

# Joint Neural Network Equalizer and Decoder

Weihong Xu<sup>1,2,3</sup>, Zhiwei Zhong<sup>1,2,3</sup>, Yair Be'ery<sup>4</sup>, Xiaohu You<sup>2</sup>, and Chuan Zhang<sup>1,2,3,\*</sup>

<sup>1</sup>Lab of Efficient Architecture for Digital Communication and Signal Processing (LEADS)

<sup>2</sup>National Mobile Communications Research Laboratory

<sup>3</sup>Quantum Information Center, Southeast University, China

<sup>4</sup>School of Electrical Engineering, Tel-Aviv University, Israel

Email: <sup>2</sup>{wh.xu, zwzhong, xhyu, chzhang}@seu.edu.cn, <sup>4</sup>ybeery@eng.tau.ac.il

**Abstract**—Recently, deep learning methods have shown significant improvements in communication systems. In this paper, we study the equalization problem over the nonlinear channel using neural networks. The joint equalizer and decoder based on neural networks are proposed to realize blind equalization and decoding process without the knowledge of channel state information (CSI). Different from previous methods, we use two neural networks instead of one. First, convolutional neural network (CNN) is used to adaptively recover the transmitted signal from channel impairment and nonlinear distortions. Then the deep neural network decoder (NND) decodes the detected signal from CNN equalizer. Under various channel conditions, the experiment results demonstrate that the proposed CNN equalizer achieves better performance than other solutions based on machine learning methods. The proposed model reduces about 2/3 of the parameters compared to state-of-the-art counterparts. Besides, our model can be easily applied to long sequence with  $\mathcal{O}(n)$  complexity.

**Index Terms**—Channel equalization, channel decoding, neural networks, deep learning.

## I. INTRODUCTION

Recently, deep learning-aided communication systems have demonstrated amazing improvements regarding performance and adaptability. An end-to-end communications systems based on deep learning is introduced in [1]. For channel decoding, the authors in [2] propose a systematic framework to construct deep neural network decoder of BCH code, which achieves encouraging performance. The neural network decoder (NND) of polar codes [3] in [4, 5] shows significant improvement compared with conventional methods.

Except for noise, communication channel usually introduces inter-symbol interference (ISI) to the transmitted signal. The amplifiers and mixers will additionally cause nonlinear distortion. Channel equalization is exploited to deal with these dispersive effects and delivers recovered signal to the channel decoder. Then the channel decoder performs decoding to correct errors during transmission.

Various machine learning approaches have been utilized to address the nonlinear equalization without accurate knowledge of channel state information (CSI). These methods include deep neural network (DNN) [6], convolutional neural network (CNN) [7], Gaussian processes for classification (GPC) [8] and support vector machine (SVM) [9], enabling the receiver side adaptively realize equalization. However, DNN equalizer in [6] is only practical for short codes and requires large amounts of parameters. The equalizers in [8, 10] both use multiple

training sequences to estimate the channel filter coefficients of ISI. These two methods also require other a priori knowledge such as the distribution of channel filter coefficients and the Gaussian noise variance. Such assumptions may not hold in practice. Thus, it is hard to obtain the accurate results of CSI analytically without the a priori information. Indicated in [8], the performance of SVM equalizer after decoder is poor.

In this work, we propose neural network based joint channel equalizer and decoder. The CNN equalizer is first utilized to compensate the distortion of signal. Then the other polar NND is cascaded after CNN to decode the recovered sequence. The networks are trained with an offline and end-to-end manner, which is capable of carrying out equalization and decoding without the side information of CSI. The experiments show that promising bit error rate (BER) improvement is achieved compared with the other machine learning methods.

The rest of this paper is organized as follows. Brief preliminaries about system setup and ML equalizer are reviewed in Section II. In Section III, the details of proposed joint neural network equalizer and decoder as well as corresponding training methods are presented. The experiment results are shown in Section IV and analysis is given in Section V. Section VI finally concludes the paper.

## II. PRELIMINARIES

### A. System Setup

Throughout this paper, the discrete-time dispersive digital-communication system with a nonlinear channel is considered. In this case, the ISI, nonlinearities, and AWGN jointly result in the channel distortion. The system is in Fig. 1.

