A new design of channel denoiser using *residual autoencoder*

Soyoung Han,[1] Junghyun Kim,[2,✉] and Hong-Yeop Song[1]
[1]*Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea*
[2]*Department of Artificial Intelligence, Sejong University, Seoul, South Korea*
✉Email: j.kim@sejong.ac.kr

A joint neural network decoder and denoiser scheme demonstrated superior performance compared to individual modules. However, there is still a limitation that the existing denoisers cannot effectively learn patterns of encoded signals. To overcome the limitation, a novel denoiser based on a residual autoencoder structure is proposed. The proposed denoiser speeds up the training process and boosts the performance due to its structure effectively extracting compressed features. For the evaluation, a joint system model with a hyper-graph-network decoder that is known for outstanding decoding performance is considered. Simulation results show that this denoiser outperforms the existing denoisers. Furthermore, the proposed joint model shows significant performance improvement compared to the individual hyper-graph-network decoder with only 1% of the number of epochs for the training.

Introduction: Deep learning (DL) has been attracting attention as a technology with great potential to revolutionize communication systems. For instance, at the physical layer, DL shows remarkable progress in both performance and efficiency of channel estimation [1, 2], positioning [3], beam prediction [4], and channel state information (CSI) feedback [5]. In addition, for channel coding, especially for short linear block codes, there are several previous works [6, 7] that demonstrate the effect of DL by improving decoding performance. Recently, hyper-graph-network (HGN) decoder [8] shows better performance than existing works for short block-length codes. However, due to the high complexity, there is a need for improvement in terms of training time.

Neural network (NN) denoiser has been proposed as a technology to remarkably improve decoding performance with only low additional complexity by removing noise in code block level rather than symbol level. In [9], three types of denoisers with multi-layer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN) are proposed for Bose–Chaudhuri–Hocquenghem (BCH) codes. In [10], a joint denoiser and decoder model with a residual NN structure is proposed for Polar codes. However, existing models still have a limitation in that they can not effectively learn patterns of encoded signals.

To address the issue, we design a new NN denoiser. The proposed denoiser is based on a denoising autoencoder (DAE) [11] that performs well for denoising in many applications. More specifically, our denoiser consists of stacked 1D-convolutional layers with double nested residual skip connections. We combine it with the HGN decoder, known as one of the NN decoders with excellent performance, and train the denoiser and decoder jointly using a multi-task learning strategy [10, 12]. Our experiments show the effectiveness of the proposed model by achieving significant performance improvement.

Neural network decoder: The authors in [6] proposed an NN decoder that generalizes the belief propagation (BP) decoder. The layers of the NN decoder are associated with the messages sent by either the check nodes or the variable nodes of the BP decoder. In contrast to the BP decoder, which uses universally fixed weights, the weights in the NN decoder can be tuned to yield near-optimal performance. Later, an improved NN decoder [7] was introduced which used an offset min-sum algorithm instead of sum-product algorithm [13]. More recently, there is a further upgrade to the HGN decoder. The BP algorithm is transformed into a graph NN by replacing each variable node with a learnable NN g. Applying the "hypernetwork" method [14], in which one network predicts weights of the other, the weights of the networks g of each variable node are determined by the hypernetwork f.

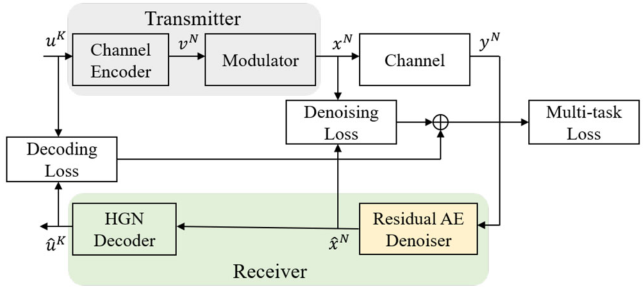

Fig. 1 *A system model for the proposed residual AE denoiser*

Neural network denoiser for channel codes: In communication systems, the NN denoiser for removing noise from a received signal is proposed in [9]. The authors utilize three different types of NNs, that is, multi-layer perceptron (MLP), convolutional neural network (CNN), and long short-term memory (LSTM). They are concatenated with a traditional BCH decoder and help the decoder improve the bit error rate (BER) and block error rate (BLER) performances significantly. After that, residual learning denoisers (RDs) are introduced for polar codes [10] and combined with the existing NN decoder. Simulation results verify that the joint models show better performance in comparison to the decoder without denoiser as the denoiser enhances received SNRs.

