Deep Joint Source-Channel Coding for Two-Way Relay Channels

Hyerin Uhm, Doyun Lee, and Hoon Lee, *Member, IEEE*

***Abstract*—This paper proposes a deep joint source-channel coding (DeepJSCC) scheme for two-way relay channels (TWRC) where two source nodes exchange image samples by means of a cooperative relay node. Unlike conventional DeepJSCC approaches developed for one-way relay channels (OWRC), the proposed DeepJSCC-TWRC facilitates self-interference cancellation at source nodes, thereby utilizing time resources more efficiently. We propose to share identical encoder and decoder models at source nodes so that they can observe a number of image samples during the training step. The effectiveness of the proposed method is demonstrated through numerical results.**

I. Introduction

Recently, deep learning-based wireless transceiver designs have got great attention owing to their capability of optimizing end-to-end communication systems [1]–[4]. In particular, deep joint source-channel coding (DeepJSCC) approaches have been regarded as promising solutions to resource-constrained wireless communication networks [1]–[3]. This framework leverages deep neural network (DNN) models that combine source and channel encoder/decoder modules. Such an integrated design successfully addresses the inefficiency of traditional separated source-channel coding schemes in a short block length regime. As a result, the data recovery performance can be fairly improved under strict resource constraints.

Albeit its potential, the DeepJSCC framework has been mostly confined to point-to-point communication systems where a source node conveys information to a destination node. However, in harsh environments, a direct communication link between source and destination nodes might not exist due to the long distance and obstacles. This issue has been recently tackled by [3] which develops the DeepJSCC method for one-way relay channels (OWRC). A cooperative relay node helps forward signals conveyed from a source to a destination, thereby establishing reliable communication links.

In practice, a destination node can also act as a source node since it also wishes to send data symbols. The OWRC is not suitable for realizing such a two-way communication as it requests a number of time resources to build two-way data transmissions. To this end, two-way relay channels (TWRC) were proposed where a relay node supports data exchange between two source nodes [5]–[7]. In the TWRC, two source nodes simultaneously transmit their signals to a relay node. Then, the relay broadcasts its received signal back to the source nodes. With self-interference cancellation (SIC), individual source nodes can perfectly recover the data symbols of others. Compared to the OWRC which requires four time slots for the data exchange, we only need two time slots by using the TWRC protocol.

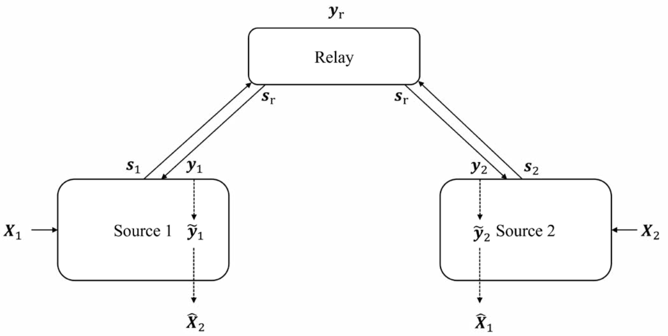

Fig. 1. System of Two way relay network.

This paper studies the deepJSCC-TWRC method which employs DNN-aided transceivers for source nodes in the TWRC. In particular, image transmission scenarios are considered where two source nodes wish to exchange their image samples through a cooperative relay. We propose to use channel attention (CA) techniques that allow encoder and decoder models to seamlessly adapt to varying propagation environments [2]. To further enhance the system performance, the parameter sharing approach is adopted where encoder and decoder models at two source nodes are realized with identical DNNs. As a result, the image recovery performance can be improved compared to conventional schemes that use dedicated encoder and decoder models. Simulation results verify the advantage of the proposed deepJSCC-TWRC over existing deepJSCC-OWRC [3]. It is confirmed that the proposed scheme achieves a better peak signal-to-noise ratio (PSNR) performance in all SNR regimes.

II. System Model

As shown in Fig. 1, we consider a cooperative network consisting of two source node S_1 and S_2 and a relay node R. Each S_i $(i = 1, 2)$ wishes to exchange its image data with the aid of the coordination of the relay. To this end, we adopt the two-way relay protocol [5]–[7]. First, in the multiple access channel (MAC) phase, the source nodes convey their encoded signals to the relay. Next, in the subsequent broadcast channel (BC) phase, the relay amplifies and forwards the received signal to the source nodes. Without loss of the generality, it is assumed that the total time resources are given as $2K$, which are evenly allocated to the MAC and BC phases, respectively.

