing classical [3]–[6] and NN-based [9]–[12] detectors, the proposed GEPNet detector can achieve a high detection performance in a high MUI scenario and significantly improves the EP performance. To the best of the authors' knowledge, the GEPNet is the first detector outperforming the EP by replacing the independent Gaussian approximation. The simulation results show that the GEPNet detector outperforms the EP, OAMPNet, and RE-MIMO detectors by more than 4 dB at the SER of 10^{-4} for 64×64 MU-MIMO configuration.

Notations: \mathbf{I}_n denotes an identity matrix of size n . For any matrix \mathbf{A} , the notations \mathbf{A}^T and \mathbf{A}^\dagger stand for transpose and pseudo-inverse of \mathbf{A} , respectively. $\|\mathbf{q}\|$ denotes the Frobenius norm of vector \mathbf{q} . q^* denotes the complex conjugate of a complex number q . Let $\mathbf{x} = [x_1, \dots, x_K]^T$ and $\mathbf{c} = [c_1, \dots, c_K]^T$. $\mathbb{E}[\mathbf{x}]$ is the mean of random vector \mathbf{x} , and $\text{Var}[\mathbf{x}] = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])^2]$ is its variance. $\mathcal{N}(x_k : c_k, v_k)$ represents a single variate Gaussian distribution for a random variable x_k with mean c_k and variance v_k .

II. SYSTEM MODEL

We consider an uncoded MU-MIMO system used to transmit information streams generated by N_t single-antenna users. The streams are received by a base station, which is equipped with $N_r \geq N_t$ antennas to simultaneously serve the users. The system is depicted in Fig. 1. User k maps $\log_2(\tilde{M})$ bits of its information stream \mathbf{b}_k to a symbol $\tilde{x}_k \in \tilde{\Omega}$ using a quadrature amplitude modulation (QAM) technique, where $\tilde{\Omega} = \{s_1, \dots, s_{\tilde{M}}\}$ is a constellation set of \tilde{M} -QAM and s_m is one of the constellation points. The transmitted symbols are

uniformly distributed, and the corresponding received signal is given by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{n}}, \quad (1)$$

where $\tilde{\mathbf{x}} = [\tilde{x}_1, \dots, \tilde{x}_{N_t}]^T$, $\tilde{\mathbf{y}} = [\tilde{y}_1, \dots, \tilde{y}_{N_r}]^T$, $\tilde{\mathbf{H}} = [\tilde{\mathbf{h}}_1, \dots, \tilde{\mathbf{h}}_{N_t}] \in \mathbb{C}^{N_r \times N_t}$ is the coefficient matrix of complex memoryless Rayleigh fading channels between N_t transmit and N_r receive antennas, $\tilde{\mathbf{h}}_k$ is the k -th column vector of matrix $\tilde{\mathbf{H}}$ that denotes wireless channel coefficients between the receive antennas and the k -th transmit antenna, where each coefficient follows a Gaussian distribution with zero mean and unity variance, and $\tilde{\mathbf{n}} \in \mathbb{C}^{N_r}$ denotes the additive white Gaussian noise (AWGN) with a zero mean and covariance matrix $\sigma^2 \mathbf{I}_{N_r}$. The SNR of the system is defined as $\text{SNR} = 10 \log_{10} \left(\frac{N_t \tilde{E}_s}{\sigma^2} \right)$ dB, where \tilde{E}_s is the energy per transmit antenna. We normalize the total transmit energy so that $N_t \tilde{E}_s = 1$. For convenience, the complex-valued variables are transformed into real-valued variables. Accordingly, we define $\mathbf{x} = [\mathcal{R}(\tilde{\mathbf{x}})^T \ \mathcal{I}(\tilde{\mathbf{x}})^T]^T \in \mathbb{R}^K$, $\mathbf{y} = [\mathcal{R}(\tilde{\mathbf{y}})^T \ \mathcal{I}(\tilde{\mathbf{y}})^T]^T \in \mathbb{R}^N$, $\mathbf{n} = [\mathcal{R}(\tilde{\mathbf{n}})^T \ \mathcal{I}(\tilde{\mathbf{n}})^T]^T \in \mathbb{R}^N$, and $\mathbf{H} = \begin{bmatrix} \mathcal{R}(\tilde{\mathbf{H}}) & -\mathcal{I}(\tilde{\mathbf{H}}) \\ \mathcal{I}(\tilde{\mathbf{H}}) & \mathcal{R}(\tilde{\mathbf{H}}) \end{bmatrix} \in \mathbb{R}^{N \times K}$, where $K = 2N_t$, $N = 2N_r$, $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ are the real and imaginary parts, respectively. Therefore, we can rewrite (1) as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \quad (2)$$

Note that the covariance matrix of \mathbf{n} is $\sigma^2 \mathbf{I}_N \triangleq (\tilde{\sigma}^2/2) \mathbf{I}_N$, the energy per transmit antenna in the real-valued system is $E_s \triangleq \tilde{E}_s/2$, and the real-valued constellation is $\Omega = \{\mathcal{R}(s_m) | s_m \in \tilde{\Omega}\}$ with $|\Omega| = M \triangleq \sqrt{\tilde{M}}$. We consider the system model (2) for the rest of the paper.