Denoising autoencoder: DAE is a module for learning end-to-end mapping from a corrupted input to a corresponding clean version. The DAE corrupts the input with random noise at the training stage to make the model robust to data with noise or large variation. The basic architecture includes a nonlinear stage and a linear stage as:

$$\begin{aligned} h(y) &= \sigma(W_1 y + \mathbf{b}_1), \\ \hat{x} &= W_2 h(y) + b_2, \end{aligned} \tag{1}$$

where y and $\hat{x}$ are noisy and denoised signal, respectively. W_1 and W_2 are weight matrices. $\mathbf{b}_1$ and b_2 are bias vectors for the encoding and decoding parts, respectively. $\sigma(\cdot)$ is a nonlinear function of hidden neurons. The model parameters are optimized to minimize the reconstruction error, which can be assessed by using loss functions.

Proposed residual AE denoiser: A system model for the proposed residual autoencoder (AE) denoiser is depicted in Figure 1. Based on the system model [10], we propose the joint model with the novel residual AE denoiser and the HGN decoder. At the transmitter, a K-bit information vector $u \in \mathbb{F}_2^K$ is first encoded into an N-bit codeword vector $v \in \mathbb{F}_2^N$. Using a modulation technique, the N-bit codeword v is mapped to a modulated symbol vector x. This symbol vector subsequently passes through a channel such as additive white Gaussian noise (AWGN) and Rayleigh fading channel. The received noisy symbol vector $y \in \mathbb{R}_2^N$ can be expressed as $y = h \cdot x + n$, where $n \in \mathbb{R}^N$ is a vector of Gaussian noise with zero mean and specific variance. $h > 0$ is the fading coefficient which follows a Rayleigh distribution with $E[h^2] = 1$.

At the receiver, y enters the denoiser. After the denoising process, the vector $\hat{x}^N \in \mathbb{R}^N$ is then sent to the HGN decoder. As in conventional NN decoders [15, 16], it performs soft decoding using the filtered vector. Finally, we can obtain an estimated information vector $\hat{u} \in \mathbb{F}_2^K$.

By denoting the arbitrary stochastic corrupting function of the channel as $\zeta : \mathbb{R}_2^N \to \mathbb{R}^N$ and denoising function as $\xi : \mathbb{R}^N \to \mathbb{R}^N$, we can formulate the denoising task as

$$\hat{x}^N = \xi\left(\zeta\left(x^N\right)\right). \tag{2}$$

For the denoising and decoding tasks, our aim is to minimize $L(x, \hat{x})$ and $L(u, \hat{u})$. We utilize mean squared error (MSE) as a loss function for both

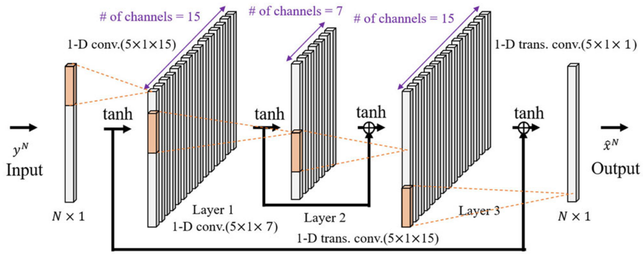

Fig. 2 *An illustration of the proposed residual AE denoiser*

Table 1. Total numbers of learnable parameters of four types of the joint model

	Decoder/Denoiser	
	BCH (63,51)	Polar (64,48)
HGN with AE-RD	38,141,104 / 1,202	64,247,296 / 1,202
HGN with CNN-RD	38,141,104 / 1,133	64,247,296 / 1,133
HGN with MLP-RD	38,141,104 / 1,441	64,247,296 / 1,461
HGN with LSTM-RD	38,141,104 / 4,046	64,247,296 / 4,174

tasks. By explicitly combining the denoising and the decoding loss, a total loss is given by:

$$L_{total} = w_1 L_{denoise} + w_2 L_{decode}. \tag{3}$$

The total loss value can be minimized through the multi-task learning technique.

