Selective Depthwise Separable Convolution for Lightweight Joint Source-Channel Coding in Wireless Image Transmission

Ming Ye, *Member, IEEE,* Kui Cai, *Senior Member, IEEE,* Cunhua Pan, *Senior Member, IEEE,* Zhen Mei, *Member, IEEE,* Wanting Yang, and Chunguo Li, *Senior Member, IEEE*

***Abstract*—Depthwise separable convolutional (DSConv) layers have been successfully applied to deep learning (DL)-based joint source-channel coding (JSCC) schemes to reduce computational complexity. However, a systematic investigation of the layer-wise and ratio-wise replacement of standard convolutional (Conv) layers with DSConv layers in JSCC systems for wireless image transmission remains largely unexplored. In this letter, we propose a configurable lightweight JSCC framework that incorporates a selective replacement strategy, enabling flexible Conv-to-DSConv replacement at different replacement ratios and positions. By varying the replacement ratio, we obtain models with different computational complexities and analyze their impact on reconstruction performance. Furthermore, we investigate how replacements at different encoder and decoder depths influence reconstruction quality under a fixed replacement ratio. Our results show that Conv-to-DSConv replacement at the intermediate layers of the encoder and decoder achieves a favorable complexity-performance trade-off, revealing layer-wise redundancy in DL-based JSCC systems. Extensive experiments further demonstrate that the proposed framework achieves substantial parameter reduction with only slight performance degradation, enabling flexible complexity-performance trade-offs for resource-constrained edge devices.**

***Index Terms*—Lightweight joint source-channel coding, deep learning, depthwise separable convolution, wireless image transmission.**

I. Introduction

SEMANTIC communication has emerged as a promising technology for sixth-generation (6G) communication systems, serving as an innovative paradigm for integrating communications and artificial intelligence [1]. Unlike traditional communication systems, it focuses on conveying the meaning of transmitted information instead of solely minimizing bit error rates. Deep learning (DL)-based joint source-channel coding (JSCC) has become a promising and widely adopted enabler of this paradigm [2], [3]. It directly maps the original information into continuous channel inputs, facilitating end-to-end optimization of the communication system [4], [5].

M. Ye, K. Cai, and W. Yang are with the Singapore University of Technology and Design, Singapore 487372 (e-mail: 230198460@aa.seu.edu.cn; cai_kui@sutd.edu.sg; wanting_yang@sutd.edu.sg).

C. Pan and C. Li are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: cpan@seu.edu.cn; chunguoli@seu.edu.cn).

Z. Mei is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: meizhen@njust.edu.cn).

Recent advancements in DL, especially in image processing [6], [7] and natural language processing [8], have further accelerated the development of semantic communication. Nevertheless, the limited computational, memory, and energy resources of edge devices pose significant challenges to the deployment of DL-based JSCC models. Reducing the model size of DL-based JSCC systems is therefore essential, as it directly affects real-time performance, storage requirements, energy consumption, and deployment flexibility. Consequently, there is a pressing need for lightweight DL-based JSCC methods that achieve acceptable performance with low computational complexity.

Several recent studies have explored lightweight DL-based JSCC approaches [11]–[15]. To reduce computational overhead, the authors of [9] proposed a JSCC method based on a state-space model architecture for wireless image transmission. A JSCC framework based on a Swin Transformer backbone, termed SwinJSCC, was proposed in [10], achieving lower computational complexity than conventional Transformer-based models. However, both methods still incur high computational complexity, which limits their deployment on resource-constrained edge devices. In [11], a lightweight method for semantic image reconstruction and classification was proposed, which employs ConvNeXt-based modules with depthwise separable convolutional (DSConv) layers to reduce computational complexity. Similarly, the authors of [12] proposed a lightweight DL-based JSCC approach, termed DeepJSCC-T, which uses ConvNeXt-based modules to reduce model complexity for wireless image transmission. A modified MobileNetV2 comprising DSConv layers was used in [13] to reduce computational overhead in the task-oriented JSCC scheme for downstream tasks such as face detection and image classification. A lightweight DL-based JSCC framework with an adaptive module was proposed in [14], where convolutional (Conv) layers were replaced with DSConv layers to reduce model complexity for feature extraction. The authors of [15] proposed a lightweight JSCC method for image reconstruction, where DSConv layers and a mixture block were used to reduce complexity. However, the aforementioned DL-based JSCC works [11]–[15] mainly focus on developing specific lightweight architectures, while a systematic investigation of Conv-to-DSConv replacement at different layer positions and ratios in JSCC systems for wireless image transmission is still lacking.

