Deep Hybrid Neural Network-Based Channel Equalization in Visible Light Communication

Pu Miao, *Member, IEEE*, Gaojie Chen, *Senior Member, IEEE*, Kanapathippillai Cumanan, *Senior Member, IEEE*
Yu Yao, and Jonathon A. Chambers, *Fellow, IEEE*

***Abstract*—In this letter, the channel impairments compensation of visible light communication is formulated as a time sequence with memory prediction. Then we propose efficient nonlinear post equalization, using a combined long-short term memory (LSTM) and deep neural network (DNN), to learn the complicated channel characteristics and recover the original transmitted signal. We leverage the long-term memory parameters of LSTM to represent the sequence causality within the memory channel and refine the results by DNN to improve the reconstruction accuracy. Results demonstrate that the proposed scheme can robustly address the overall channel impairments and accurately recover the original transmitted signal with fairly fast convergence speed. Besides, it can achieve better balance between performance and complexity than that of the conventional competitive approaches, which demonstrates the potential and validity of the proposed methodology for channel equalization.**

***Index Terms*—Deep learning, nonlinear equalization, visible light communication, hybrid neural network, long-short term memory**

I. Introduction

Visible light communication (VLC) based on light-emitting diode (LED) has gained significant attention for indoor short-range wireless communication [1]. The spectral efficiency of VLC has been increased with the help of high-order modulation, however, it is distorted by the channel impairments, which are mainly derived from the inherent nonlinearity of the LED and the inevitable inter-symbol interference (ISI) from the multipath optical transmission [2].

Various nonlinear post-equalization (NPE) techniques, including model-solving based and feature-learning based schemes, have been proposed to deal with the aforementioned problem. For model-solving approaches [2], the equalization performance mainly depends on the accuracy of the equalizer model and the parameter identification, which will face the cumbersome problems of computational complexity and insufficient accuracy. Deep learning (DL), which shows unparalleled superiority for feature learning with unknown or complex channels, is also applied to optical communication for channel impairments compensation [3].

Comprehensive introduction of DL-based equalization can be found in [4]–[10]. In [4] and [5], a fully connected deep neural network (FC-DNN) was employed at the receiver to demodulate the output signals. Because the learning ability of FC-DNN for memory nonlinearity is limited, the bit error rate (BER) performance is not good. The recurrent neural network (RNN) with long short-term memory (LSTM) cell [6]–[8] is powerful for modeling the nonlinear channel with memory since it could handle the long-term dependencies and store the memory parameters, which are directly related with the channel characteristics. Nevertheless, the convergence speed of the simple LSTM is merely adequate, and the equalization accuracy doesn't possess a good robustness to the noise variation. To learn the useful features, a combined convolutional neural network (CNN) with an LSTM scheme was proposed in [9] and [10]. However, more convolution and pooling layers are introduced to achieve satisfactory accuracy for deep memory scenarios. Thus the network complexity is high, and the training period is severely extended since the inner parameters are intricate. Therefore, the trade-off between the performance metrics is not well achieved, which is one of the critical challenges in practical VLC application [1].

Inspired by the mathematical expression of a Volterra equalizer, an ingenious design of the input form is employed whereby the channel equalization is formulated as a time-sequence with memory prediction problem. A deep hybrid neural network (DHNN) is proposed as an efficient NPE to predict the equalized outputs while guaranteeing the performance balance. Simulation results demonstrate that the proposed scheme can offer excellent BER performance with reduced complexity compared to the existing competitive methods. In addition, the proposed method exhibits relatively fast convergence. It is more robust to the mismatched conditions of training and testing, which shows the applicability of the DHNN for memory nonlinearity compensation in VLC.