1) *Inter-symbol Interference*: The channel with ISI is modeled as a finite impulse response (FIR) filter. The corresponding filter coefficients with length  $L$  are given by  $\mathbf{h} = [h_1, h_2, \dots, h_L]^T$ . The signal with ISI is equivalent to the convolution of channel input with the FIR filter as follows:

$$\mathbf{v} = \mathbf{s} \otimes \mathbf{h}, \quad (1)$$

where  $\mathbf{s}$  denotes the channel input sequence after modulation and  $\otimes$  represents linear convolution operation.

2) *Nonlinear Distortion*: The nonlinearities in the communication system are mainly caused by amplifiers and mixers, which can be modeled by a nonlinear function  $g[\cdot]$ . After being

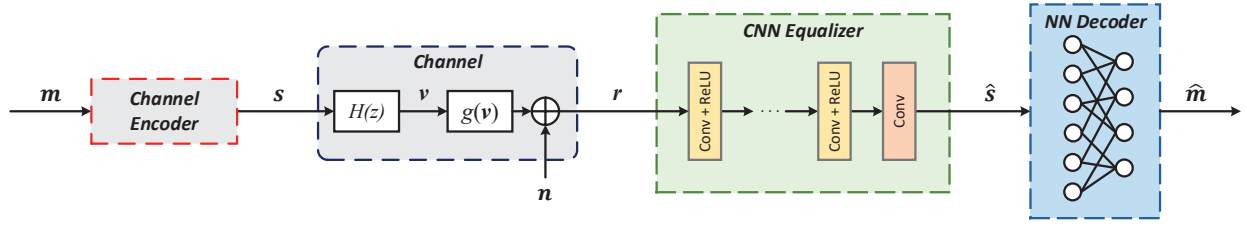


Fig. 1. System architecture.

transmitted through the channel, the output  $\mathbf{r}$  involved with ISI, nonlinearities, and AWGN is expressed as:

$$r_i = g[v_i] + n_i. \quad (2)$$

### B. Maximum Likelihood Equalizer

For the ML channel equalizer, a training sequence  $\mathbf{s}^\circ = \{s_1^\circ, s_2^\circ, \dots, s_n^\circ\}$  with  $n$  known bits is sent to the receiver side to estimate the channel condition. The ML criterion for the channel coefficients is considered:

$$\hat{\mathbf{h}}_{ML} = \arg \max_{\mathbf{h}} p(\mathbf{r}^\circ | \mathbf{s}^\circ, \mathbf{h}). \quad (3)$$

After the channel coefficients are estimated, the following posterior probability of each transmitted bit can be calculated:

$$p(s_i = s | \mathbf{r}, \hat{\mathbf{h}}_{ML}), i = 1, 2, \dots, N. \quad (4)$$

The output of ML equalizer is then fed into the channel decoder to obtain the estimate  $\hat{\mathbf{m}}$  for information bits  $\mathbf{m}$ .

## III. JOINT NEURAL NETWORK EQUALIZER AND DECODER

### A. Convolutional Neural Network Equalizer

Due to the nonlinear nature of distortion, various neural networks are exploited to realize adaptive equalization. DNN [6] show promising performance. However, the densely connected networks require a large number of parameters and the complexity grows exponentially with the code length. Although recurrent neural network (RNN) [11] can process long code, some performance degradation appears since the neural network is unable to make the best use of the continuous inputs. Compared to other machine learning based equalization schemes, CNN requires minimal data preprocessing and have strong capabilities of feature extraction. A collection of neurons in CNN only respond to a restricted area of given inputs, which is suitable for aforementioned nonlinear channel model since the ISI only exists between consecutive bits of transmitted sequence and the influence of nonlinear distortion is independent for each bit.