To address these limitations, we propose a lightweight

DL-based JSCC framework, termed DSC-JSCC, for wireless image transmission. The framework incorporates a selective replacement strategy that enables flexible Conv-to-DSConv replacement at different layer positions and ratios. Specifically, Conv layers in the encoder and transposed convolutional (TConv) layers in the decoder are selectively replaced with their DSConv and depthwise separable transposed convolutional (DSTConv) counterparts, respectively, thereby reducing computational complexity. The primary contributions of this letter are summarized as follows:

- We propose DSC-JSCC, a configurable lightweight DL-based JSCC framework that significantly reduces computational complexity while maintaining comparable reconstruction performance through a selective replacement strategy.
- We systematically analyze the impact of Conv-to-DSConv replacement at different layer positions and ratios on performance. Our results show that Conv-to-DSConv replacement at the intermediate layers of the encoder and decoder achieves a favorable complexity-performance trade-off and reveals layer-wise redundancy in DL-based JSCC systems.
- Extensive experiments demonstrate that the proposed framework achieves significant parameter reduction with only slight performance degradation, enabling flexible complexity-performance trade-offs for resource-constrained edge devices.

II. System Model

We consider an end-to-end DL-based JSCC transmission system consisting of a trainable encoder $f_{\boldsymbol{\theta}}$, a trainable decoder $g_{\boldsymbol{\psi}}$, and a noisy wireless channel, as shown in Fig. 1. $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ denote the parameters of the encoder and the decoder, respectively. The source image $\boldsymbol{x} \in \mathbb{R}^d$ is mapped into a complex-valued vector $\boldsymbol{z} \in \mathbb{C}^k$ by the encoder as

$$\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) \in \mathbb{C}^k, \tag{1}$$

where ρ is the bandwidth compression ratio defined as $\rho = k/d$. After encoding, the vector $\boldsymbol{z}$ is transmitted over a Rayleigh slow-fading channel, yielding

$$\hat{\boldsymbol{z}} = \eta(\boldsymbol{z}, \sigma^2) = h\boldsymbol{z} + \boldsymbol{n}, \tag{2}$$

where $h \sim \mathcal{CN}(0,1)$ denotes the Rayleigh fading channel coefficient, $\eta(\cdot)$ denotes the channel transfer function, and $\boldsymbol{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ denotes circularly symmetric complex Gaussian noise, with σ^2 representing the noise power. The transfer function $\eta(\cdot)$ can be extended to other channel models, as long as it remains differentiable [4], [5]. Note that the vector $\boldsymbol{z}$ is obtained by reshaping the output of the final convolutional layer of the encoder into a complex-valued vector $\tilde{\boldsymbol{z}}$, which is then normalized to satisfy the average transmit power constraint $\tilde{P}$ as $\boldsymbol{z} = \sqrt{k\tilde{P}}\frac{\tilde{\boldsymbol{z}}}{\sqrt{\tilde{\boldsymbol{z}}^H\tilde{\boldsymbol{z}}}}$. The average transmit power is normalized to $\tilde{P} = 1$, and the signal-to-noise ratio (SNR) is defined as $\mathrm{SNR} = 10\log_{10}(\tilde{P}/\sigma^2)$.

Finally, the corrupted vector $\hat{\boldsymbol{z}}$ is fed into the decoder to reconstruct the image as

$$\hat{\boldsymbol{x}} = g_{\boldsymbol{\psi}}(\hat{\boldsymbol{z}}) \in \mathbb{R}^d. \tag{3}$$

Unlike existing lightweight DL-based JSCC methods that mainly focus on designing specific lightweight architectures, this work systematically investigates how Conv-to-DSConv replacement at different layer positions and ratios affects performance and computational complexity. The objective is to develop a configurable lightweight DL-based JSCC framework that achieves a favorable complexity-performance trade-off and provides practical design guidelines for Conv-to-DSConv replacement in DL-based JSCC systems.

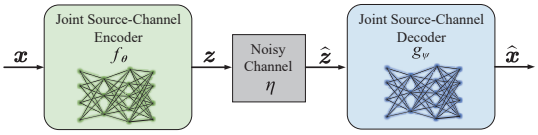


Fig. 1. Block diagram of the image transmission system for the proposed framework.

