Received 6 March 2025; revised 13 April 2025; accepted 31 May 2025. Date of publication 4 June 2025; date of current version 23 June 2025.

Digital Object Identifier 10.1109/OJCOMS.2025.3576502

5G Indoor/Outdoor Field Trial of Deep Joint Source-Channel Coding Method

DAISUKE HISANO[1] (Member, IEEE), KEIGO MATSUMOTO[1], YOSHIAKI INOUE[1] (Member, IEEE), YUKO HARA[2] (Member, IEEE), KAZUKI MARUTA[3] (Senior Member, IEEE), YU NAKAYAMA[4] (Member, IEEE), HIROSHI TATSUKAWA[5], YUJI KAWAI[5], YOSHINORI SHINOHARA[5], AND HIROKI IKEDA[5]

[1]Graduate School of Engineering, Osaka University, Suita 565-0871, Japan
[2]School of Engineering, Institute of Science Tokyo, Meguro 152-8550, Japan
[3]Faculty of Engineering, Tokyo University of Science, Katsushika 125-8585, Japan
[4]Institute of Engineering, Tokyo University of Agriculture and Technology, Koganei 184-8588, Japan
[5]Technology Development Department, Magna Wireless Corporation, Hachioji 192-0072, Japan

CORRESPONDING AUTHOR: D. HISANO (e-mail: hisano@ieee.org)

This work was supported in part by the "Research and Development Project of the Enhanced Infrastructures for Post-5G Information and Communication Systems", Commissioned by the New Energy and Industrial Technology Development Organization (NEDO) under Grant JPNP20017; in part by the JST, ACT-X, Japan, under Grant JPMJAX24MA; and in part by the Okawa Foundation, Japan, under Grant 23-04.

ABSTRACT This paper presents the first outdoor field trials of deep joint source-channel coding (DeepJSCC) for image transmission over a 5G system. DeepJSCC is a deep learning-based end-to-end method that unifies source and channel coding to enable robust and low-latency image transmission, particularly in low signal-to-noise power ratio (SNR) environments. Unlike conventional methods, DeepJSCC inherently avoids the cliff effect and maintains stable image quality under harsh channel conditions. To evaluate its feasibility of practical environments, we modified a commercial 5G base station (gNB) and user equipment (UE) to support DeepJSCC signal transmission and reception. Extensive experiments were conducted under indoor and outdoor settings, including line-of-sight (LoS) and non-line-of-sight (NLoS) conditions. A key contribution of this study is the empirical verification that even a DeepJSCC model trained solely on an additive white Gaussian noise (AWGN) channel can maintain stable reconstruction performance in real 5G environments, demonstrating its generalization capability. Compared to baseline systems using JPEG2000 and LDPC, DeepJSCC achieved higher PSNR stability and was able to restore image content even when conventional schemes completely failed. These findings suggest that DeepJSCC is a promising candidate for next-generation visual communication over 5G infrastructure.

INDEX TERMS 5G mobile communication, data compression, deep learning, edge computing, image communication.

I. INTRODUCTION

FIFTH-GENERATION cellular systems (5G) are widely used. In recent years, Beyond 5G and 6G, which are extensions of 5G, have been investigated. The targets of the research are not only for mass users but also for industrial networks [1], [2], [3]. Japan has considered industrial 5G systems, termed local 5G to encourage the digital transformation of mission-critical services, such as smart factories and agriculture, robot control, and railways. Industrial 5G aims to forward extreme large data from surveillance cameras inside and outside factories, camera sensors used for inspection devices, video streaming at event sites, and disaster monitoring. The industrial 5G system needs to be a leading candidate for high-speed telecommunications, and it is also complied the typical 5G specification such as 3GPP, whereas it may have a special characteristic that a lot of resource blocks (RBs) are occupied by a part of users who connect continuously to the 5G system to transmit or receive video/image data. In other words, a mobile operator must determine one policy to operate

the 5G system: the system limits the number of connected users to ensure higher network quality, or the system degrades the network quality to increase the number of users.

At the same period of the development of 5G system, the deep learning technologies have been extremely and rapidly studied. In recent years around 2020s, semantic communication (SC), also known as task-oriented communication, is emerging as a concept that exceeds the limitations of bit-based communication [4], [5]. In the conventional telecommunication network, the physical layer functions only have to deal with bit information fairly because the priority and quality are set for each application, at the media access control (MAC) layer, transmission control protocol (TCP) / Internet protocol (IP) layer and higher ones. Thus, the quality of the PHY layer is determined by simply using bit error rate (BER) in the almost cases. Meanwhile, the SC systems adaptively change the types of encoding and modulation formats based on the application, for example, video, images and audio. Additionally, the requirements of each application for the BER and latency differ. As one of SC systems, deep learning-based joint source-channel coding (DeepJSCC) methods have been widely studied and it is comparable to Shannon's capacity than two-stage encoding, source coding such as JPEG2000 and BPG, and channel coding such as LDPC [6], [7]. In addition, DeepJSCC avoids the cliff effect, which rapidly degrades the image quality when the signal-to-noise power ratio (SNR) falls below a certain level, and it is used in the lower SNR region well. For these reasons, we expect high-quality image transmission using DeepJSCC over industrial 5G systems. DeepJSCC is particularly beneficial for real-time image transmission applications such as remote healthcare, autonomous driving, and smart city surveillance. In remote healthcare, DeepJSCC enables low-latency and robust transmission of medical images, where traditional compression methods such as JPEG2000 often fail under wireless conditions due to error propagation. Similarly, in self-driving car networks, DeepJSCC can improve real-time image transmission reliability in varying SNR conditions. In the network system for surveillance cameras, DeepJSCC provides an advantage in transmitting semantically important features at lower bitrates while maintaining object recognition accuracy. However, despite the many advantages of the cooperation of DeepJSCC and 5G, to the best of our knowledge, the Deep JSCC operation has not been sufficiently reported under the actual 5G condition.

This paper presents a proof-of-concept (PoC) of DeepJSCC in an industrial 5G system. We modify a commercially available 5G base station (gNB) and its user equipment (UE) to input and output DeepJSCC signals and conduct a radio transmission experiment in outdoor environment. This study extends the analysis in [8]. The previous work demonstrated a wired cable connection, while this article includes the results of wireless transmission. The experimental results imply that the DeepJSCC has the potential to perform well under wireless conditions. The contributions of this paper are as follows:

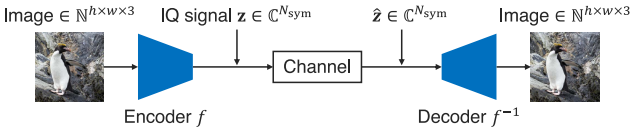


FIGURE 1. Encoder-Decoder model for DeepJSCC.

- We present a simple DeepJSCC architecture consisting of convolutional layers, GDN, and PReLU, which achieves competitive PSNR performance compared to conventional source-channel separation methods. Although simulation results show limited advantage in high-SNR regimes, we identify that DeepJSCC exhibits remarkable robustness in low-SNR conditions based on field trials.
- We conduct the 5G field experiments of DeepJSCC in both indoor and outdoor environments. We describe the system integration process, including modifications to the 5G gNB and UE, and clarify how DeepJSCC signals were directly injected into physical resource blocks.
- We verify that DeepJSCC can operate on actual 5G infrastructure using commercial gNB and UE devices, leveraging native functions such as channel estimation. The system was validated in indoor, outdoor LoS, and NLoS scenarios with real data transmission.
- We empirically demonstrate that a DeepJSCC model trained solely on an AWGN channel remains effective in practical 5G environments without retraining or channel-specific adaptation.

The subsequent sections are structured as follows. Section II describes the related work for DeepJSCC. Section III give a detailed description of the interface design of the modified 5G system. As preliminary verification, the simulation results are shown in Section IV. Sections V and VI reveal the indoor/outdoor experimental results, respectively. According to the results, we indicate the limitation in Section VII. Section VIII is the conclusion.