The architecture of proposed convolutional neural network equalizer is illustrated in Fig. 1. The adopted CNN is composed of several convolutional layers with rectified linear units (ReLU) while the last layer only contains convolution operation. Under the circumstances of channel equalization, the input of CNN is 1-D real vector instead of typical 2-D

image in the field of computer vision. Hence, we rewrite the 2-D convolution with ReLU into 1-D form:

$$\mathbf{y}_{i,j} = \sigma \left( \sum_{c=1}^C \sum_{k=1}^K \mathbf{W}_{i,c,k} \mathbf{x}_{c,k+j} + b_i \right), \quad (5)$$

where  $\mathbf{W} \in \mathbb{R}^{M \times C \times K}$  denotes the weights tensor of  $M$  filters with  $C$  channels in a layer, each containing  $1 \times K$  sized filter. Besides,  $b_i$  is the  $i$ -th element of the bias vector  $\mathbf{b} \in \mathbb{R}^M$  and  $\sigma(\cdot)$  here denotes the ReLU function  $\max(x, 0)$ .

Different from ML equalizers [10], proposed convolutional neural network equalizer is trained with an offline and end-to-end manner. The received training channel output  $\mathbf{r}^\circ$  and true channel input  $\mathbf{s}^\circ$  are required during training. The optimal parameters  $\hat{\boldsymbol{\theta}} = \{\mathbf{W}, \mathbf{b}\}$  of neural network are estimated instead of the ML criterion  $\hat{\mathbf{h}}_{ML}$ :

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} p(\hat{\mathbf{s}} = \mathbf{s}^\circ | \mathbf{r}^\circ, \boldsymbol{\theta}). \quad (6)$$

The optimal parameters  $\hat{\boldsymbol{\theta}}$  can be obtained by using the stochastic gradient descent (SGD) with back propagation. Once the neural network has been trained, only channel output  $\mathbf{r}$  is fed into the neural network equalizer for inference phase, which provides a blind equalization.

### B. Deep Neural Network Decoder

Deep neural network of [4] shares similar feedforward structure with CNN, but the neurons in a fully-connected layer are densely connected to the previous layer and each neuron associates with its weight. According to [4], deep neural network can be trained to achieve optimal BER performance over short code length. We use a multiple-layer NND to decode the output of convolutional neural network equalizer (see Fig. 1). The input of NND is a 1-D vector from CNN equalizer. The computation of single layer in NND can be formulated as following matrix multiplication:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}), \quad (7)$$

where  $\mathbf{W}$  is the weights corresponding the input  $\mathbf{x}$ . Note that  $\sigma(\cdot)$  denotes the activation function (ReLU or sigmoid). The sigmoid activation  $(1 + \exp^{-x})^{-1}$  is applied to squash the output of NND to range  $(0, 1)$ .

### C. Training

The performance of neural networks greatly depends on the training process. First, the loss function should be carefully selected to give an accurate measure of the distance

between neural network outputs and correct labels. Then the hyperparameters related to network structure and training also determine the capabilities of neural network.

Proposed neural network equalizer and decoder consist of two neural networks. For the CNN equalizer, mean squared error (MSE) is adopted as the equalization loss function:

$$\mathcal{L}(\hat{s}, s) = \frac{1}{N} \sum_i |\hat{s}_i - s_i|^2, \quad (8)$$

where  $\hat{s}$  denotes the estimated equalization output of CNN equalizer and  $s$  is the actual transmitted signal.

For the NND, we use binary cross-entropy (BCE) as the decoding loss function:

$$\mathcal{L}(\hat{m}, m) = \frac{1}{N} \sum_i \hat{m}_i \log(m_i) + (1 - \hat{m}_i) \log(1 - m_i), \quad (9)$$

where  $\hat{m}$  denotes the decoded output of NND and  $m$  is the original information sequence.

Hence, the total loss of these two neural networks becomes:

$$\mathcal{L}_{total} = \mathcal{L}(\hat{s}, s) + \mathcal{L}(\hat{m}, m). \quad (10)$$

The goal of training is to minimize the total loss of Eq. (10). SGD and other deep learning approaches can be utilized to find optimal parameters resulting in the minimal loss.