We design a new denoiser based on DAE. It consists of an encoding and a decoding part. The encoding part contains doubly stacked 1D-convolutional layers [17] followed by hyperbolic tangent activation. The 1D-convolutional layers process the raw 1D input and learn to extract features in the sequence. Unlike 2D-CNN, 1D arrays replace 2D matrices, both kernels and feature maps. The decoding part contains doubly stacked 1D-transposed convolutional layers followed by hyperbolic tangent activation. Additionally, we embed nested residual skip connections between the encoding and the decoding part. These accelerate the training process as well as enhance feature extraction. It is also confirmed through experiments that double connections are more effective than a single connection in the proposed model. The detailed structure of the proposed denoiser is illustrated in Figure 2.

Experiments: We evaluate the performance of three RDs and the proposed denoiser. For convenience, we name the RDs with MLP, CNN, RNN, and AE as MLP-RD, CNN-RD, LSTM-RD, and AE-RD, respectively. We consider two code families: BCH and Polar codes. All generator matrices and parity check matrices are taken from [18].

The CNN-RD employs three 1D-convolutional layers as hidden layers. The feature maps of them are 24, 12, and 5, respectively. The size of kernels was set to 3 equally. In addition, 1D-maxpooling with kernel size 2 and stride 2 is utilized between the adjacent layers. The MLP-RD employs three fully connected layers of size 15, 8, and 4 as hidden layers. The LSTM-RD employs one LSTM unit with a hidden layer of dimension N. For all models, there is a residual connection between the input and the output layer. The activation function of each hidden layer is the hyperbolic tangent function, as in the proposed model. We set three RDs to have a similar number of learnable parameters as the proposed denoiser to avoid performance differences due to complexity. The total number of learnable parameters of all joint models is given in Table 1. Note that the number of learnable parameters of the HGN decoder is strongly influenced by the code length, while the denoiser is not.For the HGN decoder, we use a 10-layer NN structure for the same effect as 5 iterations in the BP decoder. The subnetwork f has 4 layers with 32 neurons at each layer while the subnetwork g has 2 layers with 16 neurons at each layer. These hyperparameters are set identically for a fair comparison. To simulate with valid codewords, training data are created by a generator matrix using all possible random information vectors. We set the training with varying SNRs ranging from 1 to 8 dB. Batch-based training is used with a maximum of 120 codeword samples per batch, with 15 codeword samples per SNR value. An epoch comprises 500 batches. We use Adam optimizer with a learning rate of 10^{-5} and a momentum value of 0.99. The evaluation was performed on random codewords, and simulation curves are obtained with 12,000,000 samples.

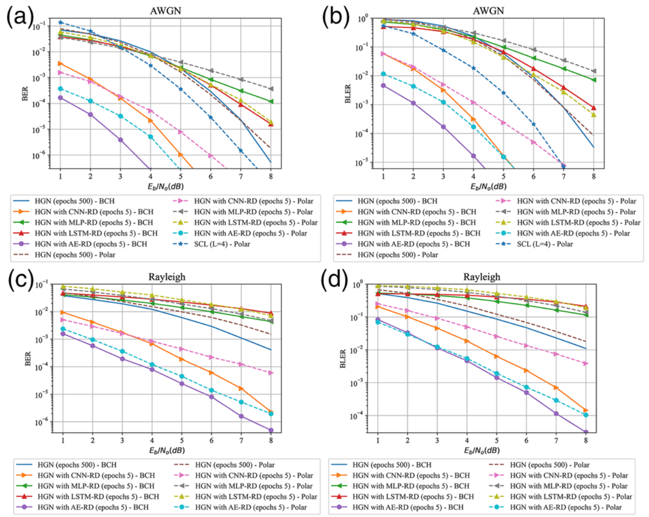

Fig. 3 *The BER and BLER performance curves over AWGN and Rayleigh fading channels for BCH (63,51) and Polar (64,48)*

Table 2. Ablation analysis on residual connections

	BCH (63,51)		Polar (64,48)	
	4	5	4	5
(i) Double residual connections	15.15	18.31	12.18	15.16
(ii) A Single residual connection	14.74	17.05	11.16	13.89
(iii) No residual connection	12.75	15.76	10.09	12.16