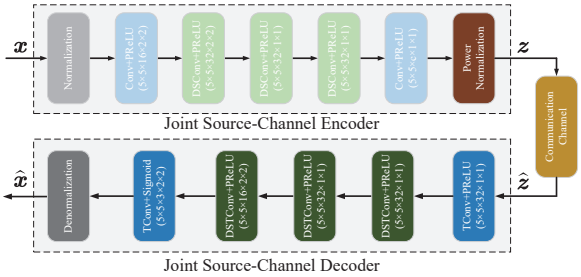


Fig. 2. Architecture of the proposed DSC-JSCC-60 (E2D2) model.

III. Proposed Framework

In this section, we present the architectures of DSC-JSCC variants based on the selective replacement strategy.

A. Lightweight DSC-JSCC Architecture

In this work, we propose a lightweight DL-based JSCC framework, termed DSC-JSCC, which redesigns the network architecture using DSConv and DSTConv layers. The framework adopts a selective replacement strategy, enabling flexible replacement of Conv layers with DSConv layers at different ratios and positions. Specifically, Conv layers in the encoder are replaced with DSConv layers, while TConv layers in the decoder are replaced with DSTConv layers. The architecture of DSC-JSCC-60 (E2D2), a representative variant of the proposed framework, is illustrated in Fig. 2. Each DSConv layer consists of a depthwise convolution for spatial feature extraction followed by a pointwise convolution for channel mixing. Each DSTConv layer consists of a transposed depthwise convolution for spatial upsampling followed by a pointwise convolution for channel mixing.

The normalization layer scales the pixel values of the encoder input $\boldsymbol{x}$ from [0, 255] to [0, 1] to mitigate potential exploding gradients. As shown in Fig. 2, the Conv and DSConv layers in the joint source-channel encoder extract features from the normalized input. Compared with Conv layers, DSConv layers significantly reduce computational complexity while maintaining efficient feature extraction. The extracted features are transformed into the complex-valued vector $\boldsymbol{z}$ using a

power normalization layer to satisfy the average transmit power constraint. Note that we can adjust the number of channels c to vary the bandwidth compression ratio.

The resulting vector $\boldsymbol{z}$ is transmitted over the wireless channel as defined in (2), yielding the corrupted vector $\hat{\boldsymbol{z}}$. The corrupted vector is then reshaped into a real-valued matrix and processed by the joint source-channel decoder. In the decoder, TConv and DSTConv layers extract features from the reshaped matrix. Similar to DSConv layers, DSTConv layers significantly reduce computational complexity while maintaining efficient feature extraction. Finally, the denormalization layer scales the decoder output to the [0, 255] range, producing the final reconstructed image $\hat{\boldsymbol{x}}$.

The proposed framework consists of two phases: the offline training phase and the online reconstruction phase. During the offline training phase, the encoder and the decoder of the proposed framework are optimized end-to-end by minimizing the mean squared error (MSE) loss. During the online reconstruction phase, a test image $\boldsymbol{x}$ is fed into the trained framework to obtain the reconstructed image:

$$\hat{\boldsymbol{x}} = g_{\boldsymbol{\psi}}(\hat{\boldsymbol{z}}) = g_{\boldsymbol{\psi}}\big(\eta(f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho), \sigma^2)\big). \tag{4}$$

Table I. Architectures of DSC-JSCC variants with different replacement ratios.

Model	Encoder	Decoder
DSC-JSCC-20	DSConv (Layer 1) Conv (Layers 2–5)	DSTConv (Layer 1) TConv (Layers 2–5)
DSC-JSCC-40	DSConv (Layers 1–2) Conv (Layers 3–5)	DSTConv (Layers 1–2) TConv (Layers 3–5)
DSC-JSCC-60 (E1D1)	DSConv (Layers 1–3) Conv (Layers 4–5)	DSTConv (Layers 1–3) TConv (Layers 4–5)
DSC-JSCC-80	DSConv (Layers 1–4) Conv (Layer 5)	DSTConv (Layers 1–4) TConv (Layer 5)
DSC-JSCC-100	DSConv (Layers 1–5)	DSTConv (Layers 1–5)

B. Layer Replacement at Different Ratios

The proposed DSC-JSCC framework adopts a selective replacement strategy. Based on this strategy, various DSC-JSCC variants are obtained by replacing different ratios of Conv layers in the encoder and TConv layers in the decoder with DSConv and DSTConv layers, respectively, while keeping all other settings unchanged. Varying replacement ratios lead to different levels of model compression, and the structures of these ratio-based variants are summarized in Table I. To evaluate the impact of different replacement ratios, the early Conv layers in the encoder and the early TConv layers in the decoder are replaced with their corresponding DSConv and DSTConv layers.

