A Double-CNN BP Decoder on Fast Fading Channels Using Correlation Information

Jun Li*, Xiwei Zhao*, Jihao Fan*, Feng Shu*, Shi Jin†, and Yuwen Qian*
*Nanjing University of Science and Technology, Nanjing, China
†Southeast University, Nanjing, China
E-mail:{jun.li, zhaoxiwei, shufeng, admon}@njust.edu.cn, jihao.fan@outlook.com, jinshi@seu.edu.cn

***Abstract*—This paper develops a novel double convolutional neural network (CNN) based belief propagation (BP) decoder to improve the error correcting performance on fast fading channels with correlated channel gain and correlated noise. The proposed double-CNN BP decoder consists of two CNNs for predicting channel gain and noise samples, respectively, concatenated with a BP decoder. The input of the BP decoder is the log-likelihood-ratio (LLR) values obtained according to the predicted channel gain along with denoised signals based on predicted noises. We note that the residual noises no longer obey a Gaussian distribution when denoising the signals. Thus, we further derive a new method to obtain the LLR values accurately. Simulation results show that the proposed double-CNN BP decoder achieves a maximum of 7dB gain compared to the conventional BP decoder and the proposed algorithm compensates the performance loss of BP decoder caused by correlation information.**

***Index Terms*—Channel decoder, LDPC, CNN, correlated channel gain, correlated noise.**

I. Introduction

Low density parity check (LDPC) [1] codes, based on a belief propagation (BP) decoder [2], are widely adopted to various channel environments [3], [4] due to their excellent decoding performance. However, there are still some problems in BP decoder. One of them is that in fading channels, the performance of BP decoder is largely depend on the accuracy of channel gain prediction. Another problem of BP decoder is that when noise or channel gain is correlated, the performance of BP decoder declines.

In recent years, artificial intelligence (AI) technology has become more and more important to drive the development of science and technology because of the improvement of computing power and related algorithms. As one of the AI technologies, deep learning shows outstanding performance in a wide range of fileds. These exciting results inspire us to adopt the AI technology to solve traditional communication problems. The method such as recurrent neural network (RNN) [5] assisted channel estimation has been proposed in [6]. This method requires historical information to predict future information and cannot predict channel information in real time. The work in [7] presented a method of constructing channel decoders using neural networks,which does not need to construct a special code structure, the neural network is difficult to be trained for large code length.

Interestingly, the work in [8] constructed an concatenated decoding structure based on the iterative processing between a BP decoder and a convolutional neural network (CNN) [9] in the context of correlated noises. Particularly, the work [8] predicted noise samples by taking the advantage of the correlation among the noises. Thus, it achieved a significant performance gain relative to a conventional BP decoder. The ideas of signal denoising and the method of using noise correlation information are worth learning.

The causes of channel correlation [10] are quite different from the noise correlation [11]. The filter and oversampling in the receiver are the causes of noise correlation. By contrast, channels correlation stems from multipath propagation, antenna separation, and collocated nodes in cognitive radio (CR) networks [12]. On the one hand, the correlation of noise or channel gain restricts the performance of BP decoder, on the other hand, it is a breakthrough for us to solve the problem of performance attenuation of BP decoder.

We utilize channel and noise correlations for predicting fast channel gain and noise samples, which is the most important step for compensating the performance loss of BP decoder caused by correlation information. When dealing with the prediction task with long code length, the full connection of deep neural network will be difficult to train because of too many parameters. The ability of CNN to extract local information and reduce parameters due to Local connectivity and shared weights determines that they are no longer limited by code length, but can still handle the prediction tasks of noise and channel gain better. In addition, the proposed method also avoids the problem that the data set is too large and difficult to train due to the long code length, and does not need to obtain historical information.

In this paper, we aim at constructing a CNN-based BP decoder by exploiting the correlations in order to improve the decoding performance. Since codewords are a series of sequences, we adopt 1D-CNN at the receiver. Specifically, we propose a double-CNN BP decoder structure with two 1D-CNNs. One of the CNNs is utilized to predict the channel gain, and the other one is for predicting noise samples. Before performing the BP decoding, the received signal is first denoised based on the predicted noises. Then combined with the predicted channel gain, we propose a novel approach for calculating the LLR for the BP decoder. Simulations show that the proposed decoder achieves 1-7dB performance gain compared with the traditional BP decoder in the correlated environment.