III. THE GRAPH EXPECTATION PROPAGATION NETWORK

In this section, we propose the GEPNet detector integrating the EP [6] and GNN [10] schemes. As shown in Fig. 2, the GEPNet detector consists of the observation, GNN and estimation modules, which iteratively exchange the outputs (see Fig. 2a).

A. The Observation Module

The posterior probability distribution of the transmitted symbols conditioned on the received signal in (2) can be expressed as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} \cdot p(\mathbf{x}) \propto \underbrace{\mathcal{N}(\mathbf{y} : \mathbf{H}\mathbf{x}, \sigma^2 \mathbf{I}_{N_r})}_{p(\mathbf{y}|\mathbf{x})} \underbrace{\prod_{k=1}^K p(x_k)}_{p(\mathbf{x})}, \quad (3)$$

where $p(x_k) = \frac{1}{M} \sum_{x \in \Omega} \delta(x_k - x)$ is a priori probability density function of x_k , δ is the Dirac delta function, and $p(\mathbf{y})$ is omitted as it is not related to random variable x_k . A direct calculation of (3) results in an exponential complexity, which

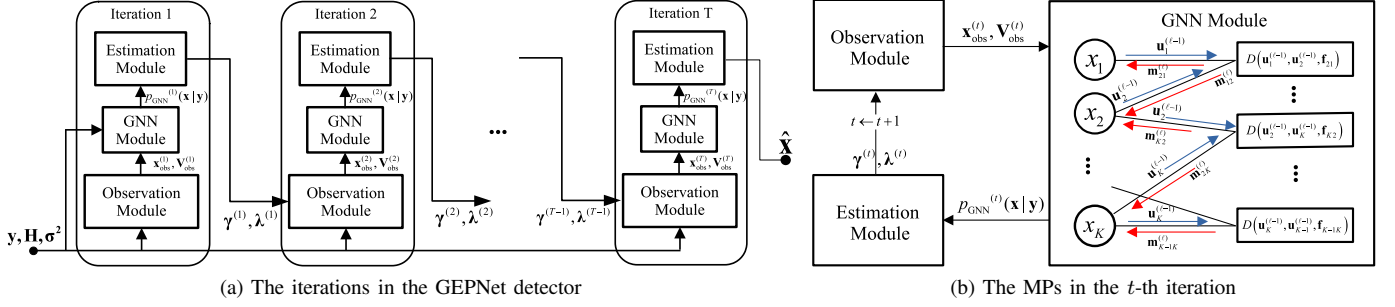


Fig. 2: The GEPNet detector model

is prohibitive. Therefore, the EP scheme is used to approximate $p(\mathbf{x}|\mathbf{y})$ at the t -th iteration by a Gaussian posterior function

$$\begin{aligned} p^{(t)}(\mathbf{x}|\mathbf{y}) &\propto p(\mathbf{y}|\mathbf{x}) \cdot \chi^{(t)}(\mathbf{x}) \\ &\propto \mathcal{N}(\mathbf{x} : \mathbf{H}^\dagger \mathbf{y}, \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1}) \\ &\quad \cdot \mathcal{N}(\mathbf{x} : (\boldsymbol{\lambda}^{(t-1)})^{-1} \boldsymbol{\gamma}^{(t-1)}, (\boldsymbol{\lambda}^{(t-1)})^{-1}) \\ &\propto \mathcal{N}(\mathbf{x} : \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}), \end{aligned} \quad (4)$$

where $\chi^{(t)}(\mathbf{x})$ is an approximation of $p(\mathbf{x})$ obtained from the exponential family [6], $\boldsymbol{\lambda}^{(t)}$ is a $K \times K$ diagonal matrix with diagonal elements $\lambda_k^{(t)} > 0$ and $\boldsymbol{\gamma}^{(t)} = [\gamma_1^{(t)}, \dots, \gamma_K^{(t)}]^T$. Both $\lambda_k^{(t)}$ and $\gamma_k^{(t)}$ are real numbers with $\lambda_k^{(0)} = 1/E_s$ and $\gamma_k^{(0)} = 0$. Note that $p(\mathbf{y}|\mathbf{x})$ in (4) is approximated by treating \mathbf{x} as a random real-valued vector. The product of two Gaussians in (4) is computed by using the Gaussian product property¹, given in Appendix A.1 of [14]. Accordingly, we obtain the variance and mean of $p^{(t)}(\mathbf{x}|\mathbf{y})$ as