As shown in Table I, DSC-JSCC-X denotes the variant in which $X\%$ of Conv and TConv layers are replaced by DSConv and DSTConv layers, respectively, where $X \in \{20, 40, 60, 80, 100\}$. A higher value of X corresponds to a higher level of model compression, thereby further reducing computational complexity.

C. Layer Replacement at Different Positions

Based on the selective replacement strategy, additional DSC-JSCC variants at a fixed replacement ratio are generated by replacing Conv and TConv layers at different positions in the encoder and the decoder with DSConv and DSTConv layers, respectively. Meanwhile, other settings remain unchanged. Even when layer replacement at different positions is conducted at a fixed ratio, it can still lead to noticeable model compression, potentially affecting reconstruction performance. To evaluate the effect of layer positions, the early, middle, and late Conv layers in the encoder and TConv layers in the decoder are replaced under a 60% replacement ratio.

Table II. Architectures of DSC-JSCC variants with different replacement positions at a 60% replacement ratio.

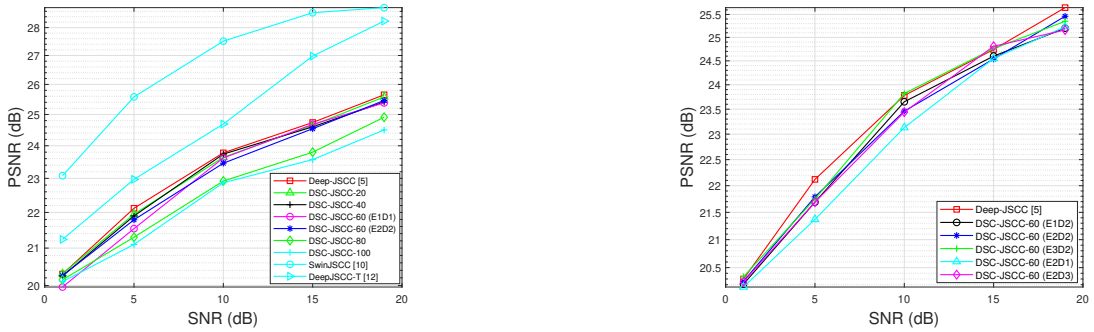
Model	Encoder	Decoder
DSC-JSCC-60 (E2D1)	Conv (Layer 1) DSConv (Layers 2–4) Conv (Layer 5)	DSTConv (Layers 1–3) TConv (Layers 4–5)
DSC-JSCC-60 (E2D2)	Conv (Layer 1) DSConv (Layers 2–4) Conv (Layer 5)	TConv (Layer 1) DSTConv (Layers 2–4) TConv (Layer 5)
DSC-JSCC-60 (E2D3)	Conv (Layer 1) DSConv (Layers 2–4) Conv (Layer 5)	TConv (Layers 1–2) DSTConv (Layers 3–5)
DSC-JSCC-60 (E1D2)	DSConv (Layers 1–3) Conv (Layers 4–5)	TConv (Layer 1) DSTConv (Layers 2–4) TConv (Layer 5)
DSC-JSCC-60 (E3D2)	Conv (Layers 1–2) DSConv (Layers 3–5)	TConv (Layer 1) DSTConv (Layers 2–4) TConv (Layer 5)

Table II summarizes the structures of position-based DSC-JSCC variants. E1, E2, and E3 correspond to the early, middle, and late DSConv layers in the encoder, respectively, while D1, D2, and D3 correspond to the early, middle, and late DSTConv layers in the decoder, respectively. For example, DSC-JSCC-60 (E2D2) denotes the variant where the encoder's middle Conv layers and the decoder's middle TConv layers are replaced by DSConv and DSTConv layers, respectively. Similarly, DSC-JSCC-60 (E1D2) denotes the variant in which the early Conv layers of the encoder and the middle TConv layers of the decoder are replaced by DSConv and DSTConv layers, respectively.