The structure of this paper is as follows. Section II introduces the system model, including the part of communication process and the part of channel environment. Section III shows the proposed double-CNN BP decoder in detail. The simulation results and analyses are presented in Section IV. Finally, we summarize this paper in Section V.

II. System Model

In this paper, we take the digital wireless communication system on fast fading channels as the system model. Fig. 1 shows the communication system from signal encoding to receiving.

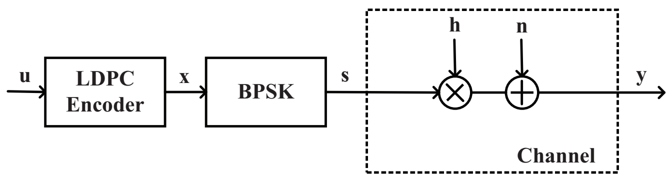


Fig. 1. Communication system. The information bit $\mathbf{u}$ is encoded by the LDPC encoder into a codeword $\mathbf{x}$, which is modulated into $\mathbf{s}$ via BPSK. The channel gain $\mathbf{h}$ is generated by fast fading channel transmission and the noise $\mathbf{n}$ is generated at the receiver. Finally The communication system obtains a signal $\mathbf{y}$.

A. Communication Process

Firstly, we briefly introduce the encoding of the signals. At the channel encoder, the input message $\mathbf{u} = [u_0, u_1, \cdots, u_{k-1}]$ of k information bits is encoded into an n bits codeword $\mathbf{x} = [x_0, x_1, \cdots, x_{n-1}]$ by some certain encoding rules. The encoding formula is given as follows:

$$\mathbf{x} = \mathbf{u}\mathbf{G}, \tag{1}$$

where $\mathbf{G}$ is a $k \times n$ generator matrix, the codeword $\mathbf{x}$ is expressed as the matrix product of the message vector $\mathbf{u}$ and $\mathbf{G}$ over binary field GF(2). Then, the codeword $\mathbf{x}$ is modulated to a symbol $\mathbf{s}$ by BPSK, which is given as

$$\mathbf{s} = \mathbf{1} - \mathbf{2}\mathbf{x}. \tag{2}$$

The symbol vector $\mathbf{s} = [s_0, s_1, \cdots, s_{n-1}]$ is transmitted by the transmitter to the receiver via the wireless channel. Finally, the signal vector $\mathbf{y} = [y_0, y_1, \cdots, y_{n-1}]$ is received at the receiver. The signal vector $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \mathbf{s} \odot \mathbf{h} + \mathbf{n}, \tag{3}$$

where the correlated channel gain $\mathbf{h} = [h_0, h_1, \cdots, h_{n-1}]$ is generated in the wireless communication process. $\mathbf{s}$ and $\mathbf{h}$ carry out hadamard product operation, and the correlated noise $\mathbf{n} = [n_0, n_1, \cdots, n_{n-1}]$ is generated during the receiving process.

B. Channel Environment and Correlation

In this paper, we model the channel as a fast fading channel, which follows a complex Gaussian distribution. The amplitude of the channel gain obeys the Rayleigh distribution. The channel gain vector $\mathbf{g} = [g_0, g_1, \cdots, g_{n-1}]$ is given as

$$\mathbf{g} = \mathbf{a} + \mathrm{j}\mathbf{b}, \tag{4}$$

where $\mathbf{a}$ and $\mathbf{b}$ are independent and identically distributed Gaussian random vectors with zero mean and unit variance. Therefore, the amplitude of the channel gain $|\mathbf{g}|$ which follows the Rayleigh distribution is given as

$$|\mathbf{g}| = K\sqrt{\mathbf{a}^2 + \mathbf{b}^2}, \tag{5}$$

where K is a scaling factor. For ensuring that the average power of the channel gain is equal to 1, K is equal to $\sqrt{1/2}$ by calculation.

As in [13], we model the channel correlation as an $n \times n$ single coefficient exponential correlation matrix Θ_f:

$$\Theta_{i,j} = \begin{cases} \xi_f^{(j-i)}, & i \le j, \\ (\xi_f^{(i-j)})^*, & i > j, \end{cases} \quad (i, j = 1, 2, \cdots, n) \tag{6}$$

where each element of the correlation matrix Θ_f is denoted by $\Theta_{i,j}$, the noise correlation coefficient is represented by ξ_f. The channel correlation is described in [14]. The correlated channel gain $\mathbf{h}$ can be defined as follows:

$$\mathbf{h} = \mathbf{g}\Theta_f^{\frac{1}{2}}, \tag{7}$$

where the correlation matrix Θ_f needs to perform the matrix square root calculation.