$$\boldsymbol{\Sigma}^{(t)} = (\sigma^{-2} \mathbf{H}^T \mathbf{H} + \boldsymbol{\lambda}^{(t-1)})^{-1}, \quad (5a)$$

$$\boldsymbol{\mu}^{(t)} = \boldsymbol{\Sigma}^{(t)} (\sigma^{-2} \mathbf{H}^T \mathbf{y} + \boldsymbol{\gamma}^{(t-1)}). \quad (5b)$$

We then compute the likelihood function $p^{(t)}(\mathbf{y}|\mathbf{x})$ based on the Gaussian posterior function $p^{(t)}(\mathbf{x}|\mathbf{y})$,

$$\begin{aligned} p^{(t)}(\mathbf{y}|\mathbf{x}) &\triangleq \frac{p^{(t)}(\mathbf{x}|\mathbf{y})}{\chi^{(t)}(\mathbf{x})} \\ &\propto \frac{\mathcal{N}(\mathbf{x} : \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})}{\mathcal{N}(\mathbf{x} : (\boldsymbol{\lambda}^{(t-1)})^{-1} \boldsymbol{\gamma}^{(t-1)}, (\boldsymbol{\lambda}^{(t-1)})^{-1})} \\ &\propto \mathcal{N}(\mathbf{x} : \mathbf{x}_{\text{obs}}^{(t)}, \mathbf{V}_{\text{obs}}^{(t)}), \end{aligned} \quad (6)$$

where $\mathbf{x}_{\text{obs}}^{(t)} = [x_{\text{obs},1}^{(t)}, \dots, x_{\text{obs},K}^{(t)}]$ and $\mathbf{V}_{\text{obs}}^{(t)}$ is a $K \times K$ diagonal matrix with $v_{\text{obs},k}^{(t)}$ as the k -th diagonal element,

which can be expressed as

$$v_{\text{obs},k}^{(t)} = \frac{\Sigma_k^{(t)}}{1 - \Sigma_k^{(t)} \lambda_k^{(t-1)}}, \quad (7a)$$

$$x_{\text{obs},k}^{(t)} = v_{\text{obs},k}^{(t)} \left(\frac{\mu_k^{(t)}}{\Sigma_k^{(t)}} - \gamma_k^{(t-1)} \right). \quad (7b)$$

Here, $\mu_k^{(t)}$ is the k -th element of vector $\boldsymbol{\mu}^{(t)}$ and $\Sigma_k^{(t)}$ is the k -th diagonal element of matrix $\boldsymbol{\Sigma}^{(t)}$. We treat the pair $(\mathbf{x}_{\text{obs}}^{(t)}, \mathbf{V}_{\text{obs}}^{(t)})$ from (7) as a prior information for the variable nodes x_1, \dots, x_K in the GNN module, as shown in Fig 2b.

B. The GNN Module

The GNN module employs the message passing (MP) scheme of the pair-wise MRF model [10], as described in the Fig. 2b. The variable and factor nodes of the GNN are displayed as circles and rectangles, respectively. As in a pair-wise MRF, the k -th variable node is characterized by a self potential $\phi(x_k)$, and the (k, j) -th pair of variable nodes is characterized by a pair potential $\psi(x_k, x_j)$, where

$$\phi(x_k) = \exp \left(\frac{1}{\sigma^2} \mathbf{y}^T \mathbf{h}_k x_k - \frac{1}{2} \mathbf{h}_k^T \mathbf{h}_k x_k^2 \right) p(x_k), \quad (8a)$$

$$\psi(x_k, x_j) = \exp \left(-\frac{1}{\sigma^2} \mathbf{h}_k^T \mathbf{h}_j x_k x_j \right). \quad (8b)$$

The GNN is used to infer the posterior probability of the transmitted symbols by using the mean $x_{\text{obs},k}^{(t)}$ and variance $v_{\text{obs},k}^{(t)}$ for the Gaussian approximation of x_k obtained from the observation module, $k = 1, \dots, K$. The mean and variance are concatenated as

$$\mathbf{a}_k^{(t)} = \begin{bmatrix} x_{\text{obs},k}^{(t)} \\ v_{\text{obs},k}^{(t)} \end{bmatrix}, \quad (9)$$

and then $\mathbf{a}_k^{(t)}$ is added as an attribute to the corresponding variable node x_k . The posterior probability corresponding to the pair-wise MRF can be written as [10]

$$p_{\text{GNN}}(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \prod_{k=1}^K \phi(x_k) \prod_{\substack{j=1 \\ j \neq k}}^K \psi(x_k, x_j), \quad (10)$$

