

Performance Evaluation of Autoencoder for Coding and Modulation in Wireless Communications

Jialong Xu^{1,3}, Wei Chen^{1,3}, Bo Ai^{1,3}, Ruisi He^{1,3}, Yujian Li^{2,3}, Junhong Wang^{2,3}, Tutun Juhana⁴, Adit Kurniawan⁴

¹State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

²Key Laboratory of All Optical Network and Advanced Telecommunication Network of Ministry of Education, Beijing Jiaotong University, Beijing, China

³School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

⁴School of Electrical Engineering and Informatics, Institute of Technology Bandung, Bandung, Indonesia

Corresponding author: Bo Ai, boai@bjtu.edu.cn

Abstract—The end-to-end autoencoder is a novel and attracting concept to innovate communication system architecture. In its training stage, the end-to-end autoencoder needs differentiable channel models to execute back-propagation algorithm. This shortage impedes the further development of the end-to-end autoencoder. Estimating the channel model by reinforcement learning is a good way to alleviate this problem. However, these training methods increase the difficulty in the training stage. In this paper, we set up a general channel model and employ the measured channel data as input data to alleviate this problem. We compare the performance of the end-to-end autoencoder and Hamming code modulated by Binary Phase Shift Keying under additive white Gaussian noise channel and Rayleigh channel and explore the performance of the end-to-end encoder under the real scenarios. Furthermore, we investigate the performance of the end-to-end autoencoder under mismatched channels. The results demonstrate that the performance of the end-to-end autoencoder is similar with Hamming code and has a significant performance improvement under mismatched channels.

Index Terms—deep learning, modulation coding, performance evaluation

I. INTRODUCTION

The great success of deep learning (DL) in computer vision (CV), automatic speech recognition (ASR) and natural language processing (NLP) in recent years has attracted the attention of non-artificial intelligence scholars and inspired them to handle complex issues by using DL in their respective fields. Within the domain of wireless communications, driven by artificial intelligence, big data, and powerful computation, intelligent communication is considered to be one of the candidate technologies for further development after the fifth generation mobile networks (5G). Previous research focuses on the network layer (e.g., network optimization, resource allocation and management, routing control) and the application layer (e.g., network prediction, facial recognition and data mining). Such upper layer research reveals the incredible power of DL.

This work was supported in part by the Beijing natural Haidian joint fund under Grant L172020, National key research and development program under Grant 2016YFE0200900, the National Science Foundation of China under Grant 61725101, U1834210 and 61961130391, the Royal Society Newton Advanced Fellowship under Grant NA191006, Major projects of Beijing Municipal Science and Technology Commission under Grant Z181100003218010, State Key Lab of Rail Traffic Control and Safety under Grant RCS2018ZZ007 and RCS2019ZZ007.

It promotes us to explore the method of combining DL with the physical layer and dig out the potential of DL on the physical layer.

With the progress of coding technologies, Turbo code, LDPC code, and Polar code are already close to the theoretical bound. It seems to show that there is little room to improve the coding performance. However, Polar code is proved to achieve channel capacity only under binary symmetric channel (BSC) and binary erasure channel (BEC). Turbo code and LDPC code are not proved to achieve channel capacity till now. These coding methods under complex channels and specific constraints(e.g., latency constraint, power constraint) are not guaranteed to be optimal. Considering the environment of the real wireless communication, the difficulty of capturing tractable mathematical channel models in complex communication scenarios, the imperfections and non-linearities of the practical hardware device give DL an opportunity to shine in the physical layer of wireless communications.

The conventional wireless communication system architecture adopts the divide-and-conquer strategy to reduce system complexity and focus on the specific function of each block. This architecture receives great success in the progress of wireless communication. Nevertheless, this architecture is considered sub-optimal [1] (e.g., 8-PSK trellis codes for a Rayleigh channel in [2]). The original intention of communication is not to construct complex steps to transform messages but to transfer messages directly from source to sink. An innovative perspective considers wireless communication system as an end-to-end autoencoder [3]. The experiments in [3] show that the end-to-end autoencoder can achieve similar performance with the conventional method in additive white Gaussian noise (AWGN) channel without needing any domain specific information. Based on this viewpoint, the recurrent neural network (RNN) in [4] is used to optimize encoder and decoder under AWGN channel with feedback and obtain a more reliable code than state-of-the-art known codes. [5] extends the work about 2-user interference AWGN channel in [3] to k -user ($k \geq 2$) interference AWGN channel and achieves the effect of interference strength. [6] and [7] combine the end-to-end autoencoder with orthogonal frequency division multiplexing (OFDM) [8] and MIMO [9] to design code, respectively. All of the aforementioned works evaluate the