#### IV. EXPERIMENTS

##### A. Experimental Setup

The proposed models are implemented on the advanced deep learning framework *PyTorch*. GPU (NVIDIA GeForce GTX 1080 Ti) is used to accelerate training. Polar code with 1/2 rate and binary phase shift keying (BPSK) modulation is adopted. Without loss of generality, we use the following notations to define the architecture of neural networks used in this paper. The CNN with  $D$  convolutional layers, the  $i$ -th layer containing  $M_i$  filters with size  $1 \times K$ , is denoted as CNN  $\{M_1, M_2, \dots, M_D\}$ . The deep neural network with  $D$  layers, the  $i$ -th layer containing  $H_i$  nodes, is expressed by DNN  $\{H_1, H_2, \dots, H_D\}$ . It is noted that the first and last layer of DNN are the input layer and the output layer, respectively.

TABLE I  
SUMMARY OF EXPERIMENTAL SETTINGS.

Parameters	Value
CNN structure	$\{6, 12, 24, 12, 6, 1\}, K = 3$
DNN structure	$\{16, 128, 64, 32, 8\}$
Channel	AWGN
SNR Range	0 dB to 11 dB
Training Samples per SNR	20
Testing Samples per SNR	50000
Mini-batch Size	240
Optimizer	Mini-batch SGD with Adam
Learning Rate	Lr=0.001
Weights Initialization	$\mathcal{N} \sim (\mu = 0, \sigma = 1)$

We adopt the structure of CNN  $\{6, 12, 24, 12, 6, 1\}$  with padding size 1 since  $1 \times 3$  filter size can not only maintain the length of input vector but also preserve the dimension

of intermediate feature maps. Besides, this configuration balances complexity and performance. According to our test, no more gain is observed over larger CNN while fewer filters will cause performance degradation. DNN structure is  $\{16, 128, 64, 32, 8\}$ , which has been demonstrated to achieve optimal decoding performance for short polar codes in [4].

The training and testing data are both randomly generated over AWGN channel with ISI and nonlinearity distortion from SNR 0 dB to 11 dB. Each training mini-batch contains 240 samples (20 frames per SNR). 50000 frames per SNR are used for testing. The length of each frame is 8-bit. We use mini-batch SGD with back propagation to update the weights. Learning rate is set to 0.001 and Adam optimizer [12] is utilized to adjust the learning rate dynamically. The weights are initialized with normal distribution  $\mathcal{N}(\mu = 0, \sigma = 1)$ . CNN is trained for 5000 iterations. The experimental settings are summarized in Table I.

##### B. Experiment Results over Linear Channel

In the first experiment, the dispersive channel with ISI and AWGN. To remain consistent with [8, 10, 11], the associated impulse response with length  $L = 3$  can be expressed as the following equation:

$$H(z) = 0.3482 + 8704z^{-1} + 0.3482z^{-2}. \quad (11)$$

There is no strict rules that precisely measure the proper size of neural networks for a specific task. Hence, various configurations of CNN are tested to determine the best structure. The BER results for each configuration is shown in Fig. 2. It can be observed that more filters guarantee better performance under 4-layer CNN (red line). However, deeper structure outperforms the shallow ones with equal or less amount of parameters (blue and green line). CNN  $\{6, 12, 24, 12, 6, 1\}$  (containing 2257 weights) achieves very close performance with CNN  $\{8, 16, 32, 16, 8, 1\}$  (containing 3969 weights) and CNN  $\{32, 64, 32, 1\}$  (containing 12609 weights).

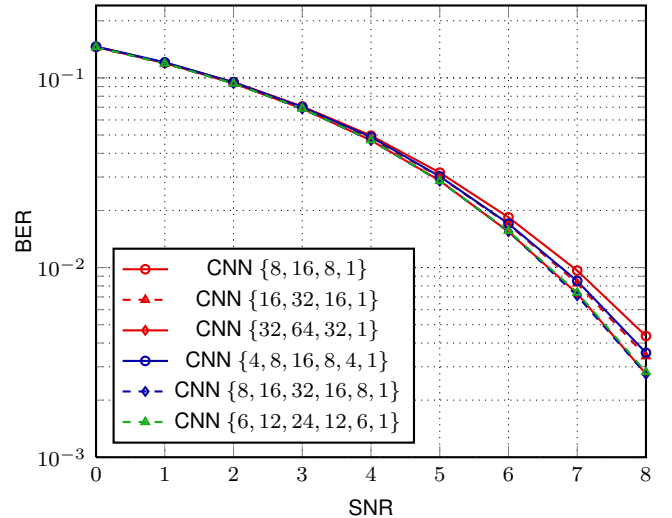


Fig. 2. BER comparison of CNN equalizer with different structures over linear channel.