II. RELATED WORK

A. DEEP JOINT SOURCE-CHANNEL CODING (DEEPJSCC)

DeepJSCC is the neural network (NN) of a typical encoder-decoder configuration. Figure 1 shows the basic configuration of DeepJSCC. DeepJSCC encoder converts an input image into latent vector and the decoder restores the image. That is, DeepJSCC is one of latent representation learning techniques based on autoencoder. The difference compared with the conventional learning, is the image is directly mapped on IQ baseband symbols. Thus, the autoencoder plays the role of source and channel coding and modulation. DeepJSCC, which was first proposed, comprises a series of convolutional layers to achieve wireless image transmission [6], [7]. In recent years, several types of DeepJSCC have been proposed to extend from [6], [7] to

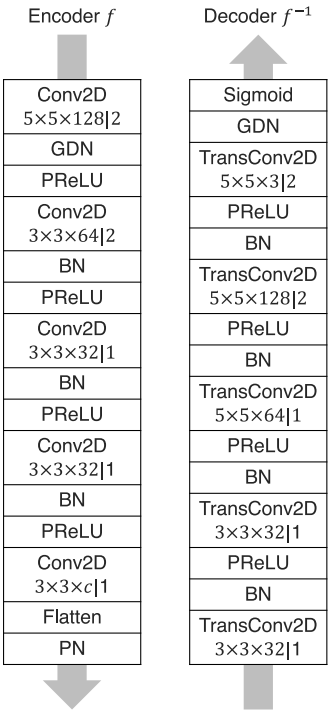


FIGURE 2. DeepJSCC encoder/decoder structure. Conv2D: two-dimensional convolutional layer, GDN: generalized divisive normalization, BN: batch normalization, PReLU: parametric rectified linear unit, PN: power normalization, TransConv2D: two-dimensional transposed convolutional layer.

video transmission [9], point cloud [10] and DeepJSCC with encryption [11]. It is expected that DeepJSCC will continue to be proposed to be integrated with various deep learning applications.

Here, we introduce the fundamental layer structure and basic concept of DeepJSCC for image transmission. Figure 2 shows the layered structure. The structure was adopted from [6], [7] and we slightly modified it accordingly for stabilization. The encoder of DeepJSCC contains a series of two-dimensional convolutional layers (Conv2D), generalized divisive normalization (GDN) or batch normalization layers (BN), and parametric rectified linear unit (PReLU) layers as activation layers. The notation $H \times W \times F|S$ denotes the kernel size with height H, width W, number of output channels F, and stride size S. The encoder is represented as a function f:$\{0, 1, \ldots, 255\}^{h_{\text{in}} \times w_{\text{in}} \times 3} \rightarrow \mathbb{C}^{N_{\text{sym}}}$ where h_{in} and w_{in} are the height and width of the input image. N_{sym} is the number of IQ symbols as bellows,

$$N_{\text{sym}} = \frac{h_{\text{in}} w_{\text{in}} c}{2 \prod s_i^2}, \tag{1}$$

where c denotes the number of output channels in the last Conv2D. The compression rate is discretely adjusted using c. s_i denotes the stride size at the i-th Conv2D layer. N_{sym} contains a coefficient $1/2$. This is because the encoder output is divided into two parts and combined to derive IQ complex symbols. The first and latter parts correspond to the In-phase- and Quadrature- symbol sequences, respectively. Since the compression ratio is adjusted at the final layer of the encoder and the first layer of the decoder, it does not significantly increase the number of parameters in the DeepJSCC system. For example, when comparing $c = 8$ and $c = 16$, the total number of parameters in the encoder increases by only about 2%. Therefore, it does not have a significant impact on processing latency and power consumption.

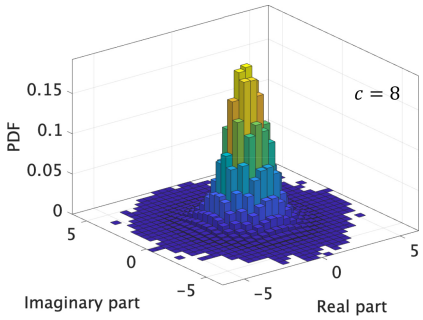


FIGURE 3. Constellation distribution of DeepJSCC signal in the case of CIFAR-10 test dataset.

Upon dividing them into IQ sections, we combine the I-symbol with the Q-symbol and create a complex baseband signal. The output symbols $\mathbf{z} \in \mathbb{C}^{N_{\text{sym}}}$ are normalized based on the average power $\mathbf{z}/\|\mathbf{z}\|$ and transmitted to the wireless channel. The decoder contains a series of two-dimensional transposed convolutional (TransConv2D) layers, BN or inverse GDN, and PReLU layers. Figure 3 shows the distribution of the encoder outputs as IQ symbols when trained on the CIFAR-10 dataset. A Gaussian-like distribution was obtained. In an AWGN channel with an average power constraint, the ideal IQ symbol distribution is Gaussian, and it can be observed that the encoder outputs a comparable distribution. This indicates that DeepJSCC automatically selects symbol mappings that asymptotically approach the Shannon limit.

For deep learning techniques, this paper employed a simple NN based on convolutional structure. However, optimization should be performed depending on the specific task. In such cases, neural architecture search (NAS) is significant effective [12]. The related work on split computing has proposed the use of NAS to optimize latency, which can also be applied to DeepJSCC [13]. Furthermore, DeepJSCC can be applied not only for image transmission but also for image recognition [14] as semantic communication. In this case, similar to split computing, the layers of an NN for image recognition or object detection can be divided, and the latent space can be transmitted as IQ signals. Moreover, in 5G, channel estimation is performed using pilot signals. Meanwhile, in scenarios where the channel dynamically fluctuates, coordination between an adaptive equalizer and a decoder of DeepJSCC is also necessary. Although we have proposed the coordination method mainly for underwater acoustic communication [15], similar approaches could be considered for DeepJSCC in the future, e.g., 6G.

B. DEEPJSCC'S IMPLEMENTATION

Unlike typical deep learning applications, processing time is also an important factor and discussed in [6], which suggests that the execution speed is comparable or faster to that of JPEG and other formats. In addition, real-time performance is an important indicator for use in video transmission. We have proposed a self-attention-based DeepJSCC with FPGA implementation, the encoder has successfully operated at more than 30 FPS [16]. The processing time is the important parameter to determine the layer structure. Owing to a relatively uncomplicated layer structure, DeepJSCC is operated in the required processing time.

More cellular communication-specific schemes have also been widely studied, such as integrating equalization and DeepJSCC decoders when orthogonal frequency division multiplexing(OFDM) schemes are used [17], adaptively tolerating channel gain and noise variations based on channel state information (CSI) [18], and achieving one-to-many communications [19]. Several studies on simulation analyses are also reported. For employing DeepJSCC on cellular system, DeepJSCC does not use bit-level information, making it less compatible with upper layers that are controlled based on bit-information data. In particular, retransmission functions, which are closely related to the physical layer, cannot be directly utilized. Thus, integrating DeepJSCC with hybrid automatic repeat request (HARQ) has been proposed [20].

For the experiment, to the best of our knowledge, only software-defined radio-based indoor experiments have been reported [21], [22]. In [21], the carrier frequency and system bandwidth were 2.4 GHz, 300 kHz. In [22], 2 GHz and 364 kHz were adopted as well. These are experimental results in a limited environment with a narrow bandwidth.

We studied DeepJSCC for cellular applications, particularly 5G systems. In this paper, an interface that connects external devices to a commercially supported 5G system (gNB and UE) is implemented. The external device directly transmits the IQ symbols to the gNB. The device then assigns RBs to OFDM signals. When receiving IQ symbols, the gNB assigns proper RBs based on indications from the external device. We conducted experiments using a commercially supported 5G gNB and UE to investigate the feasibility of image transmission.

III. INTERFACE DESIGN

We modified a commercially available industrial 5G system released in Japan to transmit and receive DeepJSCC signals. The 5G system specifications are listed in Table 1. In Japan, the frequency band ranges from 4.6 to 4.9 GHz. One frame (= 10 ms) contains ten sub-frames. One sub-frame (= 1 ms) includes two slots owing to the 30-kHz subcarrier spacing. One slot is divided into 14 OFDM symbols (= 35.7 μs). Finally, one resource block (RB) contains 12 subcarriers and one slot. Considering the frequency guard band, there are 273 RBs per slot as a physical downlink shared channel (PDSCH).

TABLE 1. 5G system parameters.

Parameter	Value
Frame format	Semi-synchronous
Center frequency	4849.98 MHz
System bandwidth	100 MHz
Subcarrier spacing	30 kHz
Synchronization signal block	20 ms period
Dummy PDSCH	MCS: 9

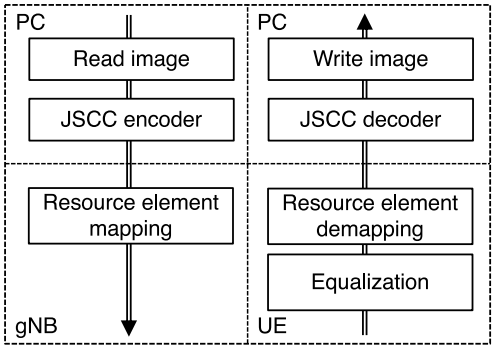


FIGURE 4. Physical layer structure. The base station was modified to receive/transmit IQ data from/to an external device.

Figure 4 shows the modified structure of the 5G system. In this implementation, IQ symbols generated by an external device are directly written into the RBs of the gNB. Consequently, higher-layer PHY functionalities, such as HARQ, are bypassed and remain inactive for these externally provided signals. It should be noted that this configuration is not designed to maintain full compatibility with the standard 5G protocol stack, but is instead a purpose-built modification intended for PoC experimentation.