DL code performance under AWGN channel or Rayleigh channel, which ignores the advantage of DL coding. To adapt the real channel, [1] trains the end-to-end autoencoder under a stochastic channel model with simulated data at first stage and then fine tune the receiver with over-the-air transmission data. [10] utilizes supervised learning to train receiver and reinforcement learning to train transmitter without channel model. However, both [1] and [10] in the training stage need the source data at receiver end as the labels of supervised learning, which are provided additionally by a reliable channel without error. [11] and [12] generate channel network instead of channel models by using generative adversarial net (GAN). While this system should be placed in the real scenario until the training stage is finished. Due to a long time for training, it is difficult to be implemented, especially in high-speed mobile scenarios [13] [14].

In the past few decades, large amount of channel response data have been measured. Human experts combined these data with their domain knowledge to derive channel property and generate various kinds of mathematical channel models. Inspired by data-driven DL, regarding channel response data as input data successfully circumvents the training of the end-to-end autoencoder under real scenarios and makes the training stage easy to be implemented, which goes back to the original end-to-end autoencoder model.

In this paper, we focus on the performance evaluation of autoencoder under different channels. The main contributions of this paper are as follows:

- We employ real channel response data as input data to train the end-to-end communication system.
- We propose a flexible coding method based on the end-to-end autoencoder. This method supports any information rate, while traditional coding method fails.
- We evaluate the end-to-end autoencoder under both analytic channels and real channels, which shows a comprehensive evaluation method of the end-to-end autoencoder, and we demonstrate under channel mismatch conditions, the end-to-end autoencoder achieves a better performance.

Notations: We use boldface lower-case letters to denote column vectors, upper-case letters to matrix. \mathbb{R} and \mathbb{C} denote the sets of real and complex numbers, respectively. \mathbb{R}^n and \mathbb{C}^n denote n-dimensional vector space of real and complex numbers, respectively. $\mathbb{CN}(\mathbf{m}, \Sigma)$ denote the real and complex multivariate Gaussian distributions with mean \mathbf{m} and covariance matrix Σ . The symbol \odot means element wise product. The expression $s_j^{(i)}$ denotes the j -th element of the i -th sample.

II. END-TO-END AUTOENCODER

End-to-end communication system is regarded as an improvement that can be end-to-end optimized by DL algorithms. An end-to-end autoencoder is shown in Fig. 1. It deals with channel data as the inputs of intermediate layers. In this model, the transmitter is represented by the encoder and the receiver by the decoder. The channel effect is introduced by the channel layers. Table I lists the configuration of each module we adopt in this paper.

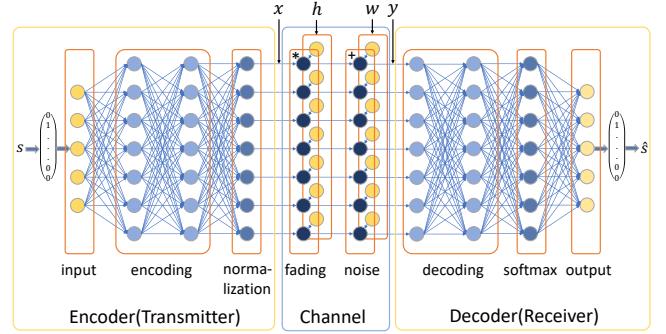


Fig. 1. End-to-end autoencoder model.