H. Uhm, D. Lee, and H. Lee are with Ulsan National Institute of Science and Technology (UNIST), Ulsan, 44919, South Korea (e-mail: {hruhm,ehdbs8964,hoonlee}@unist.ac.kr).

979-8-3315-0694-0/25/$31.00 ©2025 IEEE

A. MAC Phase

Let $\mathbf{X}_i \in \mathbb{R}^{3\times H\times W}$ be an image sample transmitted by S_i, where 3 stands for the RGB colors and H and W respectively indicate the height and width. In the MAC phase, each S_i encodes its image $\mathbf{X}_i$ by using an encdoer DNN $f_{\theta_i} : \mathbb{R}^{3\times H\times W} \to \mathbb{C}^K$ with trainable parameter θ_i. Then, the corresponding encoding process can be written by

$$\mathbf{s}_i = f_{\theta_i}(\mathbf{X}_i, \mathbf{SNR}), \tag{1}$$

where $\mathbf{s}_i \in \mathbb{C}^K$ accounts for the complex transmit symbol and $\mathbf{SNR} \triangleq [\text{SNR}_{1r}, \text{SNR}_{2r}]^T$ denotes the SNR vector collecting SNR values SNR_{ir}, $\forall i$, between S_i and R, which will be defined shortly. The transmit power constraint at S_i is given by

$$\|\mathbf{s}_i\|^2 = P_i, \tag{2}$$

where P_i represents the transmit power budget at S_i. The source nodes send their encoded signals $\mathbf{s}_1$ and $\mathbf{s}_2$ to R over the same time-frequency resources. Thus, the received signal at R, denoted by $\mathbf{y}_r \in \mathbb{C}^K$, becomes

$$\mathbf{y}_r = \sqrt{\alpha_{1r}}\mathbf{s}_1 + \sqrt{\alpha_{2r}}\mathbf{s}_2 + \mathbf{n}_r, \tag{3}$$

where $\mathbf{n}_r \sim \mathcal{CN}(\mathbf{0}_K, \sigma^2\mathbf{I}_K) \in \mathbb{C}^K$ is the AWGN at the relay node, σ^2 is the noise power and α_{ir} is the channel gain between S_i and R. Then, the SNR value SNR_{ir} can be defined as

$$\text{SNR}_{ir} = \frac{\alpha_{ir}}{\sigma^2}. \tag{4}$$

B. BC Phase

The amplify-and-forward (AF) protocol is adopted at R. Assuming that $\mathbf{s}_1$ and $\mathbf{s}_2$ are independent, the power of the received signal $\mathbf{y}_r$ is given as

$$\mathbb{E}\|\mathbf{y}_r\|^2 = \alpha_{1r}P_1 + \alpha_{2r}P_2 + \sigma^2. \tag{5}$$

Therefore, the transmitted signal at the relay, denoted by $\mathbf{s}_r \in \mathbb{C}^K$, is derived as

$$\mathbf{s}_r = \sqrt{\frac{P_r}{\beta}}\mathbf{y}_r, \tag{6}$$

where P_r is the transmit power budget at the relay node and $\beta = \alpha_{1r}P_1 + \alpha_{2r}P_2 + \sigma^2$ indicates the power scaling factor.

The received signal at S_i becomes

$$\begin{aligned}\mathbf{y}_i &= \sqrt{\alpha_{ir}}\mathbf{s}_r + \mathbf{n}_i, &\text{(7a)}\\ &= \alpha_{ir}\sqrt{\frac{P_r}{\beta}}\mathbf{s}_i + \sqrt{\frac{P_r\alpha_{ir}}{\beta}}(\sqrt{\alpha_{\bar{i}r}}\mathbf{s}_{\bar{i}} + \mathbf{n}_r) + \mathbf{n}_i &\text{(7b)}\end{aligned}$$

where $\mathbf{n}_i \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ accounts for the Gaussian noise at S_i and $\bar{i} = 1$ if $i = 2$ and 2 otherwise. Provided that S_i knows the power scaling factor β and channel gains α_{ir}, it can remove the self-interference $\alpha_{ir}\sqrt{P_r/\beta}\mathbf{s}_i$ from the received signal $\mathbf{y}_i$ in (7a). After the self-interference cancellation, the received signal at S_i is obtained as

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i - \alpha_{ir}\sqrt{\frac{P_r}{\beta}}\mathbf{s}_i = \sqrt{\frac{P_r\alpha_{ir}}{\beta}}(\sqrt{\alpha_{\bar{i}r}}\mathbf{s}_{\bar{i}} + \mathbf{n}_r) + \mathbf{n}_i, \tag{8}$$