The gNB and UE transmit and receive IQ symbols from external devices. The external device encapsulates IQ symbols in Ethernet frames and forwards them to gNB. The gNB maps the IQ data to RBs and transmits them to the UE. The UE receives the signal. The received signal is not directly fed into the DeepJSCC decoder but is first passed through the equalizer implemented in the UE. This equalizer is based on a conventional zero forcing method using channel state information. After that, the UE extracts the IQ symbols from RBs. Subsequently, the IQ symbols were encapsulated into Ethernet frames and forwarded to an external device. This operation was repeated every 10 ms. Although other systems can use the unused RBs, a dummy QPSK signal is inserted in this modification. Figure 5 shows the RB assignments. We allocated DeepJSCC signals from the 0th to 90th RBs. Additionally, the 0th and 10th slots are used. Other RBs can share with conventional 5G systems, as conventional IQ signals can be allocated arbitrarily. However, additional studies will be needed in the future regarding layers higher than the PHY layer. The number of available OFDM symbols PDSCH is 12, whereas one slot includes 14 OFDM symbols. Thus, the maximum number of transmittable IQ symbols per slot is 91 (RB/PDSCH) $\times$12 (subcarrier/RB) $\times$12 (PDSCH)

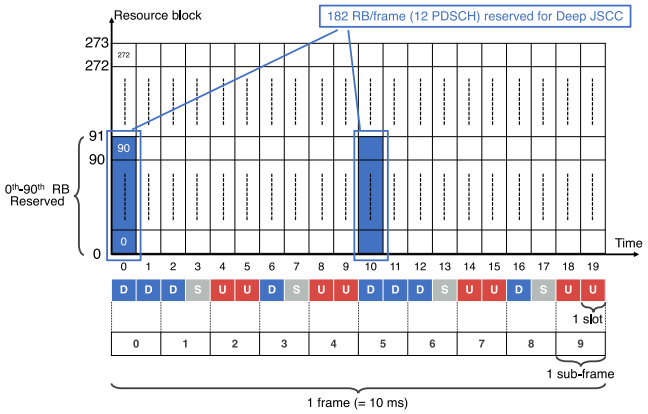


FIGURE 5. Modified 5G frame structure for DeepJSCC. D: downlink, U: uplink, S: special slot. 0-to-90 RBs and 0th & 10th slots are assigned for DeepJSCC.

= 13104 (IQ symbol). The DeepJSCC signal forwarded from the external device to gNB is a quantized 16-bit signal. The throughput required for the interface was 16 (bit) ×2 (IQ) ×13104 (IQ symbol) /10 (ms) ≃ 5.24 Mbps. In other words, a standard Ethernet cable can be used to transfer signals.

IV. PRELIMINARY VERIFICATION

This section describes the layer structure and learning method we used prior to conducting experiments. Here, we compare the DeepJSCC with the normal QAM signal with JPEG2000 and LDPC by simulation.

A. DEEPJSCC SETUP

Figure 2 shows the layer structure. The dataset we used was DIV2K which includes 800 and 100 full HD images for training and validation [23]. In the training process, we cropped four squared images from one full HD images and resized it to 256 × 256 pixel. Finally, we obtained 3200 images for training. At first, the DeepJSCC was trained 1000 epochs in CIFAR-10. After that, we trained the DeepJSCC with 5000 epochs in the cropped DIV2K of 3200 images. In anticipation of field experiments under real outdoor conditions, we trained the model using a simple AWGN channel model to avoid overfitting to specific channel characteristics and to prioritize generalization and robustness across diverse environments. The training signal-to-noise power ratio (SNR) was set to 6 dB or 13 dB. In addition, the experiment in the next section also verify training SNR = 20 dB. As to the reason for supporting these three types of the training SNR, in our previous work we confirmed the relationship between training SNRs and test SNRs on a simulation basis [24]. As a result, we have confirmed that a small number of training SNRs are sufficient to support a wide-range test SNR, although we have only investigated the AWGN channel. Therefore, we employ three types of training SNRs in this paper. In the validation process, we cropped the 100 validation images to square maximum size and then resized to 256 × 256. Loss function and optimizer were set to mean square error (MSE) and Adamax. c, which is the number of output channels at the last layer of the encoder in Fig. 2, adjusts the compression ratio. Thus, in the validation process, the number of IQ symbols is $256 \times 256c/(4 \times 4 \times 2) = 2048c$. In this paper, we set $c = 4, 8, 16$.

As a baseline method, we employed JPEG and JPEG2000 on the source coding and the LDPC on the channel coding, respectively. The LDPC coding rate was set to 1/2 or 2/3. Note that we assumed the block length of LDPC to DVB-S.2 not typical 5G. This is because the DVB-S.2 has the higher error tolerance than the typical 5G. That is, this conditions are slightly unfavorable for DeepJSCC. The block length was 64800. In the DeepJSCC transmission, the number of IQ symbols is fixed by c. Thus, to conduct fair evaluation, while we changed the compression index $\gamma_{\rm jp}$ of JPEG and JPEG2000, we sought the number of IQ symbols by each image close to that of DeepJSCC IQ symbols as below,

$$\gamma_{\rm jp}^{*} = \underset{\gamma_{\rm jp}}{\rm argmin} \left\| \frac{N_{\rm jp}(\gamma_{\rm jp})}{2kr} - N_{\rm sym} \right\|, \tag{2}$$

where k is the modulation index, r is the coding rate. $N_{\rm jp}(\gamma)$ is the number of bits after source coding with compression index $\gamma_{\rm jp}$.

B. SIMULATION – PSNR EVALUATION –

We conducted the numerical simulation using the DeepJSCC model learned by the setup in Section IV-A. The channel was set as AWGN and the PSNR was evaluated when the SNR was varied. PSNR is introduced as the image quality metric and is defined as

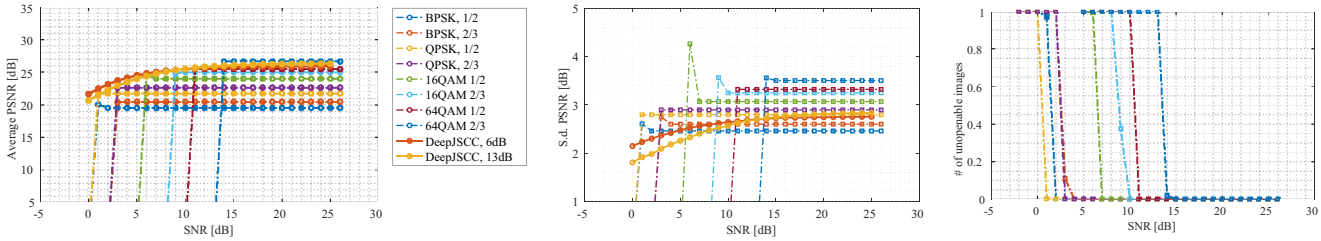
$$\text{PSNR} = 10 \log_{10} \frac{R_{\rm max}^2}{e_{\rm mse}}, \tag{3}$$

where $R_{\rm max}$ is the maximum pixel value. In the common case, $R_{\rm max} = 255$. $e_{\rm mse}$ is the MSE between the input and output images. It is defined as,

$$e_{\rm mse} = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \left[x(i,j) - x'(i,j) \right]^2. \tag{4}$$

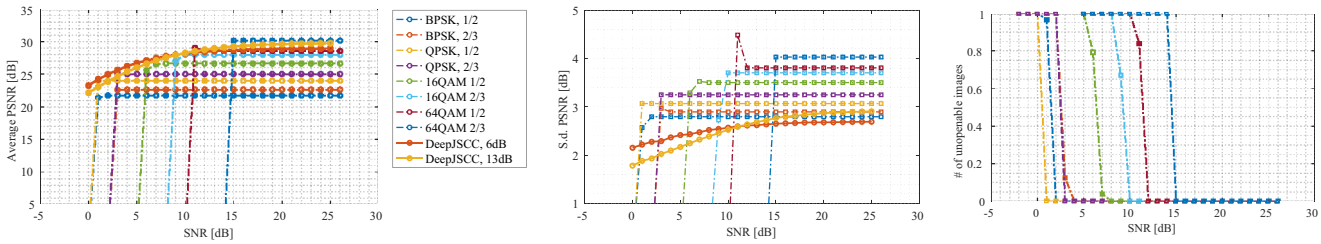
where h and w are the height and width of the images. $x(i,j)$ and $x'(i,j)$ are the pixels between the images to be compared.