II. Problem Formulation

Consider a typical VLC system employing an intensity modulation and direct detection (IM/DD) structure [1]–[3]. After optical-to-electrical conversion in the photodetector (PD), the received electrical signal can be expressed as

$$y(n) = R_{PD}\left(x(n) + I_{DC}\right) * h(n) + \varepsilon(n), \tag{1}$$

This work is supported in part by the Shandong Provincial Natural Science Foundation under Grant ZR2019BF001, by the National Natural Science Foundation of China under Grant 61801257 and 61901241, and by the Key Project of Science and Technology of Hainan under Grant ZDKJ2019003. (*Corresponding author: Gaojie Chen*)

Pu Miao is with the School of Electronic and Information Engineering, Qingdao University, Qingdao, China (e-mail: mpvae@qdu.edu.cn).

Gaojie Chen is with 5GIC & 6GIC, Institute for Communication Systems, University of Surrey, Guildford, UK (e-mails: gaojie.chen@surrey.ac.uk).

Kanapathippillai Cumanan is with the Department of Electronic Engineering, University of York, York, UK (Email: kanapathippillai.cumanan@york.ac.uk).

Yu Yao is with the School of Information Engineering, East China Jiaotong University, Nanchang, China (e-mail: shell8696@hotmail.com).

Jonathon A. Chambers is with the School of Engineering, University of Leicester, Leicester, UK (e-mails: jonathon.chambers@leicester.ac.uk).

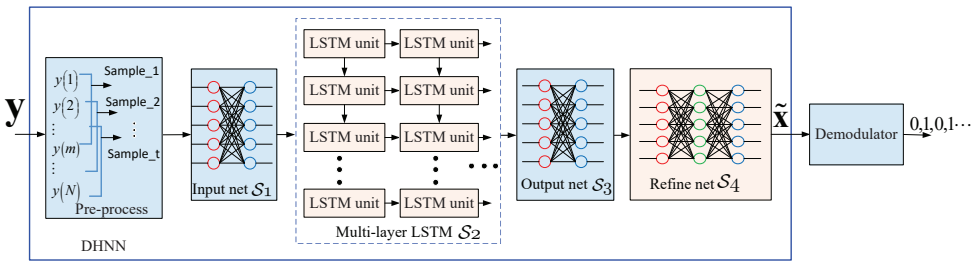

Fig. 1. Schematic of the proposed scheme in VLC system.

where $x(n)$ is the real-valued transmitted signal, R_{PD} is the optoelectronic conversion factor of the PD, I_{DC} denotes the DC component, $h(n)$ is the overall channel impulse response, $\varepsilon(n)$ is the channel noise following Gaussian distribution, and $*$ denotes convolution. Therefore, the overall impairments involved in $y(n)$ will affect the transmission quality of the VLC system.

At the receiver, $y(n)$ is fed into the Volterra-based NPE, then, the corresponding outputs $\tilde{x}(n)$ can be expressed as

$$\tilde{x}(n)=\sum_{p=1}^{P}\sum_{k_1=0}^{L-1}\cdots\sum_{k_p=0}^{L-1}h_p(k_1,\cdots,k_p)\prod_{i=1}^{p}y(n-k_i)+v(n), \tag{2}$$

where L denotes the memory length, P is the nonlinear order, $h_p(k_1,\cdots,k_p)$ is the pth order Volterra kernel, and $v(n)$ is the modeling error. Let

$$\mathbf{c}_1(n)=[y(n),\cdots,y(n-L+1)]^T, \tag{3}$$

which includes the truncated samples with memory L. Then, $\tilde{x}(n)$ can be considered as the sum results of the response for each $h_p(k_1,\cdots,k_p)$ and $\mathbf{c}_p(n)$, shown as

$$\tilde{x}(n)=\sum_{p=1}^{P}\mathbf{c}_p^T(n)\,\mathbf{h}_p+v(n), \tag{4}$$

where $\mathbf{c}_p(n)=\mathbf{c}_{p-1}(n)\otimes\mathbf{c}_1(n)$ for $p\geq 2$, and $\otimes$ is the Kronecker product, and $\mathbf{h}_p$ denote the corresponding kernel coefficients for the $h_p(k_1,\cdots,k_p)$ family. Note that the terms $h_p(k_1,\cdots,k_p)$ are arranged sequentially in $\mathbf{h}_p$ for the index $(k_1,\cdots,k_p)$.