The comparison between proposed CNN equalizer and other equalization schemes is also conducted (see Fig. 3). The structure of CNN equalizer is  $\{6, 12, 24, 12, 6, 1\}$ . ML-BCJR and Bayesian methods [10] are tested with different lengths of the training sequence (10 and 20). Besides, the performance of ML equalizer with the knowledge of perfect CSI is also depicted. Proposed CNN equalizer approaches to perfect CSI under low SNR range and outperforms both ML-BCJR and Bayesian equalizer by about 0.5 dB.

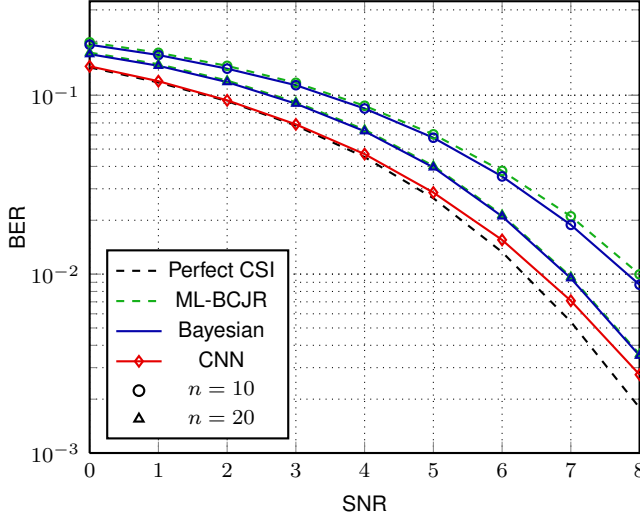


Fig. 3. BER comparison of proposed CNN, Bayesian [10] and ML-BCJR equalizers over linear channel.

### C. Experiment Results over Nonlinear Channel

In the second experiment, the nonlinearity is additionally considered. To intuitively illustrate the nonlinear function of CNN, the decision boundary of proposed equalizer is drawn in Fig. 4. The adopted CNN structure is  $\{6, 12, 24, 12, 6, 1\}$  and the corresponding nonlinear channel is  $\mathbf{h} = [1, 0.5]$  and  $g(v) = v - 0.9v^3$ . The shaded region represents that the hard decision of CNN output is  $\hat{s}_i = +1$  when  $s_i = +1$  transmitted. It can be seen that CNN constructs a nonlinear decision boundary to recover the original symbol from the received signal mixed with ISI and nonlinear distortion. This is similar to that in [9].

In the following sections, the nonlinear function in Eq. (12) is considered, which is the same as [6, 8, 10]:

$$|g(v)| = |v| + 0.2|v|^2 - 0.1|v|^3 + 0.5\cos(\pi|v|). \quad (12)$$

For the case of nonlinear channel, BCJR method requires accurate CSI, and its computational complexity grows exponentially with the code length. Machine learning methods based equalization schemes show promising performance and lower complexity. We compare proposed CNN equalizer with GPC [8] and SVM algorithms over the nonlinear channel in Eq. 2 and ISI in Eq. 11. Consistent with [8], a preamble sequence with 200 training samples is used for channel estimation. As illustrated Fig. 5, CNN outperforms GPC and SVM with approximately 0.2 dB to 0.5 dB gain in terms of BER.