For the baseline, the number of JPEG images that were not opened was also evaluated. Figures 6, 7, and 8 are the results at $c = 4,\ 8,\ 16$, respectively. (a) is the mean PSNR, (b) is the standard deviation of PSNR, and (c) is the broken image rate (= number of broken images/total number of images). All images are compressed by JPEG2000 in the case of the baseline. Overall, the mean PSNR of baseline was better than that of DeepJSCC. In the lower SNR region, a slight advantage for DeepJSCC appeared. Referring to (b), the variance of PSNR was small in the case of DeepJSCC. When considering a general autoencoder-based image compression model, it may be vulnerable to localized errors. Similar problems have been raised in split computing, where introducing a Dropout layer between the encoder and



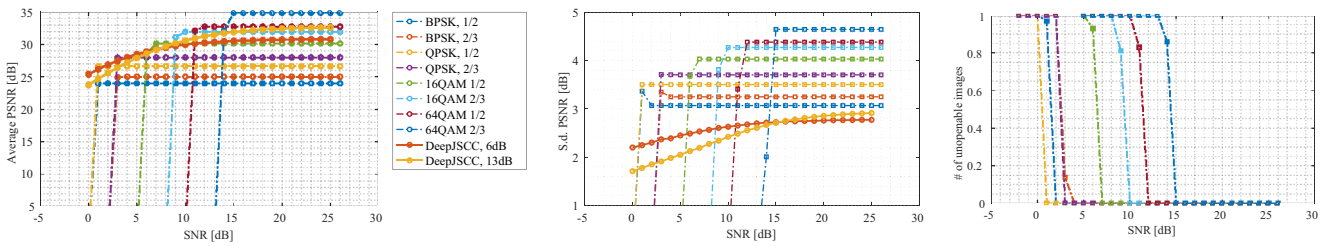
(a) Average PSNR. **(b)** Standard deviation of PSNR. **(c)** Rate of unopenable images.

FIGURE 6. Simulation results at $c = 4$.



(a) Average PSNR. **(b)** Standard deviation of PSNR. **(c)** Rate of unopenable images.

FIGURE 7. Simulation results at $c = 8$.



(a) Average PSNR. **(b)** Standard deviation of PSNR. **(c)** Rate of unopenable images.

FIGURE 8. Simulation results at $c = 16$.

decoder supports to widely distribute important information across the latent space [25]. Similarly, in DeepJSCC, a noise channel is placed between the encoder and decoder during training process, making the model more resilient to localized noise in the channel. From the PSNR results, we observed that even when changing c, the standard deviation remains largely unchanged. Additionally, as the compression ratio decreases, the average PSNR remains stable and increases consistently.

Another important consideration is that adaptive modulation must operate properly when baseline is applied. Failure to follow even the slightest variation in the SNR of the channel causes cliff effects and possibly destroys the image. In addition, DeepJSCC always achieves nearly optimal PSNR without changing the encoder-decoder configuration. These facts indicate that even if the image quality was equivalent to the baseline, the advantages are sufficient. Especially when channel variations are large, such as in moving conditions, the probability of modulation and coding scheme (MCS) selection failure increases even with conventional adaptive modulation. In contrast, DeepJSCC allows reception without modifying the model, thus enabling images to be restored with more appropriate PSNR quality.

It is also possible to optimize adaptive modulation using artificial intelligence (AI)-based techniques. For example, a method for MCS selection using reinforcement learning has been proposed [26]. While this approach allows for maintaining high PSNR with high accuracy, DeepJSCC has the advantage of eliminating the need for adaptive modulation itself. Meanwhile, when DeepJSCC signals coexist with conventional modulated signals such as QAM signal, it is necessary to incorporate a priority policy for DeepJSCC into the RE allocation algorithm.

C. SIMULATION – PAPR EVALUATION –

We estimated the PAPR when using DeepJSCC. In our simulation setting, there were 273 RBs, each containing 12 subcarriers, resulting in a total of 3276 resource elements (REs). We calculated the PAPR for each OFDM symbol. The following five signal types were compared: Random QPSK signal (3276 REs), DeepJSCC signal (3276 REs), DeepJSCC signal (1092 REs) combined with random QPSK signal (2184 REs), white Gaussian noise treated as an OFDM

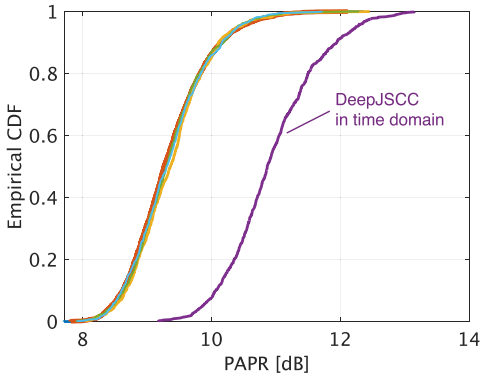


FIGURE 9. PAPR distribution.

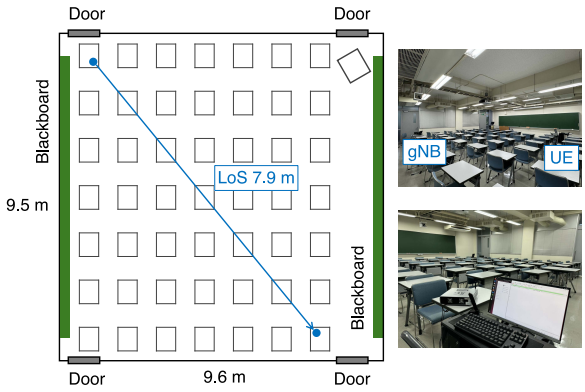


FIGURE 10. Room condition for experiment.

symbol (3276 REs), and a time-domain DeepJSCC signal called the single-carrier (SC) JSCC signal. Figure 9 shows the empirical CDF of PAPR using the CIFAR-10 dataset in the case of $c = 8$. According to (1), when $32 \times 32 \times 8/(2 \times 2^2 \times 2^2) = 256$ symbols are transmitted per image. See [6] for details on the definition of the compression ratio. When employing OFDM, the resulting PAPR was not substantially different from that of the QPSK signal. On the other hand, a higher PAPR was observed when the SC-JSCC signal was used instead of OFDM. Thus, as in typical 5G systems, techniques such as clipping are required to suppress PAPR. Notably, Shao and Gündüz proposed a method for mitigating the PAPR of JSCC signals in [27], and its effectiveness has been demonstrated. In this paper, however, we did not apply any clipping or similar methods, and evaluated the normal JSCC signal.

V. INDOOR EXPERIMENT

This scenario simulates an industrial indoor channel setting with strong multipath reflections, such as a factory or a warehouse. The transmission power was intentionally reduced to emphasize the effects of signal reflections. These conditions are challenging for conventional coding schemes, as multipath interference can cause signal degradation. DeepJSCC is expected to be more robust under such conditions due to its end-to-end optimization of source and channel coding.

A. SETUP

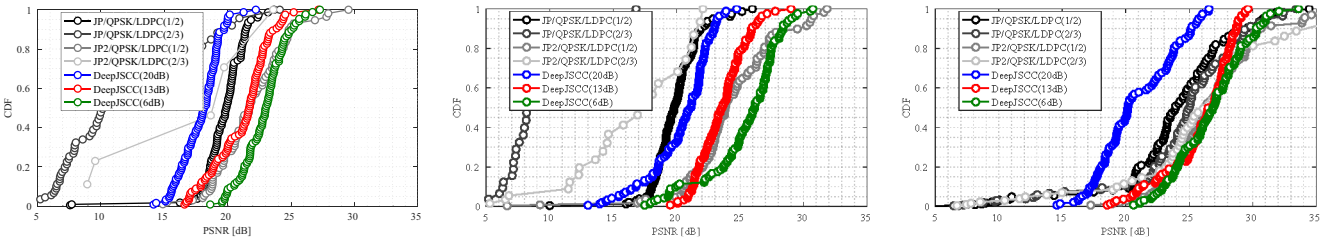
Figure 10 shows the experimental setup. The experiment was conducted in a typical classroom. First, on the gNB side, both gNB and UE are placed on a desk with a height of 70 cm. The number of antennas is one each for both gNB and UE in a single-input-to-single-output (SISO) configuration. To configure a low SNR environment, the transmission power of the gNB was set to -2.0 dBm. The maximum and minimum received power of the UE were the range of -70.0 dBm and -75.0 dBm measured by received signal strength indicator (RSSI).

The reason for conducting experiments around the lower RSSI limit was not only to demonstrate the effectiveness of DeepJSCC in low-SNR region but also, at higher SNRs, it is expected that the results are identical to those obtained in simulations. In our previous work, we conducted verification by wired-connecting the gNB and the UE and obtained nearly same PSNR between experiment and simulation [8]. In addition, according to the specifications of the gNB and UE used in our setup, certain modulation schemes are recommended based on the RSSI level. For example, when RSSI is -75 dBm, even with a direct cable connection between the gNB and UE, 16QAM cannot be reliably demodulated (resulting in a BER of approximately 0.5). Similarly, when the RSSI drops to around -80 dBm, QPSK demodulation also starts to become infeasible, indicating the lower limit of reliable reception.