The main goal of the NPE is to undo the overall nonlinearity from $y(n)$ and produce $\tilde{x}(n)$ which has a minimized error with respect to $x(n)$. As seen from (4), the calculation of $\tilde{x}(n)$ is mainly related to $\mathbf{c}_1(n)$, which contains both the current input and the past states of $y(n)$. Thus, we can infer that $\tilde{x}(n)$ can be correctly predicted from $\mathbf{c}_1(n)$ at the nth moment once the kernel coefficients of $\mathbf{h}_p$ are obtained accurately. Therefore, from the perspective of learning and classification, both $\mathbf{h}_p$ and $\tilde{x}(n)$ can be learned from the training samples set $\{x(n),\mathbf{c}_1(n)\}$, and the implementation of the NPE can be considered as a one-dimensional time sequence with memory prediction problem for the VLC channel input, in which the DL approach is well applicable and justified.

III. The Proposed Scheme

To solve the above problem, we propose a DHNN equalization scheme with good learning ability and convergence speed. The model architecture is depicted in Fig. 1. The proposed DHNN is composed of a cascade of several elaborated neural networks, which includes the input subnet $\mathcal{S}_1$, multi-layer LSTM net $\mathcal{S}_2$, the output net $\mathcal{S}_3$ and the refine net $\mathcal{S}_4$. In addition, both $\mathcal{S}_1$ and $\mathcal{S}_3$ employ a simple neural network, $\mathcal{S}_2$ involves an LSTM network with $\mathcal{L}_2$ layers and $\mathcal{S}_4$ deploys the FC-DNN with $\mathcal{L}_4$ layers, respectively. In what follows, we assume that synchronization has been perfectly achieved at the receiver [2].

Let $\mathbf{y}=[y(1),y(2),\cdots,y(m),y(m+1),\cdots,y(N)]$ denotes the original received vector, and $\mathcal{D}_q^{S_2}$ denotes the cell number of the qth layer of $\mathcal{S}_2$. As illustrated in Fig. 1, $\mathbf{y}$ is firstly split into several short overlapping samples by a sliding window in the pre-treatment block, where the window length is m and the sliding step is 1. Moreover, m also represents the time step of the LSTM unit. After $(N-m+1)$ batch sampling in the pre-processing block, the output is formed as

$$\mathcal{Y}=\begin{bmatrix} y(1) & y(2) & \cdots & y(m) \\ y(2) & y(3) & \cdots & y(m+1) \\ \vdots & \vdots & \cdots & \vdots \\ y(N-m+1) & y(N-m+2) & \cdots & y(N) \end{bmatrix}. \tag{5}$$

$\mathcal{Y}\in\mathbb{R}^{(N-m+1)\times m}$ contains time-step vectors, which are firstly fed into $\mathcal{S}_1$ for data shaping to fulfil the input dimension requirements of the following LSTM cell. Note that $\mathcal{S}_1$ is composed of the cascaded sub-layer block including the dense layer and batch normalization layer. After transformation, a *3-D* vector $\overline{\mathcal{Y}}\in\mathbb{R}^{(N-m+1)\times m\times\mathcal{D}_1^{S2}}$ is obtained and fed into $\mathcal{S}_2$. As for $\mathcal{S}_2$, it is composed of a cascaded sub-layer block including multiple LSTM cells. In addition, different numbers of LSTM cells can be deployed in different sub-layers.