DSC-JSCC variants include models with different replacement ratios (e.g., DSC-JSCC-20 and DSC-JSCC-40) and different replacement positions (e.g., DSC-JSCC-60 (E1D1) and DSC-JSCC-60 (E2D2)). Among these variants of the proposed framework, DSC-JSCC-60 (E2D2) is selected as the default configuration due to its favorable complexity-performance trade-off.

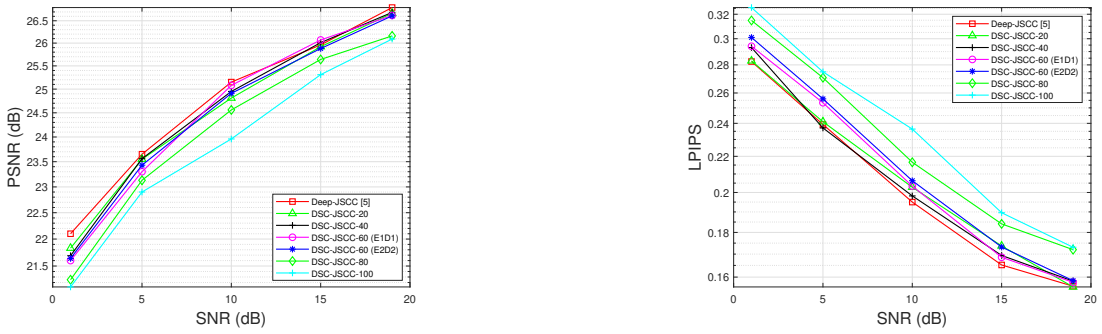
IV. Numerical Results

To evaluate the performance of the DSC-JSCC variants against three baseline models, namely Deep-JSCC [5], Swin-JSCC (without SNR and rate adaptation) [10], and DeepJSCC-T [12], we use the CIFAR-10 [16] and CelebA [17] datasets. The CIFAR-10 dataset contains 50,000 training images and 10,000 test images, where each image has a resolution of $32 \times 32 \times 3$. The CelebA dataset contains 160,000 training images and 10,000 test images, where each image is center-cropped and resized to $64 \times 64 \times 3$. The quality of the reconstructed images is evaluated using the peak signal-to-noise ratio (PSNR) and the learned perceptual image patch similarity (LPIPS) [18]. LPIPS is computed using a pre-trained



(a) (b)

Fig. 3. PSNR comparison of DSC-JSCC variants and baseline JSCC models under different replacement ratios and positions on the CIFAR-10 dataset.



(a) (b)

Fig. 4. PSNR and LPIPS performance of DSC-JSCC variants with different replacement ratios on the CelebA dataset.

VGG network. The Adam optimizer is used for all DSC-JSCC variants, with an initial learning rate of 0.001. The batch size is set to 32, and the models are trained for 100 epochs. Unless otherwise specified, all experiments are conducted under a Rayleigh slow-fading channel with $\rho = 1/12$. For a fair performance comparison, all compared models are trained separately at each SNR value and evaluated under the corresponding SNR condition. Except for SwinJSCC, all other models are implemented in Python 3.8.20 using TensorFlow 2.6.0.

To achieve the desired bandwidth compression ratio ρ, the key parameters k and c are determined as follows. The number of source symbols is given by $d = W \times H \times C$, where W, H, and C denote the width, height, and number of channels of the input image, respectively. The number of transmitted symbols is given by $k = \lfloor \rho \times d \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor operator. The number of output channels c is computed as $c = \lfloor 2k/\left(\bar{H} \times \bar{W}\right) \rfloor$, where $\bar{H} \times \bar{W}$ represents the spatial dimensions of the feature map produced by the final Conv layer in the encoder.

Fig. 3(a) presents the PSNR results on the CIFAR-10 dataset. The PSNR of the DSC-JSCC variants generally decreases as the replacement ratio increases. Nevertheless, both DSC-JSCC-20 and DSC-JSCC-40 achieve PSNR comparable to that of Deep-JSCC while exhibiting lower computational complexity. This indicates that replacing standard Conv layers with DSConv layers improves the computational efficiency of the model. DSC-JSCC-60 (E2D2) exhibits slightly lower performance than DSC-JSCC-20 while further reducing computational complexity. Furthermore, DSC-JSCC-60 (E2D2) achieves performance comparable to that of DSC-JSCC-60 (E1D1) with lower complexity. Although DSC-JSCC-80 achieves lower complexity than DSC-JSCC-60 (E2D2), it suffers from noticeable performance degradation. SwinJSCC and DeepJSCC-T achieve significantly higher PSNR than DSC-JSCC-60 (E2D2). However, they incur substantially higher computational complexity, making them less suitable for deployment on resource-constrained edge devices.