B. RESULTS

Figure 11 shows the measured CDF of PSNR calculated from the restored images. While a PSNR degradation was observed in DeepJSCC when the training SNR is 13 dB and 20 dB, the PSNR was significantly improved when the training SNR was 6 dB. In particular, the average PSNR (around 0.5 on the vertical axis) was superior to the baseline in most images. However, at $c = 4$, 8, when the CDF approaches 1.0, the PSNR employed the baseline was better than the DeepJSCC. In the $c = 16$ case, the baseline was also better than the DeepJSCC over CDF $= 0.8$. On the other hand, the number of plotted points (images) in the baseline was lower than in DeepJSCC. This is because corrupted images were not included in the plot.

Table 2 shows the results of all measurements. γ_μ and γ_σ are the mean and standard deviation of PSNR, respectively. BER is the bit error rate and N_{img} is the number of corrupted images. As the table shows, DeepJSCC (training SNR $=$ 6 dB) has the best average PSNR and a relatively small standard deviation. On the other hand, the baseline 16QAM signal degraded the BER and corrupted all images. As for the baseline QPSK signal, dozens of images were also corrupted. These results indicate that DeepJSCC can be applied predominantly in practical environments with lowwer SNR and higher compression ratio compared to the baseline



(a) $c = 4$. **(b)** $c = 8$. **(c)** $c = 16$.

FIGURE 11. CDF in indoor experiment.

TABLE 2. Experimental results in indoor experiment.

	$c = 4$				$c = 8$				$c = 16$			
	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}
JP/QPSK/LDPC(1/2)	19.55	2.24	3.19×10^{-3}	10	19.77	1.98	3.36×10^{-6}	1	22.31	6.09	4.74×10^{-3}	11
JP/QPSK/LDPC(2/3)	10.15	4.65	5.48×10^{-4}	45	8.34	2.72	1.84×10^{-3}	74	22.96	6.36	4.46×10^{-3}	21
JP/16QAM/LDPC(1/2)	N/A	N/A	3.61×10^{-2}	100	N/A	N/A	2.30×10^{-2}	100	N/A	N/A	4.94×10^{-2}	100
JP/16QAM/LDPC(2/3)	N/A	N/A	4.17×10^{-2}	100	N/A	N/A	2.87×10^{-2}	100	N/A	N/A	5.80×10^{-2}	100
JP2/QPSK/LDPC(1/2)	21.61	2.76	1.42×10^{-2}	26	23.70	4.07	1.40×10^{-2}	23	26.26	3.71	3.23×10^{-5}	6
JP2/QPSK/LDPC(2/3)	16.10	6.52	1.69×10^{-2}	95	15.31	5.06	1.14×10^{-2}	78	24.00	7.48	1.60×10^{-2}	45
JP2/16QAM/LDPC(1/2)	N/A	N/A	9.86×10^{-2}	100	N/A	N/A	5.98×10^{-2}	100	N/A	N/A	4.67×10^{-3}	100
JP2/16QAM/LDPC(2/3)	N/A	N/A	6.04×10^{-2}	100	N/A	N/A	4.69×10^{-2}	100	N/A	N/A	5.47×10^{-2}	100
DeepJSCC(20dB)	17.79	1.54	N/A	0	20.13	2.56	N/A	0	20.49	2.99	N/A	0
DeepJSCC(13dB)	20.97	2.28	N/A	0	23.46	1.86	N/A	0	25.67	2.82	N/A	0
DeepJSCC(6dB)	22.78	1.74	N/A	0	25.09	3.11	N/A	0	26.45	2.81	N/A	0

(a) $c = 4$, PSNR: 21.2 dB. **(b)** $c = 8$, PSNR: 28.3 dB. **(c)** $c = 16$, PSNR: 30.2 dB.

FIGURE 12. Example image in indoor experiment in the 6-dB training SNR case.

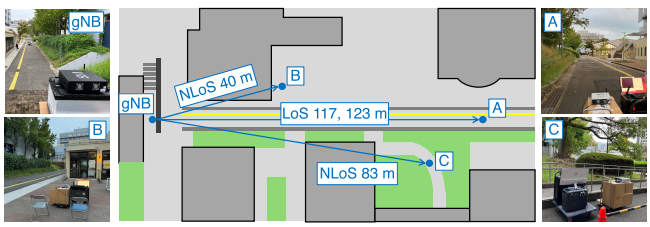


FIGURE 13. Field condition for outdoor experiment.

method. Figure 12 shows an example of the transmitted images. When $c = 4$, the distortion was observed, while the images are relatively clear.

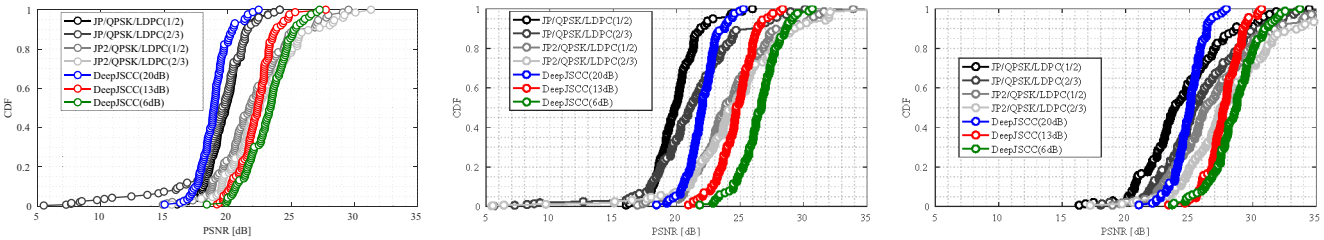
VI. OUTDOOR EXPERIMENT

A. SETUP

In our outdoor experiments, we evaluated DeepJSCC under four different transmission scenarios to analyze its performance across various wireless conditions. Figure 13 shows the experimental setup. We conducted experiments at three locations, A, B, and C. The gNB was placed at a height of 3.0 m and the UE was placed at a height of 65.7 cm. The number of antennas was one each for both the gNB and the UE in a SISO configuration. The details of each scenario are as follows:

1) LOS ENVIRONMENT 1 (RSSI NEAR LOWER LIMIT) AT LOCATION A

This scenario represents a long-range outdoor communication setup where the received signal strength indicator (RSSI) is at the minimum level required for reliable reception. Applications include surveillance cameras or drone-based monitoring systems operating at the edge of coverage areas. In such conditions, JPEG2000 and LDPC-based coding are susceptible to packet loss, leading to significant image degradation, whereas DeepJSCC is expected to maintain better image quality by dynamically adapting to channel variations. The distances between the gNB and the UE are 117.0 m. The transmission power of the gNB was set to



(a) $c = 4$. **(b)** $c = 8$. **(c)** $c = 16$.

FIGURE 14. CDF in outdoor experiment at point A (117 m).

(a) $c = 4$, PSNR: 25.1 dB. **(b)** $c = 8$, PSNR: 28.7 dB. **(c)** $c = 16$, PSNR: 30.7 dB.

FIGURE 15. Example image in outdoor experiment at point A (117 m) in the 6-dB training SNR case.

13.0 dBm. The maximum received power of the UE at this point was −75.0 dBm measured by RSSI.

2) LOS ENVIRONMENT 2 (RSSI BELOW LOWER LIMIT) AT LOCATION A

Similar to the previous outdoor LoS scenario but conducted in an even weaker signal environment where the RSSI falls below the reception threshold. This scenario is designed to evaluate the resilience of DeepJSCC in extreme wireless conditions where traditional methods typically fail. The distances between the gNB and the UE are 123.0 m. The gNB's transmission power was set to 3 dBm. The received power of the UE was below the lower RSSI measurement limit (< -75 dBm). However, since the RSSI was −60 dBm before the transmission power was attenuated by 20 dB, the received power was estimated to be around −80 dBm.