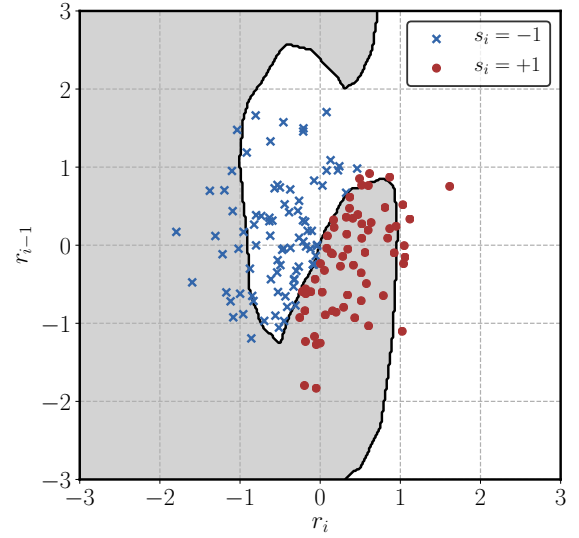


Fig. 4. Decision boundary of proposed CNN equalizer with nonlinear channel  $\mathbf{h} = [1, 0.5]$ ,  $g(v) = v - 0.9v^3$  and SNR = 1.

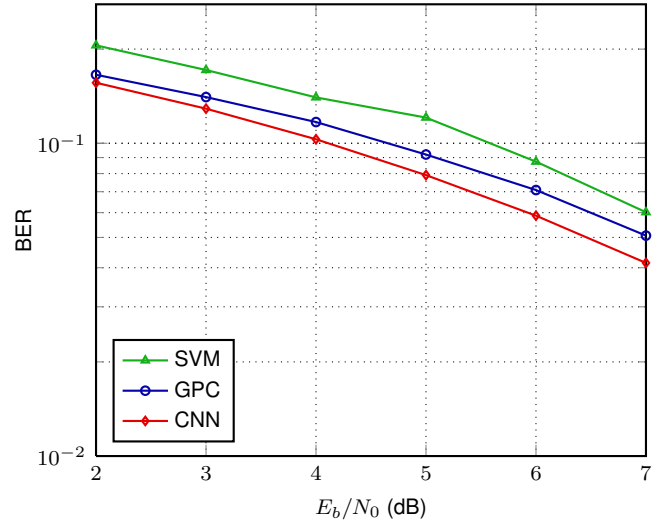


Fig. 5. BER comparison of proposed CNN, SVM and GPC [8] equalizers over nonlinear channel.

### D. Experiment Results of Joint Equalizer and Decoder

In the third experiment, a DNN with structure  $\{16, 128, 64, 32, 8\}$  is introduced as the NND to increase the error correcting capability. The NND receives the soft output of CNN equalizer to recover the originally transmitted bits. The CNN trained over nonlinear channel in Eq. (12) with ISI in Eq. (11) and the NND trained over AWGN channel can be cascaded directly (CNN+NND). But the CNN output may not necessarily follow the same distribution of AWGN channel output, which will potentially cause degradation. Hence, joint training NND using the soft output of CNN equalizer (CNN+NND-Joint) can compensate the performance loss. We test on polar code (16,8) and the baselines are identical with [6]. As shown in Fig. 6, GPC+SC denotes the GPC equalization and successive cancellation (SC) decoding while

DL denotes the deep learning method in [6]. CNN+DNN outperforms GPC+SC with 1 dB, and joint training of CNN and NND has about 0.5 dB gain over CNN+DNN. Performance of CNN+DNN-Joint scheme is very close to DL method, and there is a slight improvement on high SNR. The advantages of proposed CNN+NND is that it requires much less parameters ( $\approx 15,000$ ) than DL [6] ( $\approx 48,000$ ) without increasing the complexity. Moreover, CNN equalizer is also feasible for long codes.

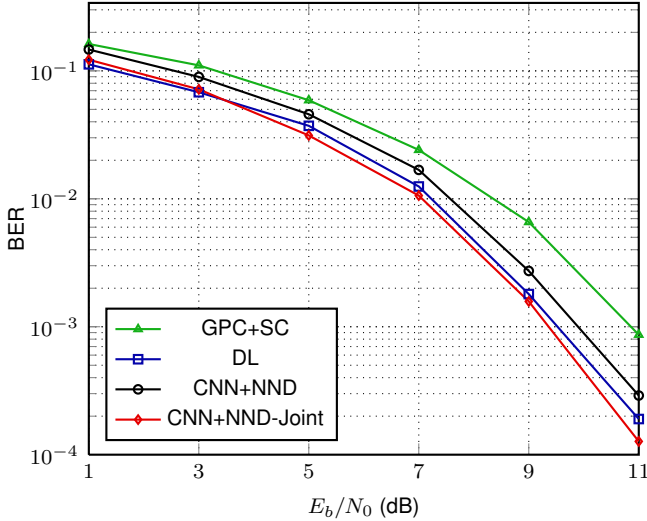


Fig. 6. BER performance of various joint equalization and decoding scheme over nonlinear channel. GPC+SC and DL methods are redrawn from [6].