¹The product of two Gaussians results in another Gaussian, $\mathcal{N}(\mathbf{x} : \mathbf{a}, \mathbf{A}) \cdot \mathcal{N}(\mathbf{x} : \mathbf{b}, \mathbf{B}) \propto \mathcal{N}(\mathbf{x} : (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1} (\mathbf{A}^{-1} \mathbf{a} + \mathbf{B}^{-1} \mathbf{b}), (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1})$.

where Z is a normalization constant. To compute $p_{\text{GNN}}(\mathbf{x}|\mathbf{y})$ in (10), we use variable and factor feature vectors corresponding to self and pair potentials in (8a) and (8b), respectively. The variable feature vector is denoted as $\mathbf{u}_k^{(\ell)}$. Its initial value is obtained from encoding the information of the received signal, corresponding channel vector, and noise variance according to (8a) using a single layer NN as

$$\mathbf{u}_k^{(0)} = \mathbf{W}_1 \cdot [\mathbf{y}^T \mathbf{h}_k, \mathbf{h}_k^T \mathbf{h}_k, \sigma^2]^T + \mathbf{b}_1, \quad (11)$$

where $\mathbf{W}_1 \in \mathbb{R}^{N_u \times 3}$ is a learnable matrix, $\mathbf{b}_1 \in \mathbb{R}^{N_u}$ is a learnable vector, and N_u is the size of the feature vector. We consider $N_u = 8$. The factor feature vector $\mathbf{f}_{jk} \triangleq [\mathbf{h}_k^T \mathbf{h}_j, \sigma^2]$ is obtained by extracting the pair potential information from (8b). The factor feature vector is used in the MPs of the GNN. As described in Fig. 2b, the initialized feature vectors are sent to the corresponding factor nodes. The factor nodes then commence the following iterative MP between the factor and variable nodes:

1) **Factor to variable:** Each factor node has a multi-layer perceptron (MLP) with two hidden layers of sizes N_{h_1} and N_{h_2} and an output layer of size N_u . In this work, we set $N_{h_1} = 64$ and $N_{h_2} = 32$. The rectifier linear unit (ReLU) activation function is used at the output of each hidden layer. For any pair of variable nodes x_k and x_j , there is a factor node connecting them. This factor node first concatenates the received feature vectors $\mathbf{u}_k^{(\ell-1)}$ and $\mathbf{u}_j^{(\ell-1)}$ with its own feature vector \mathbf{f}_{jk} . The factor node then uses the concatenated features as inputs for its MLP, denoted as \mathbf{D} , and saves the corresponding output, expressed as

$$\mathbf{m}_{jk}^{(\ell)} = \mathbf{D} \left(\mathbf{u}_k^{(\ell-1)}, \mathbf{u}_j^{(\ell-1)}, \mathbf{f}_{jk} \right). \quad (12)$$

Finally, the outputs are fed back to the variable nodes as illustrated in Fig. 2b.

2) **Variable to factor:** The k -th variable node then sums all the incoming messages from its neighbouring factor nodes $\mathbf{m}_{jk}^{(\ell)}$ and concatenates their sum with the node attribute $\mathbf{a}_k^{(t)}$ as $\mathbf{m}_k^{(\ell)} = \left[\sum_{j=1, j \neq k}^K \mathbf{m}_{jk}^{(\ell)}, \mathbf{a}_k^{(t)} \right]$. The concatenated vector is used to compute the node feature vector $\mathbf{u}_k^{(\ell)}$ as

$$\mathbf{g}_k^{(\ell)} = \mathbf{U} \left(\mathbf{g}_k^{(\ell-1)}, \mathbf{m}_k^{(\ell)} \right) \quad (13a)$$

$$\mathbf{u}_k^{(\ell)} = \mathbf{W}_2 \cdot \mathbf{g}_k^{(\ell)} + \mathbf{b}_2, \quad (13b)$$

where function \mathbf{U} is specified by the gated recurrent unit (GRU) network [15], whose current and previous hidden states are $\mathbf{g}_k^{(\ell)} \in \mathbb{R}^{N_{h_1}}$ and $\mathbf{g}_k^{(\ell-1)} \in \mathbb{R}^{N_{h_1}}$, respectively, $\mathbf{W}_2 \in \mathbb{R}^{N_u \times N_{h_1}}$ is a learnable matrix, and $\mathbf{b}_2 \in \mathbb{R}^{N_u}$ is a learnable vector. The updated feature vector (13b) is then sent to the neighbouring factor nodes to continue the MP iterations.