TABLE I
LAYOUT OF THE PROPOSED END-TO-END AUTOENCODER

Component	Layer	Activation Function	Output Dimension
Transmitter	Input		M
	Dense	ReLU	M
	Dense	Linear	n
	Normalization		n
Channel	Fading		n
	Noise		n
Receiver	Dense	Linear	M
	Dense	Relu	M
	Dense	softmax	M

A. Encoder

The input symbol s is one out of $M = 2^k$ messages the transmitter want to transmit to the receiver. s is encoded as an M-dimensional one-hot vector and fed into the Input layer. The encoder realizes a transformation $f : \mathbb{R}^k \rightarrow \mathbb{R}^n$ to transfer symbol $s \in \mathbb{R}^k$ to signal $x = f(s) \in \mathbb{R}^n$. The encoder consists of a dense layer of M units with ReLU activation function [15], followed by a dense layer of n units with linear activation. Then we impose an average energy constraint $\mathbb{E}[\|x\|^2] = n$. Different from the purpose of autoencoder in DL domain, which represents its input by low dimensional data to improve representation efficiency, making $n \geq k$ increases some redundancy for x . It can fight against the channel impairments better. The dimension n of transmitting signal x represents sending a symbol s needs n discrete uses of the channel. The communication rate of the end-to-end autoencoder $R = k/n$ represents that one use of channel can transform R bit information.

B. Channel

Channel layers inserted between encoder and decoder represents the channel perturbation when the output of the encoder x passes the wireless channel shown in Fig. 1. The channel is considered as a flat fading channel. One tap of impulse response can represent its characteristic. The channel consists of two layers, the fading layer and the noise layer. The channel effect is represented as:

$$\mathbf{y} = \mathbf{h} \odot \mathbf{x} + \mathbf{w}. \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the output of the encoder, $\mathbf{h} \in \mathbb{R}^n$ inputed to fading layer is the impulse response data, \mathbf{w} represents AWGN and its distribution is $\mathcal{CN}(0, \sigma^2 \mathbf{I})$. The assumption about AWGN means that the primary source of the noise is at the receiver and is independent of the paths over which the signal is being received [16]. Note that if we use both in-phase channel and quadrature channel to transmit two dimension of signal \mathbf{x} each time, the variance of the noise $\sigma^2 = (RE_b/N_0)^{-1}$. If we only use in-phase channel (or quadrature channel) to transmit one dimension of the signal \mathbf{x} each time, the variance of the noise $\sigma^2 = (2RE_b/N_0)^{-1}$. E_b/N_0 denotes the energy per bit (E_b) to noise power spectral density (N_0) ratio.

C. Decoder

The receiver applies a transformation $g : \mathbb{R}^n \rightarrow \mathbb{R}^k$ to transfer the noised signal $\mathbf{y} \in \mathbb{R}^n$ to message $\hat{\mathbf{s}} \in \mathbb{R}^k$, which is an estimation of transmitted message \mathbf{s} . The decoder consists of two dense layers of M unites with RELU activation function, followed by one dense layer with softmax activation. The output of the decoder is a probability distribution consisting of M probabilities. The index of the highest probability is then converted to the decoded message $\hat{\mathbf{s}}$. Training the end-to-end autoencoder is an important part for this proposed system. This end-to-end autoencoder selects categorical cross entropy

$$\mathcal{L} = - \sum_{i=1}^N (s_1^{(i)} \log \hat{s}_1^{(i)} + s_2^{(i)} \log \hat{s}_2^{(i)} + \cdots + s_M^{(i)} \log \hat{s}_M^{(i)}), \quad (2)$$

as its loss function. here, $\mathbf{s} = (s_1; s_2; \dots; s_m)$ and $\hat{\mathbf{s}} = (\hat{s}_1; \hat{s}_2; \dots; \hat{s}_m)$. We use Adam [17] to execute gradient descent algorithm.

III. SIMULATION RESULTS AND DISCUSSION ON DIFFERENT CHANNEL

In this paper, a simulation of the conventional wireless communication system architecture using Hamming code as coding method and BPSK as modulation method is done by Matlab, which provides a benchmark performance. We use Tensorflow [18] and its high-level API Keras [19] to implement the end-to-end autoencoder with the parameters listed in Table I. Although the input symbol \mathbf{s} has M finite states, the output signal of channel \mathbf{y} is generated by the random feature of the channel acting on the encoded signal \mathbf{x} . This characteristic of the end-to-end autoencoder is regarded that we have a training dataset with infinite size in training stage. Furthermore, the random feature of the channel can be regard as a form of regularization to avoid the overfitting problem. We adopt the strategy that training is done at a fixed E_b/N_0 value and test is done at arbitrary E_b/N_0 values. In the training stage, we train the end-to-end autoencoder 100 epochs, each of which uses different training symbols of 10^6 , batch size with 10^3 and Adam optimization with a learning rate of 10^{-3} . The lowest loss value model is picked up as the final model. In the test stage, we use test symbols of 10^7 to test block error rate (BLER).