With $\tilde{\mathbf{y}}_i$ at hands, S_i retrieves the image $\mathbf{X}_{\bar{i}}$ by using its decoder DNN. Let $g_{\phi_i} : \mathbb{C}^K \to \mathbb{R}^{3\times H\times W}$ be the decoder DNN at source node S_i with parameter ϕ_i. Then, the resulting recovered image $\hat{\mathbf{X}}_{\bar{i}} \in \mathbb{R}^{3\times H\times W}$ is written by

$$\hat{\mathbf{X}}_{\bar{i}} = g_{\phi_i}(\tilde{\mathbf{y}}_i, \mathbf{SNR}). \tag{9}$$

In this paper, we aim to identify the optimal encoder and decoder DNNs f_{θ_i} and g_{ϕ_i}, $\forall i$, that can minimize the image reconstruction error. To this end, we consider the mean-squared-error (MSE) measure defined as

$$\text{MSE}(\Theta) = \mathbb{E}\left[\sum_{i=1}^{2}\|\mathbf{X}_i - \hat{\mathbf{X}}_i\|_F^2\right], \tag{10}$$

where $\Theta \triangleq \{\theta_1, \theta_2, \phi_1, \phi_2\}$ is the collection of trainable parameters and the expectation is taken over the distribution of images $\mathbf{X}_i$, $\forall i$, Gaussian noise vectors $\mathbf{n}_r$ and $\mathbf{n}_i$, $\forall i$, and SNR values SNR_{ir}, $\forall i$.

III. Proposed Method

In this section, we present the proposed DNN models for the encoder and decoder, which is followed by the proposed training algorithm.

A. Encoder

Fig. 2 illustrates the proposed encoder DNN architecture which comprises several residual blocks (ResBlock), channel attention (CA), ResBlock downsampling (ResBlock-Down), and power normalization at the output layer. The image $\mathbf{X}_i$ is first processed by ResBlock consisting of two convolutional layers with kernel size of 3×3, denoted by Conv 3×3, and the leaky rectified linear unit (LeakyReLU) activation. All convolutional layers in the ResBlock have C_{feat} output channles. We adopt the skip connection from the input of the ResBlock to its output. When the number of input channels are not the same with that of the output layer, we employ Conv 1×1 to the input feature.

After the second ResBlock, we adopt the CA module [2] which leverages the SNR vector $\mathbf{SNR}$ as side information. This module is crucial to achieve the generalization ability to arbitrary given channel gain α_{ir}, $\forall i$, in the test environment. The CA module first applies the global average pooling to the output of the previous ResBlock which yields average pixel values for individual input channels. Thus, the output dimension of the global average pooling equals C_{feat}. We append the SNR vector $\mathbf{SNR}$ to the output of the global average pooling layer, which is further processed by two subsequent linear layers with the LeakyReLU hidden activation. The output of the second linear layer is fixed to C_{feat}, which is followed by the sigmoid activation that produces attention values within $[0, 1]$. It is then multiplied to the output of the previous ResBlock across the channel dimension. By doing so, hidden features can capture SNR values between the source nodes and the relay.

The subsequent ResBlock-Down module aims to downsample the height and width of image samples. More precisely, it reduces the size of input features by the factor of 4. To this