After L rounds of the MP, a readout process yields

$$\tilde{p}_{\text{GNN}}(x_k = a|\mathbf{y}) = \mathbf{R} \left(\mathbf{u}_k^{(L)} \right), a \in \Omega, \quad (14a)$$

$$p_{\text{GNN}}^{(t)}(x_k = a|\mathbf{y}) = \frac{\exp(\tilde{p}_{\text{GNN}}(x_k = a|\mathbf{y}))}{\sum_{b \in \Omega} \exp(\tilde{p}_{\text{GNN}}(x_k = b|\mathbf{y}))}, a \in \Omega. \quad (14b)$$

In this work, we set $L = 2$. The readout function \mathbf{R} is given by an MLP with two hidden layers of sizes N_{h_1} and N_{h_2} , and ReLU activation at the output of each hidden layer. The output size of \mathbf{R} is the cardinality of real-valued constellation set, i.e., M . We then assign

$$\mathbf{g}_k^{(0)} \leftarrow \mathbf{g}_k^{(L)} \text{ and } \mathbf{u}_k^{(0)} \leftarrow \mathbf{u}_k^{(L)}, k = 1, \dots, K \quad (15)$$

in order to use the GRU hidden state and variable feature vector as the starting point for the next GEPNet iteration.

Remark 1: In the EP detector, the posterior probability distribution of the transmitted symbol estimates is calculated as $p(\mathbf{x}|\mathbf{y}) \propto \prod_{k=1}^K f_{[x_{\text{obs},k}^{(t)}, v_{\text{obs},k}^{(t)}]}(x_k)$, where $f_{[x_{\text{obs},k}^{(t)}, v_{\text{obs},k}^{(t)}]}(\cdot)$ is a Gaussian function parameterized by mean $x_{\text{obs},k}^{(t)}$ and variance $v_{\text{obs},k}^{(t)}$. In our proposed detector, we replace the Gaussian function with a GNN function parameterized not only by $x_{\text{obs},k}^{(t)}, v_{\text{obs},k}^{(t)}$, but also by $\mathbf{y}^T \mathbf{h}_k, \mathbf{h}_k^T \mathbf{h}_k, \mathbf{h}_k^T \mathbf{h}_j, \sigma^2$, where $j = 1, \dots, K, j \neq k$. The GNN function gives more diversity when calculating the posterior probability distribution of the transmitted symbol estimates and enables the proposed detector to capture the MUI information characterized by the pair potential feature $\mathbf{h}_k^T \mathbf{h}_j$.

C. The Estimation Module

The soft symbol estimate and its variance are computed as [16]

$$\hat{x}_k^{(t)} = \sum_{a \in \Omega} a \times p_{\text{GNN}}^{(t)}(x_k = a|\mathbf{y}), \quad (16a)$$

$$v_k^{(t)} = \mathbb{E} \left[\left| x_k - \hat{x}_k^{(t)} \right|^2 \right], \quad (16b)$$

for each $1 \leq k \leq K$. We define a vector $\hat{\mathbf{x}}^{(t)} = [\hat{x}_1^{(t)}, \dots, \hat{x}_K^{(t)}]$ and a $K \times K$ diagonal matrix $\mathbf{V}^{(t)}$ with $v_k^{(t)}$ in the k -th diagonal element, $k = 1, \dots, K$. The work of the GEPNet detector is finished once the maximum number of iterations T has been reached. Hard estimates of the transmitted symbols are then made from $\hat{\mathbf{x}}^{(T)}$ by comparing their Euclidean distance from the symbol set Ω .

In the case of $t \neq T$, the Gaussian posterior function $p^{(t)}(\mathbf{x}|\mathbf{y})$ is re-evaluated by updating $\chi^{(t)}(\mathbf{x})$ as [6]

$$\begin{aligned} \chi^{(t+1)}(\mathbf{x}) &\propto \frac{\mathcal{N}(\mathbf{x} : \hat{\mathbf{x}}^{(t)}, \mathbf{V}^{(t)})}{\mathcal{N}(\mathbf{x} : \mathbf{x}_{\text{obs}}^{(t)}, \mathbf{V}_{\text{obs}}^{(t)})} \\ &= \mathcal{N}(\mathbf{x} : (\boldsymbol{\lambda}^{(t)})^{-1} \boldsymbol{\gamma}^{(t)}, (\boldsymbol{\lambda}^{(t)})^{-1}), \end{aligned} \quad (17)$$

where the parameters

$$\boldsymbol{\lambda}^{(t)} = (\mathbf{V}^{(t)})^{-1} - (\mathbf{V}_{\text{obs}}^{(t)})^{-1}, \quad (18a)$$

$$\boldsymbol{\gamma}^{(t)} = (\mathbf{V}^{(t)})^{-1} \hat{\mathbf{x}}^{(t)} - (\mathbf{V}_{\text{obs}}^{(t)})^{-1} \mathbf{x}_{\text{obs}}^{(t)}. \quad (18b)$$