A. AWGN Channel

AWGN channel is widely used in coding theory to evaluate coding performance. We train a set of autoencoders at a fixed value $E_b/N_0 = 7\text{dB}$.

Fig. 2(a) shows the BLER of the end-to-end autoencoder and Hamming Code with BPSK at the same rate $R=4/4$ and $R=4/7$. To get the coding gain clearly, we also simulate an uncoded BPSK (4,4). The decoder of Hamming (7,4) uses the maximum likelihood decoding (MLD) to get the optimal performance. The autoencoder does not know any prior knowledge of the encoder and decoder function. At the rate $R = 4/4$, the BLER of Autoencoder (4,4) outperforms the uncoded BPSK (4,4) at the same E_b/N_0 . It shows that improving the complexity of coding and decoding can bring coding gain. At the rate $R = 4/7$, Autoencoder (7,4) gets almost the same BLER as Hamming (7,4). Fig. 2(b) compares the BLER from Autoencoder (4,2) to Autoencoder (4,8). As the length of the encoded signal \mathbf{x} grows longer, the BLER becomes lower. The gain of the BLER comes from the reduced source transmission rate. With the length of \mathbf{x} growing, the coding gain decreases, and finally vanishes. The receiver can not get useful information from the distorted channel due to low transmit energy allocated to each channel. Fig. 2(c) shows the average throughput with different rate R . Although Autoencoder (2,4) has the highest BLER, its rate R is also high. At the same channel rate, Autoencoder (2,4) can transmit two times as much information as Autoencoder(4,4) and four times as much information as Autoencoder (8,4). Fig. 2(c) also reflects the average throughput of Hamming (7,4) is higher than that of Autoencoder (7,4) at $E_b/N_0 \leq 4\text{dB}$. Autoencoder can get an arbitrary rate by tuning the input number and output number of transmitter, while the Hamming code lacks flexibility in transmission rate R.

B. Rayleigh Channel

Rayleigh fading channel is based on a reasonable assumption that there are a large number of reflected objects and scattered objects around the transmitter and the receiver. The channel coefficients here is assumed not to change from one codeword to the next codeword. The receiver is assumed to have perfect CSI. We train a set of autoencoders at a fixed value $E_b/N_0 = 15\text{dB}$.

Fig. 3(a) shows that Autoencoder (7,4) achieves the same BLER as the Hamming (7,4) code with MLD, which is similar to the results shown in AWGN channel. Compared with Uncoded BPSK (4,4), the coding gain of Autoencoder (4,4) is relatively less. It is due to the fact that the channel gain is random and there is a significantly high probability that the channel is in a deep fade. Fig. 3(b) compares the BLER of autoencoders with different rate R and Fig. 3(c) compares the average throughput. Both of them show the similar results as in Fig. 2(a) and Fig. 2(c). Because of the deep fade, the gap of the BLER shown in Fig. 3(b) between adjacent end-to-end autoencoders in Rayleigh channel is less than in AWGN channel.