The amplitude of the correlated channel gain $|\mathbf{h}|$ is expressed as

$$|\mathbf{h}| = K\sqrt{(\mathbf{a}\Theta_f^{\frac{1}{2}})^2 + (\mathbf{b}\Theta_f^{\frac{1}{2}})^2}. \tag{8}$$

In addition, the correlated noise $\mathbf{n}$ is given by

$$\mathbf{n} = \mathbf{z}\Theta_n^{\frac{1}{2}}, \tag{9}$$

where the form of the noise correlation matrix Θ_n is consistent with formula (6). The noise correlation coefficient is expressed as ξ_n. We keep the noise vector $\mathbf{z}$ following the normal distribution with zero mean and variance σ^2. Due to the fact that the noise correlation matrix Θ_n does not change the distribution of noise and its power, so that the distribution of correlated noise $\mathbf{n}$ is the same as $\mathbf{z}$.

III. Proposed Double-CNN BP Decoder

A. Decoding Process

When the signal $\mathbf{y}$ without noise correlation is decoded on correlated fading channels, we use the CNN-1 to obtain the predicted channel gain for recalculating the LLR information as the input of the traditional BP algorithm. We define the structure as the CNN-BP decoder in Fig. 2, where $\hat{\mathbf{h}}$ is the predicted channel gain value, $\mathbf{LLR}'$ is the recalculated input of the BP decoder and $\mathbf{s}'$ is the decoding result obtained by the CNN-BP decoder in Fig. 2.

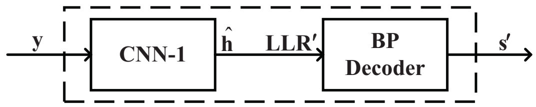


Fig. 2. Proposed CNN-BP Decoder. CNN-1 predicts the channel gain $\hat{\mathbf{h}}$ which used to calculate the $\mathbf{LLR}'$. Finally the BP decoder obtains the decoding result $\mathbf{s}'$.

The accuracy of value acquisition directly affect the performance of BP decoding, so we briefly describe the calculation process of $\mathbf{LLR}$. The original definition of $\mathbf{LLR}$ given by

$$\mathrm{LLR}_i = \ln\frac{\Pr\left(x_i = 1\,|y_i\right)}{\Pr\left(x_i = 0\,|y_i\right)}, (i = 1, 2, \cdots, n) \tag{10}$$

where x_i is the ith transmitting bit of the codeword $\mathbf{x}$, y_i is the ith received bit of the received signal vector $\mathbf{y}$ and LLR_i is the ith LLR value of vector $\mathbf{LLR}$.

If the noise $\mathbf{n}$ follows the normal distribution with zero mean and σ^2 variance, LLR is deduced as follows:

$$\mathrm{LLR}_i = \ln\left(\frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\left(\frac{y_i-1}{\sigma}\right)^2\right]}{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{1}{2}\left(\frac{y_i+1}{\sigma}\right)^2\right]}\right) = \frac{2}{\sigma^2}y_i. \tag{11}$$

In this paper, the formula for calculating the LLR value under the fading channel constructed by the predicted channel gain $\hat{\mathbf{h}}$ is as follows:

$$\mathrm{LLR}_i{}' = \ln\frac{\Pr(x_i = 1\,\big|y_i, \hat{h}_i)}{\Pr(x_i = 0\,\big|y_i, \hat{h}_i)} = \frac{2}{\sigma^2}y_i\hat{h}_i, \tag{12}$$

where $\hat{h}_i$ is the ith normalized channel gain of $\hat{\mathbf{h}}$, and $\mathbb{E}\left[\hat{\mathbf{h}}^2\right] = 1$. Using the redefined $\mathbf{LLR}'$ as the initial information, the BP decoder outputs the decoded results $\mathbf{s}'$.

When the task is extended to channel decoding with noise correlation, we develop a Double-CNN BP decoder in the paper shown in Fig. 3. Two CNNs are employed to perform different tasks respectively in the Double-CNN BP decoder, where $\hat{\mathbf{h}}$ is the predicted channel gain value, $\hat{\mathbf{n}}$ is the predicted noise, $\mathbf{y}'$ is the denoised signal, $\mathbf{LLR}''$ is the recalculated input of the BP decoder and s'' is the decoding result obtained by the Double-CNN BP decoder in Fig. 3.