Note that $\boldsymbol{\lambda}^{(t)}$ in (18a) may yield a negative value, which should not be the case as it is inverse variance term [6]. Therefore, when $\lambda_k^{(t)} < 0$, we assign $\lambda_k^{(t)} = \lambda_k^{(t-1)}$ and

| Detector | Complexity |
|---------------|---|
| AMP [3] | $\mathcal{O}(NKT)$ |
| GNN [10] | $\mathcal{O}((N + S_u N_{h_1} + N_{h_1} N_{h_2} + N_{h_2} S_u)KT)$ |
| MMSE [17] | $\mathcal{O}(K^3 + NK^2)$ |
| RE-MIMO [12] | $\mathcal{O}((N^2 K + NK^2)T)$ |
| OAMP-Net [11] | $\mathcal{O}((N^3 + K^3 + NK^2 + N^2 K)T)$ |
| EP [6] | $\mathcal{O}((K^3 + NK^2 + MK)T)$ |
| GEPNet | $\mathcal{O}((K^3 + NK^2 + MK + (N + S_u N_{h_1} + N_{h_1} N_{h_2} + N_{h_2} S_u)KL)T)$ |
| ML [7] | $\mathcal{O}(M^K)$ |

Table I: The computational complexity comparison

$\gamma_k^{(t)} = \gamma_k^{(t-1)}$. Finally, we smoothen the update of $(\lambda^{(t)}, \gamma^{(t)})$ by using a convex combination with the former values,

$$\lambda^{(t)} = (1 - \eta)\lambda^{(t-1)} + \eta\lambda^{(t-1)}, \quad (19a)$$

$$\gamma^{(t)} = (1 - \eta)\gamma^{(t-1)} + \eta\gamma^{(t-1)}, \quad (19b)$$

where $\eta \in [0, 1]$ is a weighting coefficient. The estimation module sends the parameters $(\gamma^{(t)}, \lambda^{(t)})$ to the observation module, as illustrated in Fig. 2a. The complete pseudo-code is shown in Alg. 1.

Algorithm 1 GEPNet detector

- 1: **Input:** $\mathbf{H}, \mathbf{y}, \sigma^2, E_s, L, T$
 - 2: Initialization: $\gamma^{(0)} = \mathbf{0}, \lambda^{(0)} = \frac{1}{E_s} \mathbf{I}, \eta = 0.7, \mathbf{g}_k^{(0)} = \mathbf{0}$
 - 3: **for** $t = 1, \dots, T$ **do**
 - The Observation Module:**
 - 4: Compute $\Sigma^{(t)}$ and $\mu^{(t)}$ in (5)
 - 5: Compute $v_{\text{obs},k}^{(t)}$ and $x_{\text{obs},k}^{(t)}, k = 1, \dots, K$ in (7)
 - The GNN Module:**
 - 6: Compute (9)
 - 7: **if** $t = 1$ **then**
 - 8: Compute $\mathbf{u}_k^{(0)}, k = 1, \dots, K$, in (11)
 - 9: **end if**
 - 10: **for** $l = 1, \dots, L$ **do**
 - 11: Compute $\mathbf{m}_{j,k}^{(\ell)}$ in (12), $j, k = 1, \dots, K, j \neq k$
 - 12: Compute $\mathbf{g}_k^{(\ell)}$ and $\mathbf{u}_k^{(\ell)}$ in (13), $k = 1, \dots, K$
 - 13: **end for**
 - 14: Compute $p_{\text{GNN}}^{(t)}(x_k | \mathbf{y})$ in (14), $k = 1, \dots, K$
 - The Estimation Module:**
 - 15: Compute $v_k^{(t)}$ and $\hat{x}_k^{(t)}$ in (16), $k = 1, \dots, K$
 - 16: Compute (15)
 - 17: Compute $\lambda^{(t)}$ and $\gamma^{(t)}$ in (18)
 - 18: **if** $\lambda_k^{(t)} < 0$ **then**
 - 19: $\lambda_k^{(t)} = \lambda_k^{(t-1)}$ and $\gamma_k^{(t)} = \gamma_k^{(t-1)}, k = 1, \dots, K$
 - 20: **end if**
 - 21: Smoothen $\lambda^{(t)}$ and $\gamma^{(t)}$ using (19)
 - 22: **end for**
 - 23: **Return:** Hard symbol estimates from $[\hat{x}_1^{(T)}, \dots, \hat{x}_K^{(T)}]$
-