The internal structure of a single LSTM cell in one layer is shown in Fig. 2, which contains three *Sigmoid* functions $\sigma(\cdot)$ in terms of forget gate, input gate and output gate. These gates can selectively influence the state of the DHNN at each time step. Moreover, the forget gate is the core of a single LSTM cell, since its output f_t determines the information that should be retained or discarded according to the current cell input y_t at time step t and the previous cell output H_{t-1} at time step $t-1$. The input gate picks up some new information and then adds it into the former state C_{t-1}, shown as

$$C_t=f_tC_{t-1}+i_tz_t, \tag{6}$$

where $f_t\in[0,1]$ and $i_t\in[0,1]$ indicating the proportion of the important information in C_{t-1} and in the temporary cell

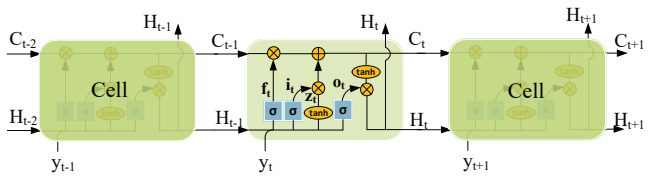

Fig. 2. Internal structure of a LSTM cell in one layer.

state z_t. Then, the output of the LSTM cell is calculated by

$$H_t = \tanh\left(C_t\right)\sigma\left(W_o\left[y_t, H_{t-1}\right] + b_o\right), \tag{7}$$

where W_o and b_o denote the parameter and bias matrix of the output gate, respectively. As a result, the outputs for every time step of this layer are calculated as above-mentioned, then the layer output $\mathbf{H}_1 \in \mathbb{R}^{(N-m+1)\times m\times \mathcal{D}_2^{\mathcal{S}2}}$ is formed accordingly.

After that, $\mathbf{H}_1$ is fed into the corresponding LSTM cells of the next layer for neural computing until the last layer. However, for the last layer, only the H_m at the last time step is selected thus all of them can be composed as the final *2-D* output $\mathbf{H}_{\mathcal{L}_2} \in \mathbb{R}^{(N-m+1)\times \mathcal{D}_{\mathcal{L}_2}^{\mathcal{S}_2}}$. However, $\mathbf{H}_{\mathcal{L}_2}$ should be transformed as a column vector because the main goal of DHNN is to predict the $(N-m+1)$ dimensional vector $\tilde{\mathbf{x}}$ from $\mathbf{y}$. Hence, the net $\mathcal{S}_3$ is applied subsequently to transform $\mathbf{H}_{\mathcal{L}_2}$ into the proper identified data $\tilde{\mathbf{x}}_{\mathcal{L}_2} \in \mathbb{R}^{(N-m+1)}$.

In order to improve the prediction ability, we feed $\tilde{\mathbf{x}}_{\mathcal{L}_2}$ into $\mathcal{S}_4$ for refining the network output. The net $\mathcal{S}_4$ is made up by cascading a sub-layer containing a dense layer, batch normalization and activation function. It is noteworthy that the *Relu* function $\rho_{RL}(\cdot)$ is used for most layers in $\mathcal{S}_4$ except for the last layer, where the linear activation function is employed. In addition, the last layer does not deploy normalization so as to produce the true value of the original transmitted signal. Finally, the equalized $\tilde{\mathbf{x}}$ can be directly obtained.

For the computational complexity, it is worth noting that the calculation of $\mathcal{S}_2$ and $\mathcal{S}_4$ are dominant in each time step. Thus we define the cell number of each layer in $\mathcal{S}_2$ and $\mathcal{S}_4$ be equal to $\mathcal{D}^{S_2}$ and $\mathcal{D}^{S_4}$, respectively. Accordingly, the overall complexity of a DHNN per time step can be approximately expressed as $\mathcal{O}\left(4mD^{S_2} + 4\left(D^{S_2}\right)^2 + 2D^{S_4}\right)$.