As shown in Fig. 3(b), different replacement positions lead to different PSNR values for the DSC-JSCC-60 variants. The performance gap between the DSC-JSCC variants and Deep-JSCC is small, suggesting that lightweight models are preferable for resource-constrained edge deployment. DSC-JSCC-60 (E2D2) achieves performance comparable to that of the other variants with a 60% replacement ratio. However, it exhibits the lowest computational complexity among them. This is because the computational cost is mainly concentrated in the intermediate layers of the encoder and the decoder. Thus, DSC-JSCC-60 (E2D2) achieves an effective trade-off between computational efficiency and reconstruction performance, making it well-suited for resource-limited applications.

We further evaluate the impact of different replacement ratios on the PSNR and LPIPS performance of the DSC-JSCC variants on the CelebA dataset. As shown in Fig. 4(a), DSC-JSCC-20, DSC-JSCC-40, and both DSC-JSCC-60 variants achieve comparable PSNR. Among these models, DSC-JSCC-60 (E2D2) exhibits the lowest computational complexity

and thus provides the best complexity-performance trade-off. Although DSC-JSCC-80 and DSC-JSCC-100 further reduce complexity, they suffer from noticeable performance degradation compared with other DSC-JSCC variants. A similar trend is observed in Fig. 4(b), indicating that DSC-JSCC-60 (E2D2) also achieves a favorable trade-off between LPIPS and computational complexity. Overall, DSC-JSCC-60 (E2D2) achieves a favorable complexity-performance trade-off on the CelebA dataset.

Table III. Computational complexity comparison of DSC-JSCC variants against existing DL-based JSCC models.

Model	Parameters (K)	FLOPs (M)
SwinJSCC	6200.2	1249.3
DeepJSCC-T	4670.2	206.8
Deep-JSCC	164.4	19.3
DSC-JSCC-20	157.4	18.0
DSC-JSCC-40	121.8	13.5
DSC-JSCC-60 (E1D1)	74.3	7.4
DSC-JSCC-60 (E2D1)	51.6	4.9
DSC-JSCC-60 (E2D2)	**46.1**	**4.7**
DSC-JSCC-60 (E2D3)	69.1	8.1
DSC-JSCC-60 (E1D2)	68.8	7.2
DSC-JSCC-60 (E3D2)	52.7	5.5
DSC-JSCC-80	39.1	3.4
DSC-JSCC-100	**33.1**	**3.0**

We evaluate the number of parameters and floating-point operations (FLOPs) of the DSC-JSCC variants and baseline models on the CIFAR-10 dataset with a batch size of 1. As shown in Table III, SwinJSCC achieves the highest computational complexity among all models, while DeepJSCC-T incurs much higher complexity than Deep-JSCC. Both methods exhibit a less favorable complexity-performance trade-off for resource-constrained edge deployment. DSC-JSCC-60 (E1D1) reduces the number of parameters by 54.8% and the FLOPs by 61.7% compared to Deep-JSCC, thereby improving computational efficiency. These reductions result from replacing 60% of the Conv and TConv layers with their DSConv and DSTConv counterparts, respectively. This replacement reduces the computational cost by decomposing standard convolutions into depthwise and pointwise convolutions [19]. Among the DSC-JSCC-60 variants with a fixed replacement ratio of 60%, DSC-JSCC-60 (E2D2) achieves the lowest computational complexity. In particular, it reduces the number of parameters and FLOPs by 38.0% and 36.5%, respectively, compared to DSC-JSCC-60 (E1D1). Although both DSC-JSCC-80 and DSC-JSCC-100 achieve lower computational complexity than DSC-JSCC-60 (E2D2), they suffer from significant performance degradation. Overall, DSC-JSCC-60 (E2D2) achieves a favorable complexity-performance trade-off, making it attractive for practical deployment in resource-constrained scenarios.