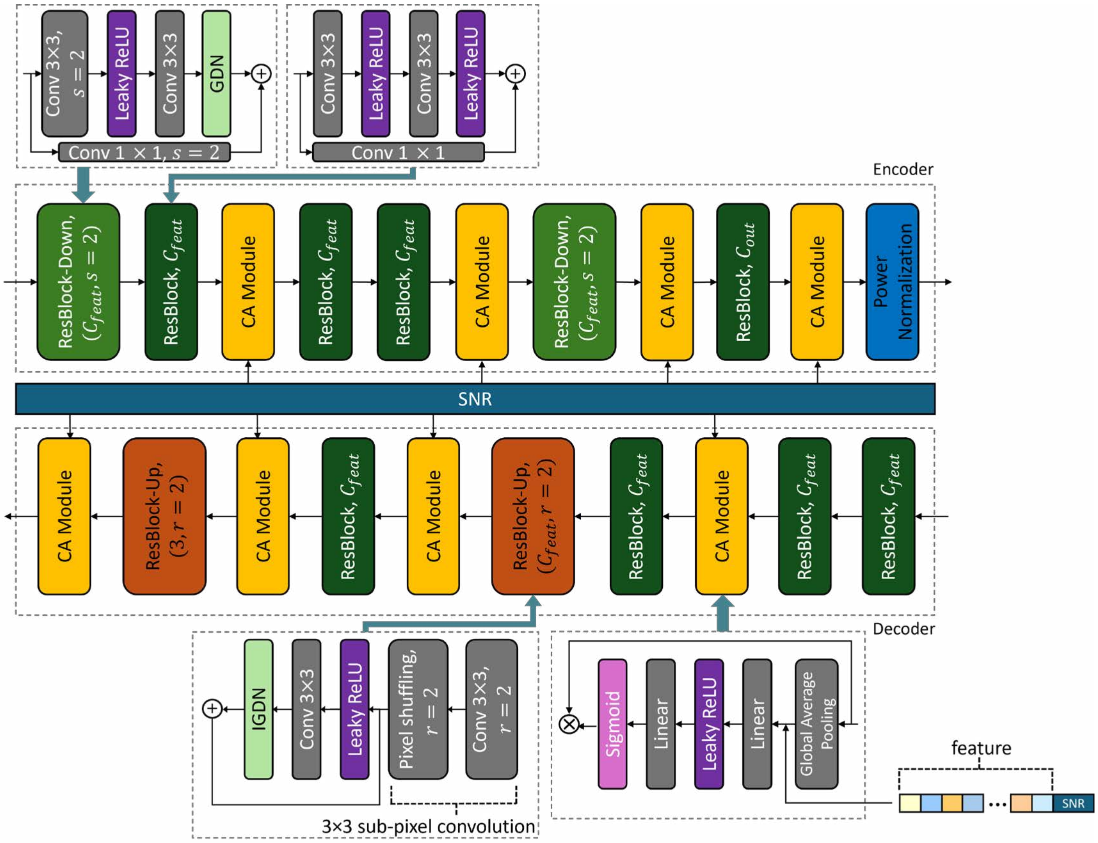

Fig. 2. Proposed DNN architecture.

end, we set the stride of Conv 3×3 at the first layer and Conv 1×1 in the skip connection to 2. The output of the second Conv 3×3 is post-processed by using the generalized divisive normalization (GDN), which has been known to be powerful in image compression tasks [8]. After the ResBlock-Down, we further employ the CA, ResBlock, and ResBlock-Down modules.

To accommodate the transmit power constraint at the source node (2), the output of the last CA module is handled by the power normalization layer. After the reshape operation and the complex representation, the input to the power normalization layer can be represented by the complex-valued vector $\tilde{\mathbf{s}}_i \in \mathbb{C}^K$. Then, the power normalization layer produces the transmitted signal vector $\mathbf{s}_i$ as

$$\mathbf{s}_i = \sqrt{P_i}\frac{\tilde{\mathbf{s}}_i}{\|\tilde{\mathbf{s}}_i\|}. \tag{11}$$

B. Decoder

Next, we explain the decoder DNN structure shown in Fig. 1. Compared to the encoder DNN in Fig. 2, ResBlock upsampling (ResBlock-Up) modules are newly introduced. This unit is dedicated to upsample the low-dimensional received signal $\tilde{\mathbf{y}}_i \in \mathbb{C}^K$ to the associated estimate $\hat{\mathbf{X}}_{\bar{i}} \in \mathbb{R}^{3\times H\times W}$.

To this end, the source node first reshapes the complex-valued signal vector $\tilde{\mathbf{y}}_i$ into a real-valued tensor of size $\frac{32K}{HW} \times \frac{H}{4} \times \frac{W}{4}$. It is then processed by using several ResBlock and CA modules. We upsample the output of the third ResBlock of size $C_{\text{feat}} \times \frac{H}{4} \times \frac{W}{4}$ by using the ResBlock-Up module. More precisely, 3×3 sub-pixel convolution with the upscale factor $r = 2$ rearranges the input tensor to a tensor of size $C_{\text{feat}} \times \frac{H}{2} \times \frac{W}{2}$ [9]. It is then followed by Conv 3×3 with C_{feat} output channels and the inverse GDN (IGDN) layer. As a result, the output size of the ResBlock-Up becomes $C_{\text{feat}} \times \frac{H}{2} \times \frac{W}{2}$.