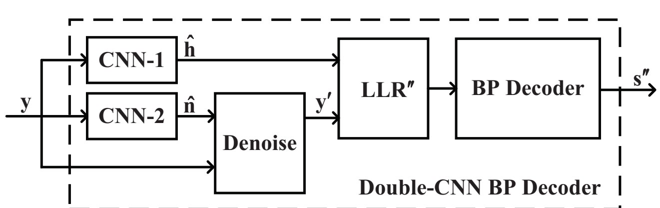


Fig. 3. Proposed Double-CNN BP decoder. The Double-CNN is divided into two parts, CNN-1 is used to predict the channel gain $\hat{\mathbf{h}}$, and CNN-2 is used to predict the noise $\hat{\mathbf{n}}$. Denoised Signal $\mathbf{y}'$ is used to calculate the $\mathbf{LLR}''$ value along with the predicted channel gain $\hat{\mathbf{h}}$.

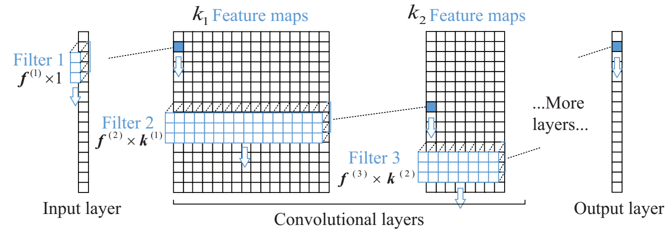


Fig. 4. CNN framework. The CNN used in this paper is composed of one-dimensional convolutional layers entirely. The input layer uses a convolution operation to form two-dimensional data on the second layer.

The first step, like the CNN-BP decoder, CNN-1 is utilized to predict the channel gain $\hat{\mathbf{h}}$. In addition, CNN-2 is adopted to predict the noise $\hat{\mathbf{n}}$. CNN-1 and CNN-2 all use the received signal $\mathbf{y}$ as the input. The second step, by denoising process using the predicted noise $\hat{\mathbf{n}}$, we get the denoised signal $\mathbf{y}'$:

$$\mathbf{y}' = \mathbf{y} - \hat{\mathbf{n}} = \mathbf{s}\odot\mathbf{h} + \mathbf{n} - \hat{\mathbf{n}} = \mathbf{s}\odot\mathbf{h} + \mathbf{n}', \tag{13}$$

where we define $\mathbf{n}'$ as the residual noise. Theoretically, the bit error rate (BER) will be significantly reduced when decoding the denoised signal $\mathbf{y}'$ by BP decoding. But we note that the process given in formula (12) will no longer be used to compute the initial information of BP decoding. Because $\mathbf{n}'$ is no longer follow normal distribution, which means that we need to recalculate the probability distribution of $\mathbf{n}'$. Through histogram statistics, we obtain the empirical probability distribution function (EPDF) of $\mathbf{n}'$ defined as $F(\cdot)$. The next step, we propose $\mathbf{LLR}''$ using $F(\cdot)$ rewritten as

$$\mathrm{LLR}_i{}'' = \ln\frac{\Pr(y_i'\,|s_i\, = 1, \hat{h}_i)}{\Pr(y_i'\,|s_i\, = -1, \hat{h}_i)} = \ln\frac{F\left(y_i' - \hat{h}_i\right)}{F\left(y_i' + \hat{h}_i\right)}, \tag{14}$$

where y_i' is the ith denoised symbol of $\mathbf{y}'$. The last step, we use the redefined $\mathbf{LLR}''$ as the initial information of our decoder to obtain the decoding result $\mathbf{s}''$.

B. Convolutional Neural Network

In order to make CNN work effectively, the following three tasks need to be carried out in turn.

(1) Get the training data set.

(2) Construct suitable neural networks structure.

(3) Set up hyper-parameters and train the neural networks.

Task (1) is to acquire the training data set. For the CNN-1, the data of the received signal $\mathbf{y}$ is obtained through the communication simulation program. The method of generating the data of the correlated channel gain $\mathbf{h}$ has been introduced in the part B of Section II. The correlated channel gain $\mathbf{h}$ is adopted as the data label paired with $\mathbf{y}$, which is the output of the CNN-1 when the CNN-1 is trained with the input data $\mathbf{y}$. For the CNN-2, it is used to predict the correlated noise. The data acquisition of CNN-2 is similar to CNN-1. We need acquire the noise $\mathbf{n}$ as the label corresponding to $\mathbf{y}$. All the data sets are generated at different signal-noise ratios (SNRs): $\{0, 2, 4, 5, 6, 7, 8\}$ dB, and 50,000 frames of data are generated at each SNR. The proportion of each SNR is the same, which ensure the generalization ability of the neural network for each noise power.