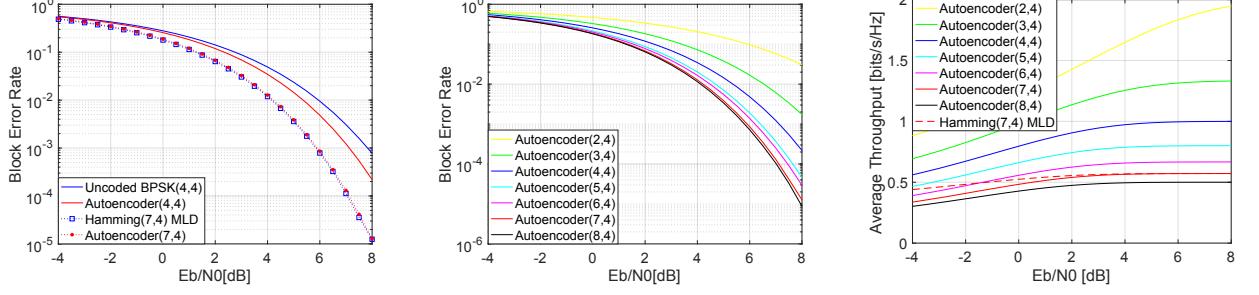


Fig. 2. Performance on AWGN channel

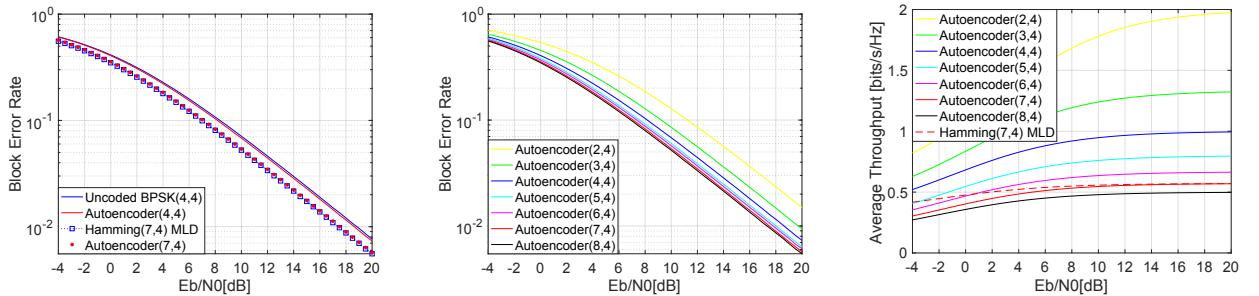


Fig. 3. Performance on Rayleigh channel

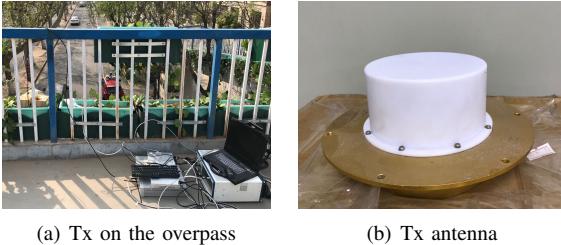


Fig. 4. The equipments of transmitter.

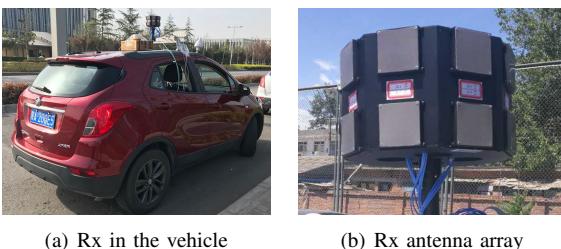


Fig. 5. The equipments of receiver.

C. Real Channel Under High-speed mobile scenario

We adopt the measurement system described in [20] to measure real channel data in urban and suburban scenarios, respectively. Channel measurement is conducted with a bandwidth of 30 MHz at 5.9 GHz. We set up a vehicle-

to-infrastructure (V2I) scenario to simulate transmitting data from an infrastructure to a vehicle. In these scenarios, the transmitter is fixed on a wayside pole or an overpass. The transmitter (Tx) antenna is placed with a height range from 2 m to 10 m above the ground shown in Fig. 4(a). The receiver is fixed in the vehicle and the Receiver (Rx) antenna is fixed on the roof of the vehicle with a height of 1.7 m above the ground shown in Fig. 5(a). The Tx antenna shown in Fig. 4(b) is an omnidirectional antenna and the Rx antenna shown in Fig. 5(b) is a 16-element dual-polarized cylindrical antenna array. The vehicle first moves toward the infrastructure, then goes through the infrastructure, and finally moves away from the infrastructure. The vehicle moves with a speed of 60 km/h.