Subsequently, we apply the CA module and ResBlock, whose output size is given as $C_{\text{feat}} \times \frac{H}{2} \times \frac{W}{2}$. To retrieve the estimated image $\hat{\mathbf{X}}_{\bar{i}} \in \mathbb{R}^{3\times H\times W}$, the ResBlock-Up with $r = 2$ is employed where the number of output channels of all convolutional layers is set to 3. Finally, the last CA module produces $\hat{\mathbf{X}}_{\bar{i}}$.

C. Training and implementation

We provide the training strategy of the proposed DeepJSCC-TWRC. The mini-batch stochastic gradient descent (SGD) algorithm, e.g., the Adam optimizer, is employed to train the encoder DNNs and decoder DNNs jointly. The associated parameter update rule at each training epoch is written as

$$\Theta \leftarrow \Theta - \eta\nabla_\Theta \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{2}\|\mathbf{X}_i^{(b)} - \hat{\mathbf{X}}_i^{(b)}\|_F^2, \tag{12}$$

where $\eta > 0$ stands for the learning rate, ∇_Θ represents the gradient operator with respect to Θ, B is the batch size, and $\mathbf{X}_i^{(b)}$ and $\hat{\mathbf{X}}_i^{(b)}$ respectively denote the b-th image in the mini-batch set transmitted at S_i and its recovery at $S_{\bar{i}}$. The

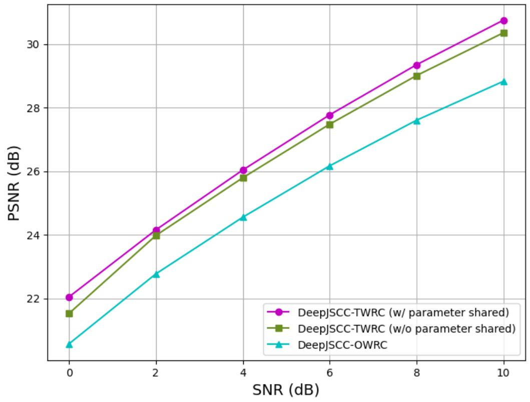

Fig. 3. Average PSNR performance of various schemes with respect to test SNR.

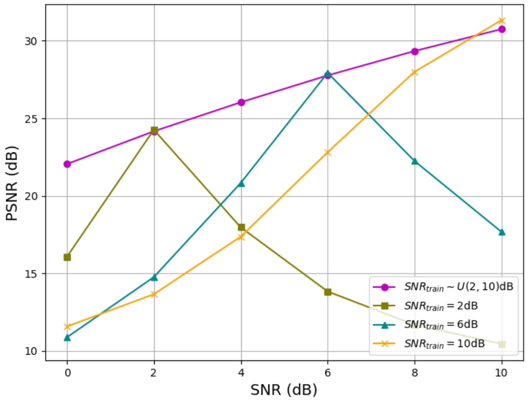

Fig. 4. Average PSNR performance of proposed DeepJSCC-TWRC with respect to test SNR.

SGD update (12) is repeated until the predetermined maximum number of epochs.

In the training procedure, we adopt the random SNR sampling where the SNR values SNR_{ir}, $\forall i$, are uniformly generated within a range, i.e., $\text{SNR}_{ir} \sim U(\text{SNR}_{\text{min}}, \text{SNR}_{\text{max}})$, where SNR_{min} and SNR_{max} account for the minimum and maximum SNR values, respectively. Once trained, the optimized encoder DNNs and decoder DNNs are installed on associated source nodes to carry out real-time image exchange applications. In this test step, the proposed DeepJSCC-TWRC can be applied to an arbitrary given test SNR.

IV. Simulation Results

In this section, we assess the proposed DeepJSCC-TWRC for CIFAR-10 dataset containing $60,000$ images of size $3 \times 32 \times 32$. We use $50,000$ and $10,000$ images for the training and test, respectively. The proposed DeepJSCC-TWRC is trained by using the Adam algorithm with learning rate $\eta = 10^{-4}$ and batch size $B = 32$. For simplicity, the SNR values of two communication links are set to be the same as $\text{SNR} = \text{SNR}_{1r} = \text{SNR}_{2r}$. In the training, the SNR is uniformly sampled within $[2, 10]$ dB, whereas it is fixed at a certain vavlue for the test.

We set the channel per pixel (CPP), which is defined as $\frac{K}{3HW} = \frac{K}{3072}$, to 0.25, thereby resulting in total $2K = 1536$ time resources in the MAC and BC phases. To this end, the convolution layers are realized with $C_{\text{feat}} = 256$ output channels. For a fair comparison, the total number of time resources of the DeepJSCC-OWRC [3] is set to $4K = 1536$, which leads to the CPP of 0.125. Therefore, the compression rate of the conventional DeepJSCC-OWRC becomes half of that of the proposed DeepJSCC-TWRC, which poses significant reconstruction errors.