Task (2) constructs the CNN structure according to the data characteristics. It is known that the 1D-CNN has powerful sequence data feature extraction capabilities [15]. Therefore, we build a network structure based on the 1D-CNN, which is shown in Fig. 4. For the CNN-1, the input of the neural network is the vector $\mathbf{y}$, which is fed to the input layer of the CNN-1, and the convolution kernel is convoluted along the vector $\mathbf{y}$. This process can be expressed as follows:

$$\mathbf{k}_{p,q} = \lambda_p(\mathbf{f}_{p,q} \otimes \mathbf{y} + \mathbf{b}_{p,q}), \tag{15}$$

where p represents the pth layer of the neural network, and q represents the qth feature map of the pth layer. When $p = 1$, the $\mathbf{k}_{1,q}$ represents the qth feature map of the first layer, the $\mathbf{f}_{1,q}$ represents the qth convolution kernel of the first layer, the $\mathbf{b}_{1,q}$ is the corresponding bias vector, and λ is the activation function. The $f^{(p)}$ stands for the length of convolution kernel of the pth layer. For the first layer, convolution kernel is an $f^{(1)} \times 1$ vector. Vector $\mathbf{y}$ has a convolution calculation with the convolution kernel $\mathbf{f}_{1,q}$, then it is added to the corresponding bias $\mathbf{b}_{1,q}$. Finally it is outputted as a feature maps $\mathbf{k}_{1,q}$ by the activation function λ_p. The $k^{(p)}$ presents the number of feature maps on the pth layer. The $k^{(1)}$ feature maps vectors form a two-dimensional matrix, which is the second convolutional layer. When $p > 1$, the convolution operation differs from the first layer in that the shape of the convolution kernel becomes $f^{(p)} \times k^{(p-1)}$. When we focus on the output layer, assuming that the CNN has L layers, the final output can be given as

$$\hat{\mathbf{h}} = \lambda_L(\mathbf{f}_{L-1,q} \otimes \mathbf{k}_{L-1} + \mathbf{b}_{L-1,q}), \tag{16}$$

where the activation function of the output layer should use a linear activation function because the output of the CNN is positive or negative. To ensure that the output is a vector, $k^{(L-1)}$ is equal to 1.

Task (3) gives the CNN appropriate parameters and hyper-parameters. The structure of CNN is determined by the number of convolutional layers, the length of the convolution kernel and the number of feature maps. In this paper we refer to the structure of the 1D-CNN in [8]. The entire CNN has only one-dimensional convolutional layer. In order to keep the same size of each layer, the same padding operation is adopted, in which the edge of the next layer is zero-padded, after the convolution operation is performed on each layer. The CNN structure parameters are shown in Table I.

TABLE I
CNN PARAMETERS

layers i	layer 1	layer 2	layer 3	layer 4	layer 5
kind of layer	Input	1D Conv	1D Conv	1D Conv	Output
kernel $f^{(i)}$	9	3	3	15	/
feature map $k^{(i)}$	64	32	16	1	/
active function λ_i	ReLU	ReLU	ReLU	Linear	/

Rectified Linear Unit (ReLU), also known as modified linear unit, is an activation function commonly used in artificial neural networks. ReLU can obviously reduce the problem of vanishing gradient, which can be expressed as $f(x) = \max(0, x)$.

Through a large number of experiments, we determines the appropriate hyper-parameters for neural network training. The specific hyper-parameters are given in table II.

TABLE II
CNN HYPER-PARAMETERS

learning rate	0.001
epoch	1000
mini-batchsize	700
initializer	Xavier
optimizer	Adam
loss function	MSE

Adam is an optimization algorithm that can replace the traditional stochastic gradient descent process. It can iteratively update the neural network weights based on the training data. The training process of the neural network is realized by one forward propagation and one backward propagation. The gradient descent method in back propagation is replaced by Adam. Mean squared error (MSE) is a commonly used loss function, which is expressed as

$$\mathrm{Loss}(h_i, f(y)) = \frac{\sum_{i=1}^{N} (h_i - f(y_i))^2}{N}, \tag{17}$$

where N represents the total number of data sets, the neural network is treated as a function, and $f(\cdot)$ is the output of the neural network.