We present simulation results for BCH(63,51) and Polar(64,48) codes. Figure 3 shows the results of BER and BLER over AWGN and Rayleigh fading channels for E_b/N_o in dB, where E_b denotes the bit energy and N_o denotes the power spectral density of the noise. To evaluate our proposed denoiser objectively, we simulate three different NN denoisers, that is, the HGN decoder with MLP-RD, the HGN decoder with CNN-RD, and the HGN decoder with LSTM-RD. As shown in Figure 3a, for the AWGN, the proposed model has a coding gain of 4.7 and 3.5 dB higher than the HGN decoder at BER 10^{-5} for the BCH and polar codes, respectively. For the Rayleigh fading channel, Figure 3c shows that our model has an improvement of 5 and 6 dB to the HGN decoder at BER 10^{-3} for the BCH and polar codes, respectively. Note that the proposed joint model was trained with 5 epochs, whereas the individual HGN decoder was trained with 500 epochs.

For BLER performance, Figure 3b,d shows that our joint model has significant improvements to the HGN decoder for both codes over the AWGN and Rayleigh channels. It is also confirmed that the proposed model shows excellent performance in all SNR regions compared to the three joint models incorporating the existing RDs.

Ablation study on residual connections: We perform an ablation analysis on residual connections. We compare a method with (i) double residual connections, (ii) a single residual connection, and (iii) no residual connection. Table 2 presents the negative natural logarithm of BER with SNR 4 and 5 dB for the BCH (63,51) and the Polar (64,48) codes. It shows that the performance degrades as more connections are removed.

Impact of the kernel size: We investigate the impact of kernel size in the proposed model. Table 3 presents the negative natural logarithm of BER for the BCH (63,51) and the Polar (64,48) codes using three different kernel sizes: (i) 5 × 1, (ii) 3 × 1, and (iii) 1 × 1. Models (ii) and (iii) for the comparison were set to have similar amounts of learnable

Table 3. Performance comparison by kernel size

	BCH (63,51)		Polar (64,48)	
	4	5	4	5
(i) 5×1 convolution kernel	15.15	18.31	12.18	15.16
(ii) 3×1 convolution kernel	10.05	11.81	8.85	10.37
(iii) 1×1 convolution kernel	4.40	5.59	4.53	5.76

Table 4. Computational complexity and the number of operations

	Computational complexity	Operation count
AE-RD	$4NfC_1C_2$	132,300
CNN-RD	$f(N-4)(C_1C_2+C_2C_3+C_3C_4)$	104,135
MLP-RD	$(N+C_3)(C_2+C_4)$	1,349
LSTM-RD	$N^2+5NH+4H^2+3H$	4,284
HGN	$l_{max}N^3d_v^3$	357,329,070
SCL [19]	$LNlog_2N$	1,512

parameters to the proposed model (i) by adjusting the channel sizes. The kernel size with 5×1 achieves the best performance, so we set it for the proposed model.

Complexity analysis: We analyze the complexity of four types of denoisers, the HGN, and conventional polar successive cancellation list (SCL) decoder. Table 4 shows the computational complexity and the number of operations by substituting the values of each parameter into the expression. N is codeword length, f is kernel size, H is the number of hidden units, and C_k is the feature map of kth hidden layer for each denoisers. For the HGN, l_{max} is the number of iterations in the Trellis graph, and d_v is the average variable node degree. L is listing size of the SCL decoder. As a result, our proposed denoiser and the joint model have higher number of operations compared with the existing RDs and the SCL decoder, respectively. However, at the expense of increased complexity, our denoiser achieves significant gains over the benchmarks for both the HGN and SCL decoders as shown in Figure 3. Additionally, as the complexity of the proposed denoiser is much smaller than that of the HGN decoder, it can be further reduced by optimizing the architecture of the HGN decoder.

Conclusion: We proposed a new channel denoiser employing the residual DAE for the joint NN decoder and denoiser scheme. Simulation results confirmed that the proposed joint model outperforms the HGN decoder without the denoiser. In particular, for the BCH code, our model demonstrated an SNR gain of 4.7 dB at BER 10^{-5} and 5 dB at BER 10^{-3} with only 1% of epochs compared to the HGN decoder over AWGN and Rayleigh channels, respectively. For the polar code, our model not only showed the best performance compared to the HGN decoder but also showed a significant improvement compared to the SCL decoder. We demonstrated that the proposed denoiser outperforms all existing denoisers configured with a similar number of parameters.