3) NLOS ENVIRONMENT 1 (RSSI BELOW LOWER LIMIT) AT LOCATION B

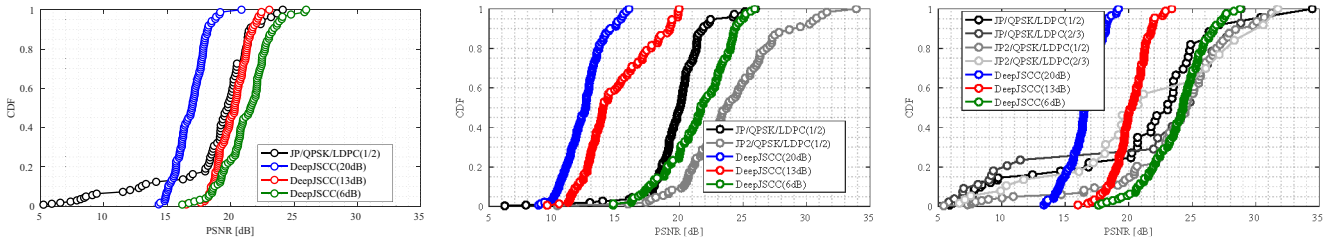
This scenario mimics urban environments where buildings or other obstacles obstruct direct signal paths, causing severe multipath fading and signal attenuation. The RSSI in this scenario is below the minimum reception threshold. Such conditions are typical in smart city infrastructure and V2X (Vehicle-to-Everything) communications. DeepJSCC's ability to reconstruct images despite high interference and signal degradation is assessed here. The gNB transmission power was set to 23 dBm. The received power of the UE fluctuated within the range between −70 and −75 dBm. The shielding object was a concrete building with glass walls.

4) NLOS ENVIRONMENT 2 (RSSI BELOW LOWER LIMIT) AT LOCATION C

This scenario represents an extreme non-line-of-sight (NLoS) environment where the received signal strength indicator (RSSI) is well below the lower reception limit. In this case, even when using QPSK signal, transmission errors are inevitable, leading to complete failure of conventional reception. The purpose of this experiment is to investigate whether DeepJSCC can maintain any meaningful image reconstruction under such severe conditions where traditional source-channel separated coding schemes would fail completely. The gNB transmission power was set at 23 dBm while the UE receive power was significantly below the lower limit of the measurement. The shielding object was a concrete building with no glass walls.

B. RESULTS

Figures 14–20 shows the CDF of the PSNR calculated from the restored images at each measurement location. Table 3–6 shows the full measurement results for each location; as in the case of Indoor, the characteristics were greatly improved when the trained SNR was 6 dB. Since the experiment was conducted near the lower RSSI measurement limit and QPSK demodulation is almost impossible, it is expected that the measurement conditions were similar or close to the training SNR = 6 dB. The discrepancy between the training SNR and the actual SNR causes performance degradation. However, when comparing training SNR = 6 dB and 13 dB, only about a 1.0 dB difference was observed. This indicates that sufficient performance can be achieved even without strictly estimating the channel SNR. On the other hand, when

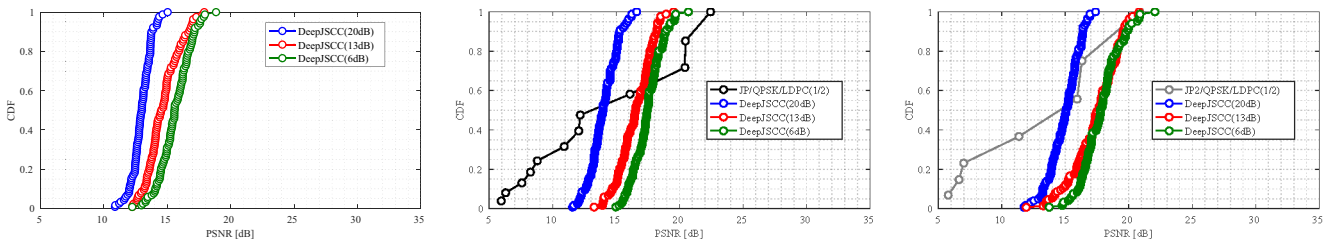


(a) $c = 4$. **(b)** $c = 8$. **(c)** $c = 16$.

FIGURE 16. CDF in outdoor experiment at point A (123 m).

(a) $c = 4$, PSNR: 21.8 dB. **(b)** $c = 8$, PSNR: 21.0 dB. **(c)** $c = 16$, PSNR: 20.7 dB.

FIGURE 17. Example image in outdoor experiment at point A (123 m) in the 6-dB training SNR case.



(a) $c = 4$. **(b)** $c = 8$. **(c)** $c = 16$.

FIGURE 18. CDF in outdoor experiment at point B.

(a) $c = 4$, PSNR: 17.6 dB. **(b)** $c = 8$, PSNR: 18.2 dB. **(c)** $c = 16$, PSNR: 18.4 dB.

FIGURE 19. Example image in outdoor experiment at point B in the 6-dB training SNR case.

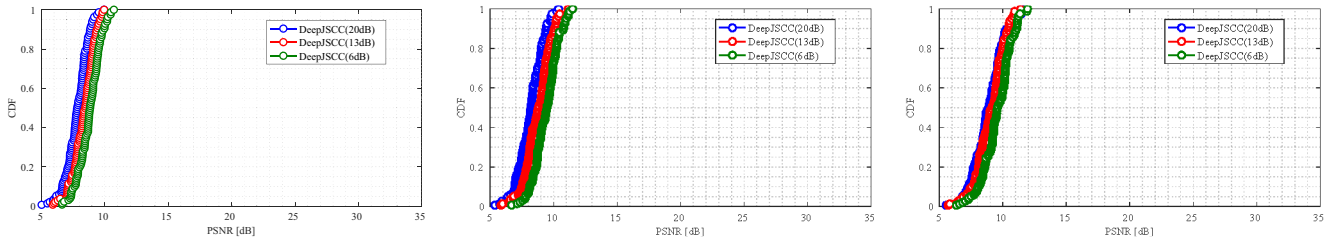
training SNR = 20 dB, the performance degrades by nearly 5.0 dB compared to training SNR = 6 dB. Because of low noise level at SNR = 20 dB, the latent space could not obtain noise resilience.

1) LOS ENVIRONMENT 1 (RSSI NEAR LOWER LIMIT) AT LOCATION A

For location A (117 m), DeepJSCC showed better characteristics at almost all points; for the JPEG2000/QPSK/LDPC(2/3) combination, some images showed higher PSNR than DeepJSCC. However, from Table 3, the characteristics of DeepJSCC are superior when comparing the mean and variance values of PSNR. Figure 15 shows an example of the transmitted images. We obtained the clear images.

2) LOS ENVIRONMENT 2 (RSSI BELOW LOWER LIMIT) AT LOCATION A

In the case of location A (123 m), the conditions were set more severely, the baseline failed to recover most of the



(a) $c = 4$. **(b)** $c = 8$. **(c)** $c = 16$.

FIGURE 20. CDF in outdoor experiment at point C.

TABLE 3. Experimental results in outdoor experiment at point A (117 m).

	$c=4$				$c=8$				$c=16$			
	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}
JP/QPSK/LDPC(1/2)	19.77	1.49	0	0	19.89	1.71	0	0	24.08	3.23	3.14×10^{-7}	0
JP/QPSK/LDPC(2/3)	18.42	3.76	2.10×10^{-5}	3	20.49	4.17	9.04×10^{-6}	0	25.62	3.35	7.02×10^{-7}	0
JP/16QAM/LDPC(1/2)	N/A	N/A	1.60×10^{-2}	100	N/A	N/A	7.06×10^{-3}	100	6.42	0	5.71×10^{-3}	99
JP/16QAM/LDPC(2/3)	N/A	N/A	1.76×10^{-2}	100	N/A	N/A	6.39×10^{-3}	100	9.85	0	78.07×10^{-3}	99
JP2/QPSK/LDPC(1/2)	21.74	2.81	0	0	24.00	3.08	0	0	26.56	3.57	1.86×10^{-6}	0
JP2/QPSK/LDPC(2/3)	22.49	2.95	1.60×10^{-5}	1	23.58	5.03	3.63×10^{-5}	3	27.73	3.66	2.54×10^{-6}	1
JP2/16QAM/LDPC(1/2)	N/A	N/A	1.23×10^{-2}	100	N/A	N/A	1.31×10^{-2}	100	6.12	0	7.87×10^{-3}	99
JP2/16QAM/LDPC(2/3)	N/A	N/A	1.31×10^{-2}	100	N/A	N/A	1.04×10^{-2}	100	N/A	N/A	9.84×10^{-3}	100
DeepJSCC(20dB)	18.86	1.22	N/A	0	21.97	1.26	N/A	0	24.93	1.22	N/A	0
DeepJSCC(13dB)	22.20	1.45	N/A	0	24.66	1.49	N/A	0	27.67	1.36	N/A	0
DeepJSCC(6dB)	23.09	1.77	N/A	0	26.45	1.68	N/A	0	28.52	1.92	N/A	0

TABLE 4. Experimental results in outdoor experiment at point A (123 m).