We adopt the direct current biased optical orthogonal frequency division multiplexing (DCO-OFDM) as the training symbols in the offline training stage. The model is trained under the IM/DD channel, which involves LED nonlinearity, optical propagation and channel noise. Notice that the channel follows IEEE 802.15.7r1 for indoor environments [11], and the receiving plane is divided into several grid units with equidistant spacing used as potential locations for the PD. Random data are first generated as the transmitted symbols in each simulation. After VLC transmission, the received electrical signals are collected under different PD locations, and split into several short overlapping samples. Every short sequence and one sample of the original transmitted signals are combined as training data. Practically, we should collect a diverse and abundant training set to enhance the parameters learning ability of the DHNN. Moreover, we employ the normalized mean squared error (NMSE) between the raw $\mathbf{x}$ and the predicted $\tilde{\mathbf{x}}$ as the training loss function, as calculated by

$$loss = \frac{\sum \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2}{\sum \|\mathbf{x}\|_2^2}. \tag{8}$$

Furthermore, *TensorFlow* is adopted and the training procedure is implemented on a work station running with a graphics processing unit of NVIDIA GeForce 2080Ti drived by CUDA 10.0. Moreover, the adaptive moment estimation (Adam) is used as the optimizer and the learning rate is fixed to 0.0001.

In the online deployment stage, the well-trained model generates the output that predicts the transmitted signal without explicitly estimating the IM/DD channel. As for testing, only several special links are adopted to evaluate the system performance. For convenience, the four-receiver locations with the root mean square (RMS) delay spread of 7.92, 8.2, 8.3 and 8.9 ns are marked as U_1, U_2, U_3 and U_4, respectively.

IV. Simulation Results

In this section, simulation results are conducted to evaluate the corresponding performance of the proposed scheme. A modulated DCO-OFDM symbol, containing total 512 sub-carriers with 16-based quadrature amplitude modulation (QAM), is randomly generated at the transmitter. The normalized DC is set as 0.4. As for the architecture of DHNN, $\mathcal{S}_1$ and $\mathcal{S}_3$ employ only one dense layer with the neuron size of 128 and 256, respectively. The $\mathcal{S}_2$ involves two LSTM layers with the cell size of $(128, 256)$. The $\mathcal{S}_4$ employs a three-layer FC-DNN where the neuron size of the hidden layer is 50. It should be noted that these hyperparameters are manually determined based on empirical trials, and the batch processing is used for the network input.

A. Convergence Performance

Taking the time step $m = 40$, the proposed scheme is individually trained as the training signal-to-noise ratio (SNR) λ varies from 20 to 50 dB, respectively. The corresponding training loss in terms of the NMSE is presented in Fig. 3. All of the curves tend to become stable gradually with the training epoch increased. Except for the curve of $\lambda = 20$ dB, the other curves eventually achieve an acceptable training performance, *e.g.*, the average final loss of the last 1000 epochs for $\lambda = 30$ to 50 dB are fluctuating around -26.1 to -34.9 dB, which indicates the successful network training since they can meet the requirements of the symbol detection in the QAM constellation. In addition, as shown in the previous 3000 epochs, the convergence speed is increased with the increase of λ, and it achieves the best value in the case of $\lambda = 40$ dB. After that, it is decreased as the λ varies from 45 to 50 dB. Furthermore, the DHNN with $\lambda = 40$ dB just cost about 2100 epochs to converge the NMSE of -30 dB while the other curves need at least 4300 epochs to reach the same NMSE level. The main reason is that the samples with high SNRs would reduce the learning ability of the DHNN to the channel noise, whereas that with low SNRs would weaken the learning of the useful information.

In addition, the time step m determines the length of the input sequence and also the memory ability in the prediction

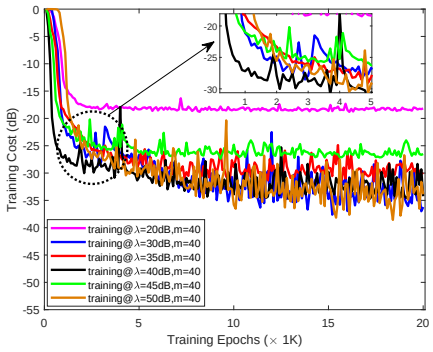

Fig. 3. The cost performance comparison under different training SNRs.