IV. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, we analyse the computational complexity of the proposed GEPNet detector depicted in Alg. 1. Note that we provide complexity for the real-valued system (2). The corresponding complexity for the complex-valued system (1) can be easily obtained by substituting $K = 2N_t$ and $N = 2N_r$. The dominant complexity of the GEPNet detector per iteration is $\mathcal{O}(NK^2 + K^3)$, which comes from (5a). Expressions (5b), (16), (18), and (19) are all related to matrix-vector multiplications and the cost is $\mathcal{O}(K^2 + NK + MK)$. The rest of the operations belong to the GNN computations, whose complexity is $\mathcal{O}((N + S_u N_{h_1} + N_{h_1} N_{h_2} + N_{h_2} S_u)KL)$. As (5)-(19) are performed T times, the total computational complexity of the GEPNet detector is $\mathcal{O}((NK^2 + K^3 + MK + (N + S_u N_{h_1} + N_{h_1} N_{h_2} + N_{h_2} S_u)KL)T)$. Table I shows the computational complexity of the proposed detector in comparison with the state-of-the-art detectors.

V. SIMULATION RESULTS

In this section, we explain the training and testing of the NN-based detectors and compare the performance of our proposed detector with the other MU-MIMO detectors.

A. Implementation Details

We implemented the NN-based detectors OAMPNet, RE-MIMO, GNN, and GEPNet in PyTorch [18]. The hyper-parameters for the existing NN-based detectors were set as in their respective papers. The number of realizations/samples in the training dataset was 80000 for all the NN-based detectors. The samples were obtained by using QAM modulation with varying SNR values. We applied Adam optimizer with learning rate 0.0001 to train the proposed detector, and used the total cross-entropy loss function expressed as

$$Loss = -\frac{1}{Q} \sum_{q=1}^Q \sum_{k=1}^K \sum_{a \in \Omega} \mathbb{I}_{x_k^{(q)}=a} \log \left(p_{\text{GNN}}^{(T)}(x_k = a | \mathbf{y}^{(q)}) \right), \quad (20)$$

where Q is the number of training samples in each batch, $\mathbb{I}_{x_k^{(q)}=a}$ is the indicator function that takes value one if $x_k^{(q)} = a$ and zero otherwise, $\mathbf{x}^{(q)} \in \Omega^K$ is the transmitted vector, $\mathbf{y}^{(q)}$ is the received signal, and $p_{\text{GNN}}^{(T)}(x_k = a | \mathbf{y}^{(q)})$ is the corresponding probability estimate obtained by the GEPNet detector for the q -th training sample and k -th user. Note that $\mathbb{I}_{x_k^{(q)}=a}$ is used as a training label. The GEPNet was trained by using mini-batches of 64 samples and validated by using 20000 samples in every epoch. The total number of epochs was 700. In the testing phase, we first created a testing dataset by randomly generating 1000000 samples for the same system configurations (K, N, M) that were used in the training phase for each SNR point. Finally, we tested all the trained detectors using the testing dataset.

B. SER Comparisons

We investigate the SER performance of our proposed detector by comparing it with those of the MMSE [17], AMP [3]