The structure, parameters and hyper-parameters of the CNN-2 are consistent with the settings of the CNN-1.

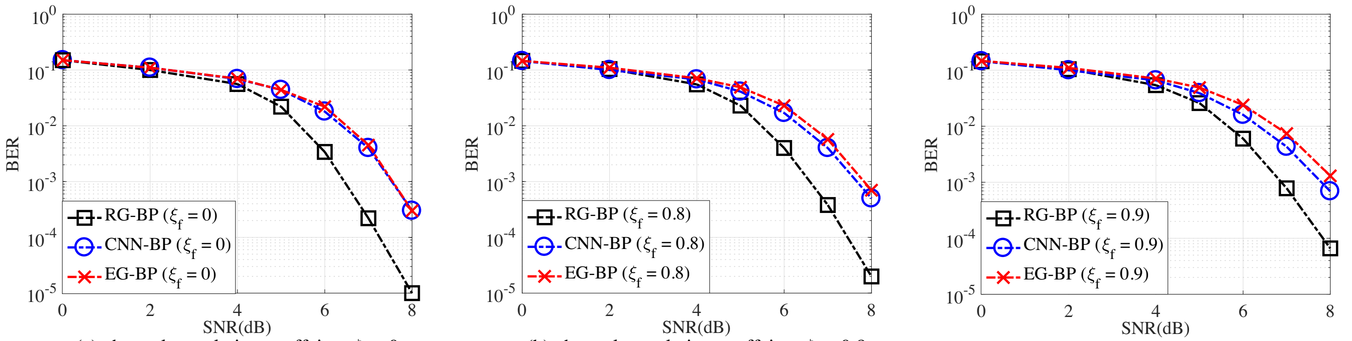
IV. SIMULATIONS

The simulation uses the WiMax standard LDPC [16] code with the block length 576 and the code rate 0.75. We perform channel decoding simulation under SNRs: $\{0, 2, 4, 5, 6, 7, 8\}$ dB. All parameters and hyper-parameters used in the CNN are described in Section III. The CNN training and inference tasks base on the Python language using a NVIDIA GTX1080 graphics card. The neural network structure is based on Keras with backend of Tensorflow. Keras highly encapsulates complex algorithms, providing a flexible and an efficient engineering implementation environment.

A. Channel Correlation without Noise Correlation

Using the single coefficient correlation model defined in Section II, the channel correlation coefficient ξ_f is set to be 0, 0.8, and 0.9 respectively in our simulations. The maximum number of iterations for the BP algorithm is set to be 25.

We compare the decoding results using predicted channel gain $\hat{\mathbf{h}}$ (denoted by "CNN-BP") with that using real channel gain $\mathbf{h}$ (denoted by "RG-BP") and expectation channel



(a) channel correlation coeffcient $\xi_f = 0$ (b) channel correlation coeffcient $\xi_f = 0.8$ (c) channel correlation coeffcient $\xi_f = 0.9$

Fig. 5. The comparison of the decoding performance using predicted channel gain with that using real channel gain and expectation channel gain under different channel correlations coefficient ξ_f.

gain $\mathbb{E}[\mathbf{h}]$ (denoted by "EG-BP") ($\mathbb{E}[\mathbf{h}] = 0.8862$, when $\mathbb{E}[\mathbf{h^2}] = 1$), which are shown in Fig. 5 (a), (b) and (c). We regard RG-BP algorithm as the upper bound of decoding performance and EG-BP algorithm as the lower bound of decoding performance, which are regarded as benchmarks for the comparison with CNN-BP algorithm.

The decoding result of CNN-BP algorithm compared with that of EG-BP algorithm, shows that it has a performance gain. When the channel correlation coefficient ξ_f is 0.9, the gain is 0.2 dB at BER = 10^{-2}. When the ξ_f decreases to 0.8, this gain decreases to 0.1 dB. When the channel is not correlated, the curve "CNN-BP" and curve "EG-BP" almost coincide. In addition, simulation results show that there is a performance gap between curve "CNN-BP" and curve "RG-BP", when the ξ_f is 0, 0.8, and 0.9, respectively.

Compared with the decoding result s of EG-BP algorithm, the simulations show that the stronger the correlation of the channel, the better gain the CNN-BP algorithm gets. Although stronger correlation can make CNN have stronger channel gain prediction ability, it also makes the performance of BP decoder decline more significantly. The decoding performance using the CNN-BP algorithm can at least achieve the decoding performance obtained by using the gain expectation value, which shows that CNN-BP algorithm is an effective method to obtain channel gain.