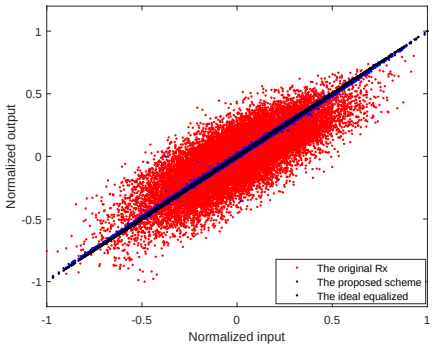

Fig. 5. Normalized amplitudes comparison of the input and output signal.

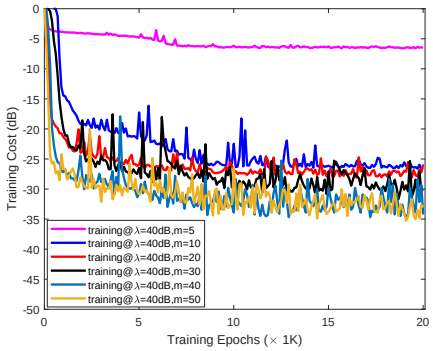

Fig. 4. The cost performance comparison under different time steps.

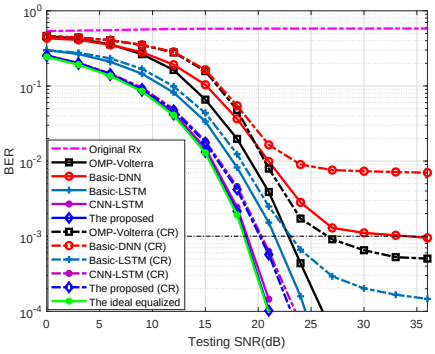

Fig. 6. BER performance comparison of different schemes under U_4.

procedure. Appropriate m is also favorable for the channel characteristics learning by DHNN. Fig. 4 shows the training loss performance as the m varies from the length 5 to 50. The curve of $m = 5$ failed in network training with the final average NMSE of -6.5 dB, whereas the others can achieve the average NMSE between -26.3 to -34.1 dB, showing acceptable prediction precision. The figure illustrates that the larger the m used, the smaller training NMSE is obtained. However, the inner structure of DHNN will become intricate as m is set too large, increasing network complexity. Under the consideration of convergence and training quality, $m = 40$ and $\lambda = 40$ dB is employed in the following studies.

B. Impairments Compensation

By mapping the amplitudes of $x(n)$ and $y(n)$, the channel impairments of U_4 can be illustrated in Fig. 5. In addition, the output of the proposed scheme and the ideal equalization are also depicted here for comparison. Note that the ideal case has the channel information perfectly-known to the receiver. We can observe that the amplitudes of the original received $y(n)$ deviates from the linear straight line and exhibits strong distortion. However, the proposed scheme is slightly diverged from the ideal case and shows an excellent linear relationship with the same channel inputs, indicating that the original inputs can be accurately predicted by the proposed DHNN thus the overall nonlinearity of the LED and the optical wireless channel can be simultaneously mitigated.

The corresponding BER performance comparison is illustrated by the solid line in Fig. 6 as the testing SNR ζ varies from 0 to 36 dB. In addition, the other four equalization schemes in terms of the basic-DNN [4], the basic-LSTM [7], the CNN-LSTM [9] and the conventional Volterra approach based on orthogonal matching pursuit (OMP) [12] are also presented here. In general, the BER performances of the above methods have noticeable improvement as compared with the original received $y(n)$. However, the BER accuracy of the basic-DNN tends to be saturated when ζ is over 27 dB. In addition, both the proposed scheme and the CNN-LSTM are competitive, because they can provide an excellent BER performance among the other methods. Furthermore, they have the similar BER performance to the one for the ideal case, which indicates that the overall channel nonlinearity can be essentially compensated perfectly. Besides that, as for the BER level of 1×10^{-3}, the two competitive schemes can save the required SNR at least by 2.8 dB as compared with the basic-LSTM, by 4 dB for the OMP-Volterra, and by 14 dB for the basic-DNN, respectively. However, the CNN-LSTM costs more hardware resources, and makes small devices unaffordable which are next analyzed.