The measurements in the urban and the suburban scenarios are conducted at Shibanfang South Road and Beihuqu Road in Beijing, China, respectively. The measurement paths and scenarios are given in Fig. 6. The urban scenario is a typical road in the urban area with two-way 8 lanes. The width of the road is 30 meters. Dense high-rise buildings are on both sides of the road. A large number of vehicles drive on the road during the measurements. Sometimes line of sight (LOS) path is blocked by large vehicles. The suburban scenario is a road in the suburban area which has 8 lanes with the width about 30 meters wide as well. There are a lot of vegetation on both sides of the road. Low-rise buildings concentrated in one part of this area. A small number of vehicles drive on

the road during the measurements, and the LOS path almost always exists. The wideband channel data are measured under these two scenarios, respectively.



Fig. 6. Measurement scenarios and routes. (a) Urban. (b) Suburban.

In this experiment, We transform the wideband channel data to 512 narrowband channel data in frequency domain and try to train a general autoencoder to transmit signal at different narrowband with the fixed value $E_b/N_0 = 15\text{dB}$. The large-scale fading is compensated. Fig. 7 shows the BLER of Hamming code and Autoencoder code in urban scenario and suburban scenario, respectively. The BLER of Hamming (7,4) in suburban scenario is around 1 dB lower than the BLER of Hamming (7,4) in urban scenario when $E_b/N_0 \leq 7\text{ dB}$. As E_b/N_0 gradually increases from 7 dB to 10 dB, the gap of the BLER in the two scenarios decreases. At high $E_b/N_0 \geq 18\text{ dB}$, the BLER in the two scenarios are almost same. These results show that the small scale fading in the urban scenario is more drastic than that in the suburban scenario, which has a strong effect on BLER at low E_b/N_0 and has a weak effect on BLER at high E_b/N_0 . Compared with Fig. 3(a), transmitting signal in suburban and urban channel suffers more severe channel fading and gets higher BLER than that in Rayleigh channel. Comparing Hamming (7,4) with Autoencoder (7,4) in the urban scenario, Autoencoder (7,4) has slightly higher BLER than Hamming (7,4). The same result reflects in the suburban scenario. The experiments in AWGN channel and Rayleigh channel reveal that by using the autoencoder, we can achieve the comparable performance with Hamming (7,4) code. However, when we use the autoencoder in the urban and suburban scenarios, we get a slightly weaker performance than Hamming (7,4). A reasonable explanation is that the capacity of the model designed in this experiment is insufficient. The hypothesis space we set does not include the optimal function for the real channel. Building a wider and deeper neural network is a feasible approach to solve this problem.

In wireless communication system, channel estimation is used to improve system performance. Many performance analysis experiments assume the CSI can be perfectly obtained and there are no errors or offset of CSI. However, in a practical wireless communication system, channel mismatch

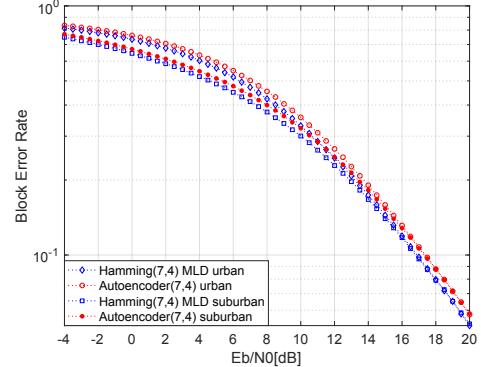


Fig. 7. BLER vs E_b/N_0 of Hamming(7,4) and Autoencoder(7,4) under urban scenario and suburban scenario.

is caused by the noise, the finite length of the pilot, the inaccuracy feedback CSI, imperfect hardware and etc. Channel mismatch especially exists in OFDM system that sub-carrier CSI is needed to be known in all sub-carries and in massive MIMO system that CSI is needed to be known in all pairs of transmitter and receiver antennas. Based on this issue, we assume the receiver can only get an estimate value $\hat{\mathbf{h}}$ of the real channel response:

$$\mathbf{h} = \hat{\mathbf{h}} + \tilde{\mathbf{h}}, \quad (3)$$

where $\hat{\mathbf{h}} \in \mathbb{R}^n$ represents the real channel response, $\tilde{\mathbf{h}} \in \mathbb{R}^n$ represents the estimation error vector. Each \tilde{h}_i is with zero mean and estimation error variance denoted by $\sigma_{\tilde{h}}^2$, $\tilde{\mathbf{h}} \sim \mathcal{CN}(0, \sigma_{\tilde{h}}^2 \mathbf{I})$.