## V. ANALYSIS

### A. Related Work

CNN is widely used in image denoising [13]. Inspired by this, a CNN is cascaded behind the belief propagation (BP) decoder to deal with correlated Gaussian noise in [14]. However, the limitation is that conventional BP decoder is untrainable thus unable to cope with ISI and nonlinear distortion, which may cause degradation to the whole system. Our proposed CNN equalizer learns the features directly from channel output, providing soft messages to the following channel decoder. This is more robust due to the strong adaptability of neural network. Compared with the DNN equalizer in [6], CNN equalizer trained under short codes can be easily extended to arbitrary code length with negligible degradation. And we consider both ISI and nonlinearities rather than just ISI in [7]. The system architecture is similar to the CNN based blind detection scheme for SCMA in [15]. More complex channel impairments are taken into consideration in this paper.

### B. Complexity Analysis

The complexity of CNN forward inference is determined by filter size  $K$ , input length  $n$  and network depth  $D$ . Proposed CNN equalizer performs 1-D convolution, associating complexity  $\mathcal{O}(n)$ . Once trained, filter size and network depth are fixed. Hence, the overall complexity of CNN equalizer is  $\mathcal{O}(n)$ , indicating that the computational complexity grows

linearly with sequence length. According to [8], the inference complexity of SVM is  $\mathcal{O}(n)$  while GPC requires  $\mathcal{O}(n^2)$  without any optimization.

## VI. CONCLUSION

We present a novel CNN based equalizer to reduce the effects of channel impairments. Then a polar NND is utilized to recover the correct information. The CNN equalizer is adaptive for different channel conditions and shows substantial improvements compared with state-of-the-art results. The future work will focus on the efficient implementation of CNN equalizer and explore the possibility of joint equalization and decoding over long codes based on [2] and [5].

## ACKNOWLEDGEMENT

This work is supported in part by NSFC under grant 61501116, Jiangsu Provincial NSF, the Fundamental Research Funds for the Central Universities, the SRTF of Southeast University, ICRI for MNC, and the Project Sponsored by the SRF for the Returned Overseas Chinese Scholars of MoE.

## REFERENCES

- [1] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.
- [2] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'er, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.
- [3] E. Arkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [4] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2017, pp. 1–6.
- [5] W. Xu, Z. Wu, Y.-L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *IEEE International Workshop on Signal Processing Systems (SiPS)*, 2017, pp. 1–6.
- [6] H. Ye and G. Y. Li, "Initial results on deep learning for joint channel equalization and decoding," in *IEEE Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5.
- [7] A. Caciularu and D. Burshtein, "Blind channel equalization using variational autoencoders," *arXiv preprint arXiv:1803.01526*, 2018.
- [8] P. M. Olmos, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Joint nonlinear channel equalization and soft ldpc decoding with gaussian processes," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1183–1192, 2010.
- [9] D. J. Sebal and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3217–3226, 2000.
- [10] L. Salamanca, J. J. Murillo-Fuentes, and F. Pérez-Cruz, "Channel decoding with a bayesian equalizer," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 1998–2002.
- [11] G. Kechriotis, E. Zervas, and E. S. Manolakis, "Using recurrent neural networks for adaptive communication channel equalization," *IEEE transactions on Neural Networks*, vol. 5, no. 2, pp. 267–278, 1994.
- [12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [13] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 769–776.
- [14] F. Liang, C. Shen, and F. Wu, "An iterative bp-cnn architecture for channel decoding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 144–159, 2018.
- [15] C. Yang, W. Xu, X. You, and C. Zhang, "A blind detection for SCMA based on image processing techniques," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018.