C. Complexity Analysis

The corresponding application complexity in terms of computational complexity, amount of floating-point operations (FLOPs) and time consumption required for the forward-pass procedure of one OFDM symbol, are shown in Table I,

TABLE I
COMPARISON OF APPLICATION COMPLEXITY

	Complexity [1]	FLOPs	Time
Basic-LSTM	$\mathcal{O}\left(4m\mathcal{D}+4\mathcal{D}^2\right)$	25.64M	1.72e-5s
CNN-LSTM	$\mathcal{O}\left(\mathcal{K}^2\mathcal{C}^2+\mathcal{M}^2\mathcal{C}+4m\mathcal{D}+4\mathcal{D}^2\right)$	572.78M	5.32e-5s
The proposed	$\mathcal{O}\left(4m\mathcal{D}^{S_2}+4\left(\mathcal{D}^{S_2}\right)^2+2\mathcal{D}^{S_4}\right)$	33.81M	1.84e-5s

[1] $\mathcal{K}$ is the size of convolutional kernel, $\mathcal{C}$ is the number of filters, $\mathcal{M}$ is spatial size of the output feature, and $\mathcal{D}$ is the number of LSTM cell.

where the FLOPs are measured based on the frozen graph. Note that the CNN employs one convolutional and pooling layer with $\mathcal{K} = 3$ and $\mathcal{C} = 20$. As the results show, the CNN-LSTM has the highest structural complexity and costs much computation time due to the complex operation in convolutional and pooling layers, which grows quadratically with $\mathcal{K}$, $\mathcal{C}$, $\mathcal{M}$ and $\mathcal{D}$. Thus, it improves the BER performance at the cost of complex signal processing with a huge amount of inner parameters. By comparison, to achieve the equivalent BER performance within the same training epochs, the DHNN needs 33.81 million FLOPs, nearly one-sixteenth of the CNN-LSTM. Therefore, the proposed scheme can effectively balance the performance and application complexity.

D. Robustness Analysis

The above results are obtained based on the fixed conditions. However, the mismatches may occur in practical deployment because the signal will undergo different clipping, ambient noise and delivering links. Therefore, it is very important for the trained DHNN to be relatively robust to these mismatches. As the clipping ratio (CR) is 6 dB, the BER curves of the proposed scheme and the other four methods are demonstrated by the dashdot line in Fig. 6. As compared with the case without clipping, the proposed scheme has the slightest performance deviation, indicating that the proposed scheme could provide the benefits of clipping to the transmitter because of its inherent robustness to the clipping distortions. In addition, although the testing SNR mismatches the training SNR, it doesn't have significant damage on the BER performance. Moreover, the well-trained DHNN is also evaluated for different receiver locations, and the corresponding results are shown in Fig. 7. The BER performances of these four cases are very similar and only have a slight difference for high ζ. Therefore, the proposed scheme can still work effectively and can provide the robust BER performance even though the testing conditions are not exactly the same as those used in the training stage, which shows a good robustness and generalization ability of the proposed scheme.

V. CONCLUSIONS

In this letter, we proposed a novel DHNN-based equalizer for channel impairments compensation in a VLC system. The results confirmed that the proposed scheme is beneficial to mitigate the overall channel nonlinearity, and it can achieve an excellent BER performance improvement with affordable complexity cost, which outperforms the conventional methods by at least 2.8 dB SNR when compared at the BER of 1×10^{-3}. Furthermore, the proposed scheme exhibited its unique advantages in channel characteristics learning, and is more robust to the mismatch conditions of the practical deployment and training stage, which validated the effectiveness and the generalization ability of the DHNN equalizer.