V. Conclusion

In this letter, we proposed a configurable lightweight DL-based JSCC framework. It employs a selective replacement strategy to enable flexible Conv-to-DSConv replacement at different replacement ratios and positions. Varying the replacement ratio enables controllable model complexity, while different replacement positions affect reconstruction performance under a fixed replacement ratio. These results reveal the relationship between layer-wise computational redundancy and performance in DL-based JSCC systems, providing practical design guidelines for lightweight DL-based JSCC systems. Experimental results demonstrate that the proposed framework achieves substantial parameter reduction with only slight performance degradation, thereby achieving a favorable complexity-performance trade-off and making it well suited for deployment on resource-constrained edge devices.

References

[1] H. Du, J. Wang, D. Niyato, J. Kang, Z. Xiong, and D. I. Kim, "AI-Generated incentive mechanism and full-duplex semantic communications for information sharing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 9, pp. 2981-2997, Sep. 2023.

[2] X. Niu, L. Tan, J. Wu, W. Yuan, and T. Q. S. Quek, "Multimodal-oriented interactive joint source-channel coding for lightweight semantic communication," *IEEE Trans. Veh. Technol.*, vol. 74, no. 10, pp. 16516-16520, Oct. 2025.

[3] J. Tu, X. Liu, Y. Wei, F. Zhou, and S. Ma, "Lightweight semantic communication for wireless image transmission," *IEEE Wireless Commun. Lett.*, vol. 14, no. 12, pp. 4132-4136, Dec. 2025.

[4] H. Wu, Y. Shao, C. Bian, K. Mikolajczyk, and D. Gündüz, "Deep joint source-channel coding for adaptive image transmission over MIMO channels," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 15002-15017, Oct. 2024.

[5] 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.

[6] E. Erdemir, T. -Y. Tung, P. L. Dragotti, and D. Gündüz, "Generative joint source-channel coding for semantic image transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 8, pp. 2645-2657, Aug. 2023.

[7] E. Grassucci, G. Pignata, G. Cicchetti, and D. Comminiello, "Lightweight diffusion models fo rresource-constrained semantic communication," *IEEE Wireless Commun. Lett.*, vol. 14, no. 9, pp. 2743-2747, Sept. 2025.

[8] J. Shi, Q. Zhang, W. Zeng, S. Li, and Z. Qin, "Secure transmission in wireless semantic communications with adversarial training," *IEEE Commun. Lett.*, vol. 29, no. 3, pp. 487-491, Mar. 2025.

[9] T. Wu et al., "MambaJSCC: Adaptive deep joint source-channel coding with generalized state space model," *IEEE Trans. Wireless Commun.*, vol. 25, pp. 9264-9279, 2026.

[10] K. Yang, S. Wang, J. Dai, X. Qin, K. Niu, and P. Zhang, "SwinJSCC: Taming Swin Transformer for deep joint source-channel coding," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 1, pp. 90-104, Feb. 2025.

[11] G. Chen et al., "Lightweight and robust wireless semantic communications," *IEEE Commun. Lett.*, vol. 28, no. 11, pp. 2633-2637, Nov. 2024.

[12] Y. Sun, J. Wang, L. Wei, H. Chen, S. Dang, and X. Li, "Lightweight and adaptive deep coding for wireless image transmission in semantic Communication," *IEEE Access*, vol. 13, pp. 158285-158301, 2025.

[13] M. A. Jarrahi, E. Bourtsoulatze, and V. Abolghasemi, "Task-oriented JSCC with adaptive deep compressed sensing," *IEEE Wireless Commun. Lett.*, to be published.

[14] T. Guo, S. Gu, Y. Wu, Q. Zhang, and W. Xiang, "A novel lightweight deep joint source-channel coding framework: Using 1D-CNN for SNR and compression rate adaptation," in *Proc. IEEE/CIC Int. Conf. Commun. China*, Aug. 2025, pp. 1-6.

[15] X. Yu, D. Li, N. Zhang, and X. Shen, "A novel lightweight joint source-channel coding design in semantic communications," *IEEE Internet Things J.*, vol. 12, no. 11, pp. 18447-18450, Jun. 2025.

[16] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 3730-3738.

[18] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586-595.

[19] G. Lu, W. Zhang, and Z. Wang, "Optimizing depthwise separable convolution operations on GPUs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 1, pp. 70-87, 1 Jan. 2022.