Fig. 8(a), Fig. 8(b) and Fig. 8(c) show plots of the BLER vs E_b/N_0 of Hamming (7,4) and Autoencoder(7,4) at $\sigma_h^2 = 0.1$, $\sigma_h^2 = 0.05$, $\sigma_h^2 = 0.01$ in urban scenario, respectively. Fig. 8(a) shows the BLER of Hamming (7,4) and Autoencoder (7,4) are almost the same when $E_b/N_0 \leq 4\text{ dB}$. As E_b/N_0 gradually increases from 4 dB to 20 dB, the BLER of Autoencoder (7,4) is lower than the BLER of Hamming (7,4) and the gap between the two becomes bigger. Fig. 8(b) and Fig. 8(c) reflect the similar results and the only difference is the demarcation point $E_b/N_0 = 6\text{ dB}$ at $\sigma_h^2 = 0.05$ and $E_b/N_0 = 13\text{ dB}$ at $\sigma_h^2 = 0.01$. Fig. 9(a), Fig. 9(b) and Fig. 9(c) show plots of the BLER vs E_b/N_0 of Hamming (7,4) and Autoencoder(7,4) at $\sigma_h^2 = 0.1$, $\sigma_h^2 = 0.05$, $\sigma_h^2 = 0.01$ in suburban scenario, respectively. Among these pictures, as the E_b/N_0 grows, the Autoencoder (7,4) shows a better performance than Hamming (7,4). Both Fig. 8 and Fig. 9 show that the more severe the channels mismatch, the better performance of the autoencoder in real channels we measure.

IV. CONCLUSION

In this paper, we have presented an end-to-end autoencoder architecture and learned an end-to-end communication system. By using this architecture, we have evaluated the performance of the Autoencoder code and Hamming code under AWGN channel, Rayleigh channel and real channels. This research has shown that the performance of the Autoencoder code is similar with the performance of Hamming code under

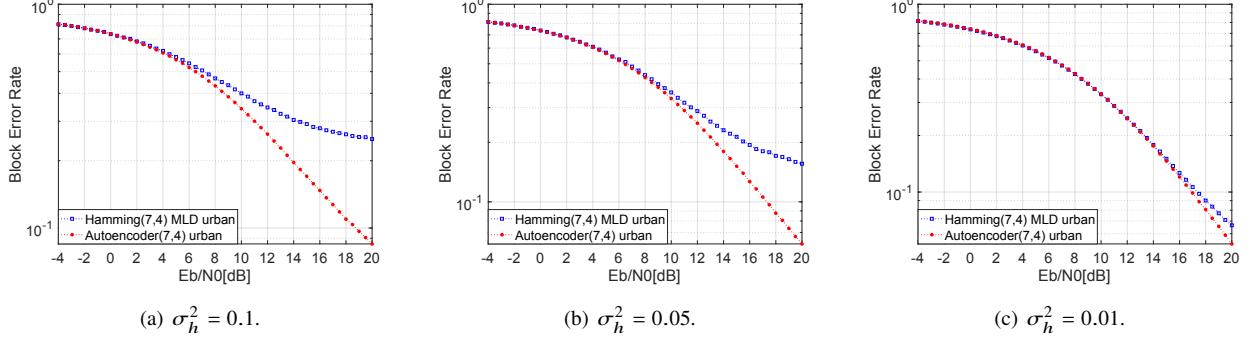


Fig. 8. Channel mismatch in urban scenario.