To evaluate the image reconstruction quality, we leverage the peak signal-to-noise ratio (PSNR) between $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$ defined as

$$\text{PSNR}(\mathbf{X}_i, \hat{\mathbf{X}}_i) = 10 \log_{10} \left(\frac{(\max_{m,n} [\mathbf{X}_i]_{mn})^2}{||\mathbf{X}_i - \hat{\mathbf{X}}_i||_F^2} \right), \quad (13)$$

where $[\mathbf{U}]_{mn}$ stands for the (m, n)-th element of a matrix $\mathbf{U}$.

Fig. 3 depicts the average PSNR performance of various schemes with respect to the SNR. The performance of the proposed DeepJSCC-TWRC is plotted for two different cases, i.e., with parameter sharing and without parameter sharing. When the parameter sharing policy is adopted, the source nodes reuse the identical encoder and decoder DNNs as $f_\theta = f_{\theta_1} = f_{\theta_2}$ and $g_\phi = g_{\phi_1} = g_{\phi_2}$. Such a technique is beneficial for enhancing the generalization ability of the DNN-based transceivers [4]. Our results confirm that the proposed parameter sharing approach exhibits a performance gain over the method without the parameter sharing. In addition, it can be seen that the proposed DeepJSCC-TWRC outperforms the conventional DeepJSCC-OWRC. Such a gain is achieved due to the increase in the CPP and self-interference cancellation.

To see the impact of the random SNR sampling in the training step, Fig. 4 presents the average PSNR performance of the proposed DeepJSCC-TWRC with various training SNR values, denoted by $\text{SNR}_{\text{train}}$. The PSNR performance of the random SNR sampling approach generally performs better than that of the fixed training SNR setups. The DeepJSCC-TWRC with fixed training SNR works well only at the identical test SNR, whereas its PSNR performance degrades at other test SNR values. In contrast, the proposed random SNR sampling provides a good average PSNR performance over all simulated test SNR values. This demonstrates the generalization ability of the proposed scheme for arbitrary given channel quality in the test environment.

V. Conclusions

In this paper, we have proposed the DeepJSCC approach for the TWRC where the image data exchanges between two source nodes are assisted by a relay node. The source nodes employ DNN-based encoders and decoders to compress and retrieve image samples, respectively. The CA module has been adopted to train efficient encoders and decoders robust to arbitrary SNR ranges. Also, we have proposed to reuse identical DNN models for two source nodes so that they can share image samples during the training procedure. Numerical results have demonstrated the superiority of the proposed DeepJSCC-TWRC over conventional OWRC counterparts. As a future research direction, it would be interesting to realize the relay operation using distinct DNNs.

Acknowledgement

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grants by MSIT (Intelligent 6G Wireless Access System) under Grant 2021-0-00467 and in part by the Korea Research Institute for Defense Technology Planning and Advancement (KRIT) grant funded by the Korea government (DAPA(Defense Acquisition Program Administration)) (21-106-A00-007, Space-Layer Intelligent Communication Network Laboratory, 2022).

References

[1] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, “Deep joint source-channel coding for wireless image transmission,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.

[2] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, “Wireless image transmission using deep source channel coding with attention modules,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2315–2328, Apr. 2022.

[3] C. Bian, Y. Shao, H. Wu, and D. Gündüz, “Deep joint source-channel coding over the relay channel,” in *Proc. IEEE Int. Conf. Mach. Learn. Commun. Netw. (ICMLCN)*, May 2024, pp. 1–6.

[4] H. Lee and S.-W. Kim, “Task-oriented edge networks: Decentralized learning over wireless fronthaul,” *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15 540–15 556, May 2024.

[5] R. Zhang, Y.-C. Liang, C. C. Chai, and S. Cui, “Optimal beamforming for two-way multi-antenna relay channel with analogue network coding,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 699–712, Jun. 2009.

[6] K.-J. Lee, H. Sung, E. Park, and I. Lee, “Joint optimization for one and two-way MIMO AF multiple-relay systems,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 12, pp. 3671–3681, Dec. 2010.

[7] H. Park, C. Song, H. Lee, and I. Lee, “MMSE-based filter designs for cognitive multiuser two-way relay networks,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1638–1643, Apr. 2015.

[8] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[9] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1874–1883.