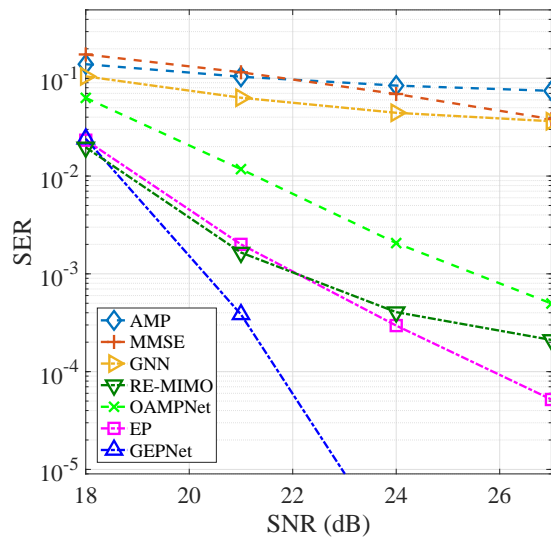
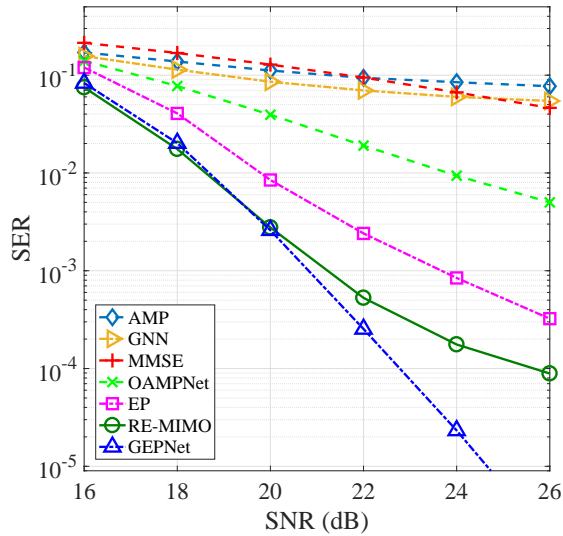


Fig. 3: The SER performance comparison

and EP [6], unfolded NN-based OAMPNet [11], RE-MIMO [12] detectors. We use 16-QAM modulation scheme. In Fig. 3, we employ $N = K = 32$ and $N = K = 64$. The AMP, MMSE, and GNN detectors perform poorly under this system configuration, as well as under other configurations with high ratios of transmit-to-receive antennas. The classical EP detector is able to achieve a better SER performance than the advanced NN-based OAMPNet detector. This is because the EP detector has a significantly better performance compared to the classical AMP based detector. It can be seen from Fig. 3 that the proposed detector achieves at least 4 dB performance gain compared to the EP detector at $\text{SER} = 10^{-4}$. We observe that the curves in Figs. 3a-b behave in a similar way. From these facts, we conclude that the GEPNet detector has a significant performance improvement over the state-of-the-art MU-MIMO detectors.

VI. CONCLUSION

We proposed a high performance MU-MIMO detector, referred to as the GEPNet detector. Simulation results showed that the SER performance of the GEPNet detector was significantly better than that of the other MU-MIMO detectors.

ACKNOWLEDGMENT

This research was supported by the research training program stipend from the University of Sydney. The work of Branka Vucetic was supported by the Australian Research Council Laureate Fellowship grant number FL160100032.

REFERENCES

- [1] D. Borges, P. Montezuma, R. Dinis, and M. Beko, "Massive mimo techniques for 5g and beyond—opportunities and challenges," *Electronics*, vol. 10, no. 14, 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/14/1667>
- [2] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, no. 3, p. 311–335, Mar. 1998.
- [3] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, p. 18914–18919, Nov. 2009.
- [4] S. Rangan, P. Schniter and A. K. Fletcher, "Vector approximate message passing," in *IEEE Int. Symp. on Inform. Theory (ISIT)*, Germany, June 2017, p. 1588–1592.
- [5] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, Jan 2017.
- [6] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Pérez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 2840–2849, Aug. 2014.
- [7] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, p. 463–471, Apr. 1985.
- [8] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel. (2018) Multilevel MIMO detection with deep learning. [Online]. Available: <http://arxiv.org/abs/1812.01571>, preprint.
- [9] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, p. 2554–2564, May 2019.
- [10] A. Scotti, N. N. Moghadam, D. Liu, K. Gafvert, and J. Huang. (2020) Graph neural networks for massive MIMO detection. [Online]. Available: <https://arxiv.org/abs/2007.05703>, preprint.
- [11] H. He, C. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, USA, Nov 2018, pp. 584–588.
- [12] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and permutation equivariant neural MIMO detection," *IEEE Trans. Signal Process.*, vol. 69, p. 459–473, Jan. 2021.
- [13] G. D. Forney, "Codes on graphs: Normal realizations," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.
- [14] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.
- [15] K. Yoon, R. Liao, Y. Xiong, L. Zhang, E. Fetaya, R. Urtasun, R. Zemel, and X. Pitkow. (2019) Inference in Probabilistic Graphical Models by Graph Neural Networks. [Online]. Available: <https://arxiv.org/abs/1803.07710>, preprint.
- [16] A. Leon-Garcia, *Probability, statistics, and random processes for electrical engineering*, 3rd Edition. Pearson/Prentice Hall, 2008.
- [17] G. Caire, R. Muller, and T. Tanaka, "Iterative multiuser joint decoding: Optimal power allocation and low-complexity implementation," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, p. 1950–1973, Sep. 2004.
- [18] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, p. 8024–8035, 2019.