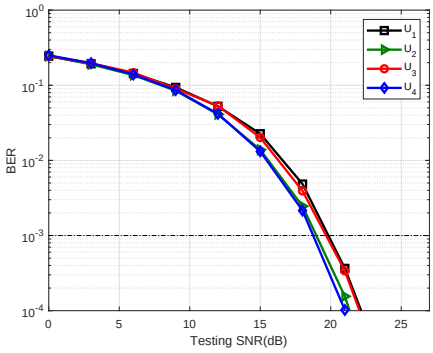

Fig. 7. BER performance of the proposed scheme at different positions.

REFERENCES

[1] L. Jia, F. Shu, N. Huang, M. Chen, and J. Wang, "Capacity and optimum signal constellations for VLC systems," *IEEE/OSA Journal of Lightwave Technology*, vol. 38, no. 8, pp. 2180–2189, Feb. 2020.

[2] P. Miao, G. Chen, X. Wang, Y. Yao, and J. Chambers, "Adaptive nonlinear equalization combining sparse Bayesian learning and Kalman filtering for visible light communications," *IEEE/OSA Journal of Lightwave Technology*, vol. 38, no. 24, pp. 6732–6745, Dec. 2020.

[3] R. Mitra, V. Bhatia, S. Jain, and K. Choi, "Performance analysis of random Fourier features-based unsupervised multistage-clustering for VLC," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2659–2663, June 2021.

[4] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communication Letters*, vol. 7, no. 1, pp. 114–117, Sept. 2017.

[5] X. Gao, S. Jin, C. K. Wen, and G. Y. Li, "Comnet: Combination of deep learning and expert knowledge in OFDM receivers," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627–2630, Oct. 2018.

[6] X. Lu, C. Lu, W. Yu, L. Qiao, S. Liang, A. P. T. Lau, and N. Chi, "Memory-controlled deep LSTM neural network post-equalizer used in high-speed PAM VLC system," *Optics Express*, vol. 27, no. 5, pp. 7822–7833, Oct. 2019.

[7] X. Dai, X. Li, M. Luo, Q. You, and S. Yu, "LSTM networks enabled nonlinear equalization in 50-Gb/s PAM-4 transmission links." *Applied optics*, vol. 58, no. 22, pp. 6079–6084, Oct. 2019.

[8] Y. Zhu, C. Gong, J. Luo, M. Jin, and Z. Xu, "Indoor non-line of sight visible light communication with a Bi-LSTM neural network," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. Dublin, Ireland: IEEE, July 2020, pp. 1–5.

[9] L. Yang, M. Chen, Y. Yang, M. T. Zhou, and C. Wang, "Convolutional recurrent neural network-based channel equalization: An experimental study," in *2017 23rd Asia-Pacific Conference on Communications (APCC)*. Perth, WA, Australia: IEEE, Dec. 2018, pp. 1–6.

[10] Z. Li, F. Hu, G. Li, P. Zou, and N. Chi, "Convolution-enhanced LSTM neural network post-equalizer used in probabilistic shaped underwater VLC system," in *2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. Macau, China: IEEE, Aug. 2020, pp. 1–5.

[11] M. Uysal, F. Miramirkhani, O. Narmanlioglu, T. Baykas, and E. Panayirci, "IEEE 802.15.7r1 reference channel models for visible light communications," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 212–217, Jan. 2017.

[12] G. Zhang, X. Hong, C. Fei, and X. Hong, "Sparsity-aware nonlinear equalization with greedy algorithms for LED-based visible light communication systems," *IEEE/OSA Journal of Lightwave Technology*, vol. 37, no. 20, pp. 5273–5281, July 2019.