	$c=4$				$c=8$				$c=16$			
	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}
JP/QPSK/LDPC(1/2)	17.47	4.97	4.89×10^{-3}	27	19.47	2.83	1.06×10^{-4}	5	17.74	7.98	7.29×10^{-2}	55
JP/QPSK/LDPC(2/3)	N/A	N/A	3.10×10^{-2}	100	N/A	N/A	2.07×10^{-2}	100	16.60	9.24	2.73×10^{-3}	76
JP/16QAM/LDPC(1/2)	N/A	N/A	1.38×10^{-1}	100	N/A	N/A	1.30×10^{-1}	100	N/A	N/A	2.15×10^{-1}	100
JP/16QAM/LDPC(2/3)	N/A	N/A	1.29×10^{-1}	100	N/A	N/A	1.29×10^{-1}	100	N/A	N/A	1.41×10^{-1}	100
JP2/QPSK/LDPC(1/2)	N/A	N/A	1.41×10^{-1}	100	23.53	3.74	5.91×10^{-3}	12	21.94	6.41	9.01×10^{-3}	33
JP2/QPSK/LDPC(2/3)	N/A	N/A	4.37×10^{-2}	100	N/A	N/A	2.50×10^{-2}	100	18.64	7.71	3.76×10^{-2}	79
JP2/16QAM/LDPC(1/2)	N/A	N/A	1.18×10^{-1}	100	N/A	N/A	1.16×10^{-1}	100	N/A	N/A	1.77×10^{-1}	100
JP2/16QAM/LDPC(2/3)	N/A	N/A	1.17×10^{-1}	100	N/A	N/A	1.50×10^{-1}	100	N/A	N/A	1.72×10^{-1}	100
DeepJSCC(20dB)	16.76	1.22	N/A	0	12.30	1.54	N/A	0	16.42	1.37	N/A	0
DeepJSCC(13dB)	20.10	1.26	N/A	0	14.50	2.52	N/A	0	20.08	1.40	N/A	0
DeepJSCC(6dB)	21.17	1.87	N/A	0	21.17	2.67	N/A	0	23.73	2.37	N/A	0

TABLE 5. Experimental results in outdoor experiment at point B.

	$c=4$				$c=8$				$c=16$			
	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}
JP/QPSK/LDPC(1/2)	N/A	N/A	1.74×10^{-1}	100	12.61	5.860	1.20×10^{-1}	88	N/A	N/A	2.15×10^{-1}	100
JP/QPSK/LDPC(2/3)	N/A	N/A	1.66×10^{-1}	100	N/A	N/A	1.78×10^{-1}	100	N/A	N/A	2.58×10^{-1}	100
JP/16QAM/LDPC(1/2)	N/A	N/A	1.87×10^{-1}	100	N/A	N/A	2.34×10^{-1}	100	N/A	N/A	2.60×10^{-1}	100
JP/16QAM/LDPC(2/3)	N/A	N/A	1.51×10^{-1}	100	N/A	N/A	2.40×10^{-1}	100	N/A	N/A	2.30×10^{-1}	100
JP2/QPSK/LDPC(1/2)	N/A	N/A	1.49×10^{-1}	100	N/A	N/A	1.58×10^{-1}	100	11.98	5.85	3.97×10^{-2}	93
JP2/QPSK/LDPC(2/3)	N/A	N/A	1.37×10^{-1}	100	N/A	N/A	1.62×10^{-1}	100	N/A	N/A	1.67×10^{-1}	100
JP2/16QAM/LDPC(1/2)	N/A	N/A	1.90×10^{-1}	100	N/A	N/A	1.76×10^{-1}	100	N/A	N/A	2.06×10^{-1}	100
JP2/16QAM/LDPC(2/3)	N/A	N/A	1.92×10^{-1}	100	N/A	N/A	1.49×10^{-1}	100	N/A	N/A	2.08×10^{-1}	100
DeepJSCC(20dB)	12.95	0.772	N/A	0	13.93	1.10	N/A	0	14.86	1.23	N/A	0
DeepJSCC(13dB)	14.61	1.29	N/A	0	16.42	1.33	N/A	0	17.20	1.92	N/A	0
DeepJSCC(6dB)	15.55	1.24	N/A	0	17.42	1.15	N/A	0	17.70	1.53	N/A	0

images. However, in the case of JPEG/QPSK/LDPC(1/2) and JPEG2000/QPSK/LDPC(1/2) with $c = 8$, the characteristics that outperform DeepJSCC were confirmed despite the presence of corrupted images. Table 4 also shows that DeepJSCC tends to be disadvantageous in the range where QPSK can be demodulated adequately. However, even under this baseline-dominated situation, DeepJSCC is still valuable from the perspective of continuing to output stable PSNR images without image corruption. Figure 17 shows an example of the transmitted images. In this region, where the reception sensitivity falls below the minimum RSSI which the UE can measure, image distortion begun to become noticeable.

3) NLOS ENVIRONMENT 1 (RSSI BELOW LOWER LIMIT) AT LOCATION B

For NLoS environments, the figure and table show that DeepJSCC was able to stably restore the images, albeit at a low PSNR, even under conditions where the baseline was barely restored. These results suggest that DeepJSCC has high utilization value. Figure 19 shows an example of the

TABLE 6. Experimental results in outdoor experiment at point C.

	$c=4$				$c=8$				$c=16$			
	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}	γ_μ [dB]	γ_σ [dB]	BER	N_{img}
JP/QPSK/LDPC(1/2)	N/A	N/A	0.489	100	N/A	N/A	0.506	100	N/A	N/A	0.491	100
JP/QPSK/LDPC(2/3)	N/A	N/A	0.488	100	N/A	N/A	0.502	100	N/A	N/A	0.498	100
JP2/QPSK/LDPC(1/2)	N/A	N/A	0.517	100	N/A	N/A	0.530	100	N/A	N/A	0.533	100
JP2/QPSK/LDPC(2/3)	N/A	N/A	0.507	100	N/A	N/A	0.524	100	N/A	N/A	0.527	100
DeepJSCC(20dB)	7.72	0.91	N/A	0	8.06	1.03	N/A	0	8.72	1.22	N/A	0
DeepJSCC(13dB)	8.27	0.89	N/A	0	8.63	1.07	N/A	0	8.87	1.18	N/A	0
DeepJSCC(6dB)	8.73	0.89	N/A	0	9.32	0.98	N/A	0	9.44	1.17	N/A	0

(a) $c=4$, PSNR: 9.3 dB. **(b)** $c=8$, PSNR: 9.5 dB. **(c)** $c=16$, PSNR: 10.1 dB.

FIGURE 21. Example image in outdoor experiment at point C in the 6-dB training SNR case.

transmitted images. Although unclear, the content of the image can still be recognized.

4) NLOS ENVIRONMENT 2 (RSSI BELOW LOWER LIMIT) AT LOCATION C

From Table 6, it appears that DeepJSCC has successfully restored images, with an average PSNR of around 8 to 9 dB, while conventional methods have completely failed to recover them. However, in PSNR calculation, when a completely random image is output with pixel values uniformly distributed in the range of 0 to 255, as derived from (3), the PSNR is given by $10\log_{10}(255^2/(255/2)^2) = 6$ dB. This means that as the PSNR approaches 6 dB, the restoration has effectively failed. In the actual image examples in Fig. 21, the content of the original image is completely unreadable. This suggests that, even when using DeepJSCC, image reconstruction is almost entirely unsuccessful in this case.

5) PROCESSING TIME

Regarding the processing time of the encoder and decoder, we measured it using a high-end consumer GPU, the NVIDIA GeForce RTX 4090. For a 256 × 256-pixel image, the encoder required approximately 5 ms, and the decoder took around 7 ms. In addition to this, transmission delay must also be considered. For example, to transmit a 256 × 256-pixel image with $c = 8$, a total of 16384 IQ symbols are required. Assuming that all 273 RBs are utilized, and each OFDM symbol accommodates 3276 REs, the transmission can be completed in approximately 5 OFDM symbols. With a subcarrier spacing of 30 kHz, the transmission delay is approximately 166.5 μs. Even if the number of usable RBs is limited to 10%, the additional transmission delay remains under 2 ms. Therefore, the total end-to-end latency, including encoding, transmission, and decoding, is confirmed to be well below 20 ms.

VII. LIMITATION

For the implementation of DeepJSCC in cellular systems. Since the input signal to the decoder is an IQ signal, it is necessary to obtain the IQ signal from the receiver. In other words, it is not easy to simply attach a DeepJSCC decoder externally to a gNB. It is necessary to modify the gNB such as our gNB. In addition, since progress in the deep learning field is extremely rapid, it is necessary to be able to change the layer structure flexibly. This also means that it is difficult to implement hardware such as ASICs along with DSPs in gNBs and UEs. That is, a GPU or CPU circuit for DeepJSCC must be prepared and hardware circuits in gNB and UE are modified to be able to input and extract IQ signals. Not only is there concern that using a GPU results in higher power consumption, but also timing jitter due to software processing. FPGA is a promising solution for addressing this issue. We have currently verified the use of an FPGA and have confirmed operation at 30 FPS, which is a typical video frame rate [16].