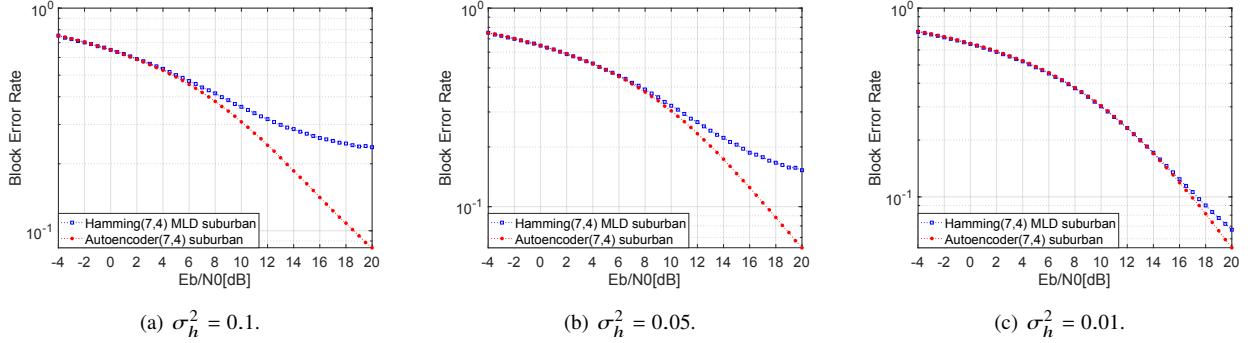


Fig. 9. Channel mismatch in suburban scenario.

AWGN channel and Rayleigh channel, and the performance of the Autoencoder code is slightly worse than the performance of Hamming code under high-speed scenarios. However, under channel mismatch conditions, the autoencoder code has shown a better performance than Hamming code. It reveals the power of DL in the physical layer. The arbitrary coding rate the autoencoder can generate endows the autoencoder a strong ability to match channel characteristics.

REFERENCES

- [1] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep learning based communication over the air,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, Feb 2018.
- [2] E. Zehavi, “8-psk trellis codes for a rayleigh channel,” *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, May 1992.
- [3] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.
- [4] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, “Deepcode: Feedback Codes via Deep Learning,” *arXiv e-prints*, p. arXiv:1807.00801, Jul 2018.
- [5] D. Wu, M. Nekovee, and Y. Wang, “Deep Learning Based Autoencoder for Interference Channel,” *arXiv e-prints*, p. arXiv:1902.06841, Feb 2019.
- [6] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, “Ofdm-autoencoder for end-to-end learning of communications systems,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jun 2018, pp. 1–5.
- [7] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Physical layer deep learning of encodings for the mimo fading channel,” in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2017, pp. 76–80.
- [8] Bo Ai, Zhi-xing Yang, Chang-yong Pan, Jian-hua Ge, Yong Wang, and Zhen Lu, “On the synchronization techniques for wireless ofdm systems,” *IEEE Transactions on Broadcasting*, vol. 52, no. 2, pp. 236–244, Jun 2006.
- [9] B. Ai, K. Guan, R. He, J. Li, G. Li, D. He, Z. Zhong, and K. M. S. Huq, “On indoor millimeter wave massive mimo channels: Measurement and simulation,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 7, pp. 1678–1690, Jul 2017.
- [10] F. A. Aoudia and J. Hoydis, “End-to-end learning of communications systems without a channel model,” in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct 2018, pp. 298–303.
- [11] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan, “Channel agnostic end-to-end learning based communication systems with conditional gan,” in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1–5.
- [12] T. J. O’Shea, T. Roy, N. West, and B. C. Hilburn, “Physical layer communications system design over-the-air using adversarial networks,” in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep 2018, pp. 529–532.
- [13] B. Ai, X. Cheng, T. Kürner, Z. Zhong, K. Guan, R. He, L. Xiong, D. W. Matolak, D. G. Michelson, and C. Briso-Rodriguez, “Challenges toward wireless communications for high-speed railway,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2143–2158, Oct 2014.
- [14] B. Ai, K. Guan, M. Rupp, T. Kurner, X. Cheng, X. Yin, Q. Wang, G. Ma, Y. Li, L. Xiong, and J. Ding, “Future railway services-oriented mobile communications network,” *IEEE Communications Magazine*, vol. 53, no. 10, pp. 78–85, Oct 2015.
- [15] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, Apr 2011, pp. 315–323.
- [16] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.
- [17] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv e-prints*, p. arXiv:1412.6980, Dec 2014.
- [18] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [19] F. Chollet *et al.*, “Keras,” <https://keras.io>, 2015.
- [20] M. Yang, B. Ai, R. He, L. Chen, X. Li, J. Li, Q. Wang, B. Zhang, and C. Huang, “A cluster-based 3d channel model for vehicle-to-vehicle communications,” in *2018 IEEE/CIC International Conference on Communications in China (ICCC)*, Aug 2018, pp. 741–746.