B. Coexistence of Channel Correlation and Noise Correlation

In this part, we fix the channel correlation coefficient ξ_f to be 0.9 and change the noise correlation coefficient ξ_n, which is shown in Fig. 6. The maximum number of iterations for the BP algorithm is set to be 25. We show the decoding results of non-denoisied signals (denoted by "EG-BP", "CNN-BP") and that of denoisied signals (denoted by "DCNN-BP", "RGD-BP") under different noise correlations. The curve "EG-BP" represents the traditional way to obtain a fixed channel gain value by statistical methods and then directly perform BP decoding. The curve "DCNN-BP" represents the decoding result of denoised signals using the predicted channel gain $\hat{\mathbf{h}}$, which is the double-CNN BP decoder we have proposed. The curve "RGD-BP" represents the decoding result of denoised signal using the real channel gain $\mathbf{h}$, which is regarded as the upper bound of decoding performance of denoised signals.

The DCNN-BP algorithm in Fig. 6 (a), (b) and (c) has significant gains compared with the EG-BP algorithm, in which the performance gains are 1 dB when ξ_n is 0.5, 4 dB when ξ_n is 0.8, and 7 dB when ξ_n is 0.9. These obvious performance improvements prove that channel gain prediction and denoising method is beneficial to decoding performance improvement compared with traditional BP algorithm under correlated environments.

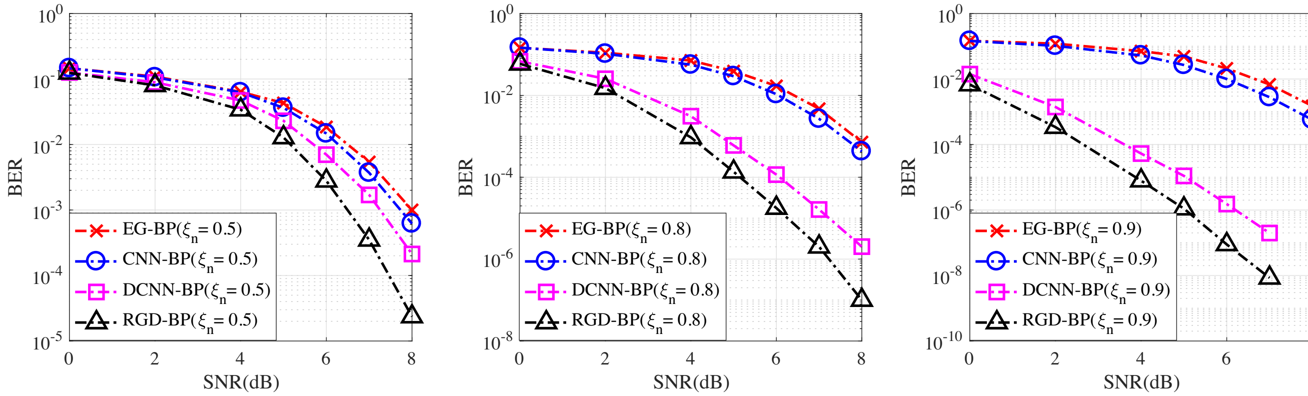
Fig. 7 shows the good adaptability of the DCNN-BP algorithm. The performance of the DCNN-BP algorithm in the correlated environment and the uncorrelated environment is not worse than the decoding performance of the traditional EG-BP algorithm, and DCNN-BP shows a huge performance advantage as the environmental correlation improves.

C. The Influence of Correlation on BP Decoder

Through comparing the three curves of "DCNN-BP" in Fig. 6, we show that the stronger the noise correlation, the worse the decoding performance of BP is. This phenomenon is also shown in Fig. 5. The LDPC decoding adopts iterative decoding and its algorithm derivation is based on the statistical independence of the information transmitted between different nodes. When there are short cycles in the Tanner graph corresponding to the check matrix of LDPC codes, the information sent by a node and is transmitted back to itself through a cycle length transmission, which results in the superposition of its own information and the dependence of information. Correlated channel gains and noises destroy the independence of each bit in the received signals, which decline the final decoding performance.