Author contributions: Soyoung Han: Conceptualization, Methodology, Validation, Writing – original draft, Visualization. Junghyun Kim: Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. Hong-Yeop Song: Conceptualization, Methodology, Validation, Writing – original draft, Supervision.

Acknowledgement: This paper was financially supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C2011969).

Conflict of interest: The authors declare no conflict of interest.

Data availability statement: Data sharing is not applicable to this article as no datasets were generated or analysed during the current study

Received: *1 November 2022* Accepted: *12 December 2022*
doi: 10.1049/ell2.12711

References

1 Kang, J.M., Chun, C.J., Kim, I.M.: Deep-learning-based channel estimation for wireless energy transfer. *IEEE Commun. Lett.* **22**(11), 2310–2313 (2018)

2 Chun, C.J., Kang, J.M., Kim, I.M.: Deep learning-based channel estimation for massive mimo systems. *IEEE Wireless Commun. Lett.* **8**(4), 1228–1231 (2019)

3 Munaye, Y.Y., Lin, H.P., Adege, A.B., Tarekegn, G.B.: Uav positioning for throughput maximization using deep learning approaches. *Sensors* **19**(12), 2775 (2019)

4 Alrabeiah, M., Alkhateeb, A.: Deep learning for mmwave beam and blockage prediction using sub-6 ghz channels. *IEEE Trans. Commun.* **68**(9), 5504–5518 (2020)

5 Wen, C.K., Shih, W.T., Jin, S.: Deep learning for massive mimo csi feedback. *IEEE Wireless Commun. Lett.* **7**(5), 748–751 (2018)

6 Nachmani, E., Be'ery, Y., Burshtein, D.: Learning to decode linear codes using deep learning. In: 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 341–346. IEEE, Piscataway, NJ (2016)

7 Lugosch, L., Gross, W.J.: Neural offset min-sum decoding. In: 2017 IEEE International Symposium on Information Theory (ISIT), pp. 1361–1365. IEEE, Piscataway, NJ (2017)

8 Nachmani, E., Wolf, L.: Hyper-graph-network decoders for block codes. Advances in Neural Information Processing Systems **32** (2019)

9 Zhu, H., Cao, Z., Zhao, Y., Li, D.: A novel neural network denoiser for bch codes. In: 2020 IEEE/CIC International Conference on Communications in China (ICCC), pp. 272–276. IEEE, Piscataway, NJ (2020)

10 Zhu, H., Cao, Z., Zhao, Y., Li, D.: Learning to denoise and decode: A novel residual neural network decoder for polar codes. *IEEE Trans. Veh. Technol.* **69**(8), 8725–8738 (2020)

11 Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international Conference on Machine learning, pp. 1096–1103. ACM, New York (2008)

12 Luo, Q., Liu, Z., Chen, G., Ma, Y., Xiao, P.: A novel multi-task learning empowered codebook design for downlink scma networks. *IEEE Wireless Commun. Lett.* **11**(6), 1268–1272 (2022)

13 Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **47**(2), 498–519 (2001)

14 Ha, D., Dai, A., Le, Q.V.: Hypernetworks. arXiv preprint arXiv:160909106 (2016)

15 Nachmani, E., Marciano, E., Lugosch, L., Gross, W.J., Burshtein, D., Be'ery, Y.: Deep learning methods for improved decoding of linear codes. *IEEE J. Sel. Top. Signal Process.* **12**(1), 119–131 (2018)

16 Doan, N., Hashemi, S.A., Mondelli, M., Gross, W.J.: On the decoding of polar codes on permuted factor graphs. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE, Piscataway, NJ (2018)

17 Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1d convolutional neural networks and applications: A survey. *Mech. Syst. Sig. Process.* **151**, 107398 (2021)

18 Helmling, M., Scholl, S., Gensheimer, F., Dietz, T., Kraft, K., Ruzika, S., Wehn, N.: Database of Channel Codes and ML Simulation Results. (2019) www.uni-kl.de/channel-codes

19 Sybis, M., Wesolowski, K., Jayasinghe, K., Venkatasubramanian, V., Vukadinovic, V.: Channel coding for ultra-reliable low-latency communication in 5g systems. In: 2016 IEEE 84th vehicular technology conference (VTC-Fall), pp. 1–5. IEEE, Piscataway, NJ (2016)