For the signal processing, although we indicated DeepJSCC is extremely effective at low SNR, the IQ signal generated by DeepJSCC cannot be obtained unless the bit information for the 5G system to recognize the control signal such as PDCCH/PUCCH and perform the equalization process is obtained. That is, there is still a lower limit to the SNR that can be received the DeepJSCC signal, so advanced synchronization and equalization mechanism are needed to further emphasize the performance of DeepJSCC.

In this study, the channel is assumed to be static. However, future applications may require adaptation to high-mobility scenarios with significant Doppler spread. If pilot-based channel estimation is available and the channel is treated as

quasi-static, DeepJSCC signals can be equalized accordingly. On the other hand, in scenarios where channel estimation must be performed concurrently with data decoding, new DeepJSCC architectures are required, as the method presented in this paper is unlikely to cope effectively with highly time-varying channels.

Furthermore, this paper conducted the experiment using SISO system. In the future, when extending to multiple-input multiple-output (MIMO) technique, ghost imaging may occur. Since DeepJSCC transmits the latent space output by the encoder directly, interference between latent spaces can appear as ghost images in the decoder. Therefore, when implementing MIMO technique, the decoder design become more important to suppress the ghost imaging.

VIII. CONCLUSION

This paper conducted the first indoor/outdoor field trials of DeepJSCC in a 5G environment, revealing its practical feasibility and robustness under diverse propagation conditions, including both LoS and NLoS scenarios. Experimental results demonstrated that DeepJSCC consistently enhanced image transmission quality, particularly in low-SNR regions where the baseline methods such as JPEG2000 and LDPC-based coding with typical modulation formats frequently failed. The system achieved higher average PSNR and no image corruption compared to baseline methods, with a notable ability to maintain stable performance across varying compression ratios and channel conditions.

A significant insight from the experiments is that DeepJSCC excels in scenarios with extremely low received signal power, effectively mitigating the cliff effect observed in baseline schemes. The successful integration of DeepJSCC with a modified OFDM-based 5G framework highlights its potential as a transformative approach for real-time, high-quality image transmission.

Furthermore, we demonstrated that even a DeepJSCC model trained solely on an AWGN channel maintains stable performance in real-world 5G environments. This result highlights the generalization ability of the model without requiring adaptation to specific channel models. We also estimated the end-to-end latency, including encoding, transmission, and decoding, and confirmed that the total processing time remains well below 20 ms, satisfying the requirements for real-time applications such as drone control. These findings reinforce the feasibility of deploying DeepJSCC in practical wireless systems.

ACKNOWLEDGMENT

The authors thank Ryunosuke Yamamoto, Tomoka Mori, Hiroki Aoyama, Taichi Isobe, and Keisuke Toyoshima in Osaka University for supporting the outdoor experiments.

REFERENCES

[1] A. Aijaz, "Private 5G: The future of industrial wireless," *IEEE Ind. Electron. Mag.*, vol. 14, no. 4, pp. 136–145, Dec. 2020, doi: 10.1109/MIE.2020.3004975.

[2] M. Wen et al., "Private 5G networks: Concepts, architectures, and research landscape," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 1, pp. 7–25, Jan. 2022, doi: 10.1109/JSTSP.2021.3137669.

[3] A. Mahmood, S. F. Abedin, T. Sauter, M. Gidlund, and K. Landernäs, "Factory 5G: A review of industry-centric features and deployment options," *IEEE Ind. Electron. Mag.*, vol. 16, no. 2, pp. 24–34, Jun. 2022, doi: 10.1109/MIE.2022.3149209.

[4] D. Gündüz et al., "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 5–41, Jan. 2023, doi: 10.1109/JSAC.2022.3223408.

[5] K. Niu et al., "A paradigm shift toward semantic communications," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 113–119, Nov. 2022, doi: 10.1109/MCOM.001.2200099.

[6] E. Bourtsoulatze, D. B. 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, doi: 10.1109/TCCN.2019.2919300.

[7] D. B. Kurka, and D. Gündüz, "DeepJSCC-*f*: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, May 2020, doi: 10.1109/JSAIT.2020.2987203.

[8] K. Matsumoto et al., "Implementation of deep joint source-channel coding on 5G systems for image transmission," in *Proc. IEEE 98th Veh. Technol. Conf.*, Sep. 2023, pp. 1–5, doi: 10.1109/VTC2023-Fall60731.2023.10333673.

[9] T. -Y. Tung and D. Gündüz, "DeepWiVe: Deep-learning-aided wireless video transmission," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2570–2583, Sep. 2022, doi: 10.1109/JSAC.2022.3191354.

[10] S. Ibuki, T. Okamoto, T. Fujihashi, T. Koike-Akino, and T. Watanabe, "Rateless deep graph joint source channel coding for holographic-type communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pp. 3330–3335, Dec. 2023, doi: 10.1109/GLOBECOM54140.2023.10437920.

[11] T. -Y. Tung and D. Gündüz, "Deep joint source-channel and encryption coding: Secure semantic communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, pp. 5620–5625, May 2023, doi: 10.1109/ICC45041.2023.10278612.

[12] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 55, pp. 1–21, Mar. 2019.

[13] S. Shimizu, T. Nishio, S. Saito, Y. Hirose, C. Yen-Hsiu, and S. Shirakawa, "Neural architecture search for improving latency-accuracy trade-off in split computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2022, pp. 1864–1870, doi: 10.1109/GCWkshps56602.2022.10008544.

[14] Z. Lyu, G. Zhu, J. Xu, B. Ai, and S. Cui, "Semantic communications for image recovery and classification via deep joint source and channel coding," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8388–8404, Aug. 2024, doi: 10.1109/TWC.2023.3349330.

[15] Y. Inoue, D. Hisano, K. Maruta, Y. Hara-Azumi, and Y. Nakayama, "Deep joint source-channel coding and modulation for underwater acoustic communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–7, doi: 10.1109/GLOBECOM46510.2021.9685931.

[16] S. Fujimaki, Y. Inoue, D. Hisano, K. Maruta, Y. Nakayama, and Y. Hara-Azumi, "A self-attention network for deep JSCCM: The design and FPGA implementation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2022, pp. 6390–6395, doi: 10.1109/GLOBECOM48099.2022.10001518.

[17] M. Yang, C. Bian, and H.-S. Kim, "Deep joint source channel coding for wireless image transmission with OFDM," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2021, pp. 1–6, doi: 10.1109/ICC42927.2021.9500996.

[18] H. Wu, Y. Shao, K. Mikolajczyk, and D. Gündüz, "Channel-adaptive wireless image transmission with OFDM," *IEEE Wireless Commun. Lett.*, vol. 11, no. 11, pp. 2400–2404, Nov. 2022, doi: 10.1109/LWC.2022.3204837.

[19] H. Hu, X. Zhu, F. Zhou, W. Wu, R. Q. Hu, and H. Zhu, "One-to-many semantic communication systems: Design, implementation, performance evaluation," *IEEE Commun. Lett.*, vol. 26, no. 12, pp. 2959–2963, Dec. 2022, doi: 10.1109/LCOMM.2022.3203984.

[20] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep source-channel coding for sentence semantic transmission with HARQ," *IEEE Trans. Commun.*, vol. 70, no. 8, pp. 5225–5240, Aug. 2022, doi: 10.1109/TCOMM.2022.3180997.

[21] M. Liu, W. Chen, J. Xu, and B. Ai, "Real-time implementation and evaluation of SDR-based deep joint source-channel coding," in *Proc. IEEE 96th Veh. Technol. Conf.*, Sep. 2022, pp. 1–5, doi: 10.1109/VTC2022-Fall57202.2022.10012971.

[22] J. Ding, P. Jiang, C.-K. Wen, and S. Jin, "Adaptive wireless image semantic transmission and over-the-air testing," May 2024, *arXiv:2405.13403*.

[23] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1122–1131.

[24] R. Yamamoto et al., "Impact of training models on deep joint source-channel coding applicable to 5G systems," *IEICE Commun. Exp.*, vol. 13, no. 12, pp. 466–469, Dec. 2024, doi: 10.23919/comex.2024COL0014.

[25] S. Itahara, T. Nishio, Y. Koda, and K. Yamamoto, "Communication-oriented model fine-tuning for packet-loss resilient distributed inference under highly lossy IoT networks," *IEEE Access*, vol. 10, pp. 14969–14979, 2022, doi: 10.1109/ACCESS.2022.3149336.

[26] L. Zhang, J. Tan, Y.-C. Liang, G. Feng, and D. Niyato, "Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3281–3294, Jun. 2019, doi: 10.1109/TWC.2019.2912754.

[27] Y. Shao and D. Gündüz, "Semantic communications with discrete-time analog transmission: A PAPR perspective," *IEEE Wireless Commun. Lett.*, vol. 12, no. 3, pp. 510–514, Mar. 2023, doi: 10.1109/LWC.2022.3232946.