V. Conclusion and Future Work

In this paper, we propose a double-CNN BP decoding system through cascading the CNN and BP decoder. The double-CNN BP decoder uses CNN to predict the correlated channel gain and correlated noise, and utilizes these two predicted values to redefine the LLR calculation method. The double-CNN BP decoder compensates for the negative impact



(a) channel correlation coeffcient ξ_f= 0.9, noise correlation coeffcient ξ_n= 0.5

(b) channel correlation coeffcient ξ_f= 0.9, noise correlation coeffcient ξ_n= 0.8

(c) channel correlation coeffcient ξ_f= 0.9, noise correlation coeffcient ξ_n= 0.9

Fig. 6. The decoding performance of denoised signals and non-denoised signals under different noise correlations coefficients ξ_n.

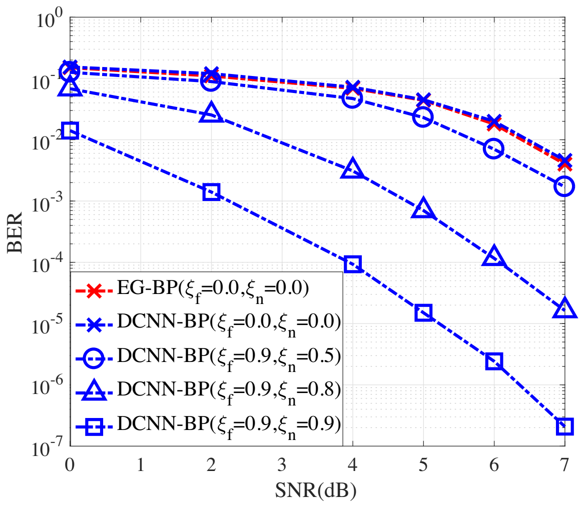


Fig. 7. Comparisons between decoding in correlation environment and decoding in non-correlation environment. The curve "EG-BP" represents the traditional BP algorithm in uncorrelated environment. The curve "DCNN-BP" represents the decoding result of channel gain prediction and noise elimination using CNN.

of correlation information on BP decoder, and achieves and surpasses the decoding performance of traditional BP decoder in correlated environment with the increasing correlation. In future work, we will consider extending the relevance modeling approach used in this paper to the privacy preservation.

REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory.*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory.*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[3] J. Li, W. Chen, Z. Lin, and B. Vucetic, "Design of physical layer network coded ldpc code for a multiple-access relaying system," *IEEE Communications Letters*, vol. 17, no. 4, pp. 749–752, 2013.

[4] J. Li, J. Yuan, R. Malaney, M. H. Azmi, and M. Xiao, "Network coded ldpc code design for a multi-source relaying system," *IEEE Transactions on Wireless Communications*, vol. 10, no. 5, pp. 1538–1551, 2011.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994.

[6] W. Liu, L.-L. Yang, and L. Hanzo, "Recurrent neural network based narrowband channel prediction," in *Proc. IEEE 63rd Veh. Technol. Conf.*, vol. 5, Melbourne, Australia, May 2006, pp. 2173–2177.

[7] T. Gruber, S. Cammerer, J. Hoydis, and S. t. Brink, "On deep learning-based channel decoding," in *Proc. IEEE 51st Annu. Conf. Inf. Sci. Syst.*, Baltimore, MD, USA, Mar. 2017, pp. 1–6.

[8] F. Liang, C. Shen, and F. Wu, "An iterative BP-CNN architecture for channel decoding," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 144–159, Feb. 2018.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[10] D.-S. Shiu, G. J. Foschini, M. J. Gans, and J. M. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 502–513, Mar. 2000.

[11] L. D. Bert, P. Caldera, D. Schwingshackl, and A. M. Tonello, "On noise modeling for power line communications," in *in Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Udine, Italy, Apr. 2011, pp. 283–288.

[12] S. K. Sharma, S. Chatzinotas, and B. Ottersten, "SNR estimation for multi-dimensional cognitive receiver under correlated channel/noise," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6392–6405, Dec. 2013.

[13] S. Chatzinotas, M. A. Imran, and R. Hoshyar, "On the multicell processing capacity of the cellular MIMO uplink channel in correlated rayleigh fading environment," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3704–3715, Jul. 2009.

[14] X. Mestre, J. R. Fonollosa, and A. Pages-Zamora, "Capacity of MIMO channels: asymptotic evaluation under correlated fading," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 829–838, Jun. 2003.

[15] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[16] M. Helmling and S. Scholl, "Database of channel codes and ML simulation results," Wimax 3/4A, University of Kaiserslautern, Kaiserslautern, Germany, 2016. [Online]. Available: www.uni-kl.de/channel-codes