Deep Joint Source-Channel Coding for Adaptive Image Transmission over MIMO Channels

Haotian Wu, *Graduate Student Member, IEEE*, Yulin Shao, *Member, IEEE*, Chenghong Bian, Krystian Mikolajczyk, *Senior Member, IEEE*, Deniz Gündüz, *Fellow, IEEE*

***Abstract*—We introduce a vision transformer (ViT)-based deep joint source and channel coding (DeepJSCC) scheme for wireless image transmission over multiple-input multiple-output (MIMO) channels, called DeepJSCC-MIMO. We employ DeepJSCC-MIMO in both open-loop and closed-loop MIMO systems. The novel DeepJSCC-MIMO architecture surpasses the classical separation-based benchmarks, while exhibiting robustness to channel estimation errors and flexibility in adapting to diverse channel conditions and antenna configurations without requiring retraining. Specifically, by harnessing the self-attention mechanism of the ViT, DeepJSCC-MIMO intelligently learns feature mapping and power allocation strategies tailored to the unique characteristics of the source image and prevailing channel conditions. Extensive numerical experiments validate the significant improvements in both distortion quality and perceptual quality achieved by DeepJSCC-MIMO for both open-loop and closed-loop MIMO systems across a wide range of scenarios. Moreover, DeepJSCC-MIMO exhibits robustness to varying channel conditions, channel estimation errors, and different antenna numbers, making it an appealing technology for emerging semantic communication systems.**

***Index Terms*—Joint source-channel coding, MIMO, semantic communication, attention mechanism, image transmission.**

I. Introduction

The exponential growth of wireless multimedia applications, encompassing augmented reality, virtual reality, real-time streaming, and edge intelligence, has significantly heightened the demand for efficient wireless transmission of image/video signals, particularly under strict delay constraints. Consequently, this has sparked a notable surge in research on developing optimized image communication systems over wireless channels [1]–[10].

H. Wu, C. Bian, K. Mikolajczyk, and D. Gündüz are with the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K. (e-mails: {haotian.wu17, c.bian22, k.mikolajczyk, d.gunduz}@imperial.ac.uk). Y. Shao is with the State Key Laboratory of Internet of Things for Smart City and the Department of Electrical and Computer Engineering, University of Macau, Macau S.A.R (e-mail: ylshao@um.edu.mo). He is also a visiting researcher at the Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K.

This work received funding from the UKRI for the projects AIR (ERC-Consolidator Grant, EP/X030806/1), SONATA (EPSRC-EP/W035960/1), and the SNS JU project 6G-GOALS under the EU's Horizon program (grant agreement No. 101139232). The work of Y. Shao was supported in part by the Start-up Research Grant under Grant SRG2023-00038-IOTSC, and in part by the Science and Technology Development Fund, Macao SAR, under Grant 0068/2023/RIB3.

For the purpose of open access, the authors have applied a Creative Commons Attribution (CCBY) license to any Author Accepted Manuscript version arising from this submission.

This paper was presented in part at the IEEE International Conference on Communications (ICC) in May 2023.

The conventional solution, motivated by Shannon's separation theorem, is to independently design source and channel coding, which is optimal in the asymptotic limit of infinite block length for ergodic source and channel distributions [11]. However, the separation-based approach is known to be sub-optimal in the practical finite block length regime, which is becoming increasingly relevant in emerging applications involving Internet-of-things and edge intelligence. Despite its known benefits, designing practical joint source-channel coding (JSCC) schemes has been an ongoing challenge for many decades. Most existing studies on JSCC either focus on theoretical scenarios, e.g., Gaussian or Gauss-Markov sources [12]–[15], or jointly optimize parameters of separate source and channel codes [16]–[19], as there is a lack of practical JSCC schemes and theoretical performance bounds on the achievable mean squared error distortion for lossy JSCC in finite block lengths [20], particularly when dealing with practical sources such as images. On the other hand, significant progress has been made in the recent years in designing deep learning-based JSCC (DeepJSCC) schemes thanks to the introduction of deep neural networks (DNNs) [1]–[4], which allow to directly map the input source signal to channel symbols, and vice versa at the decoder. This approach removes the necessity of explicit source compression or error correction, or the usage of bits as a common unit of representation of signals. Instead, the DeepJSCC approach is optimized to directly generate channel symbols with the goal of enhancing the overall transmission quality across diverse source and channel distributions. This is achieved through an end-to-end training methodology that refrains from imposing constraints on the distribution of the source or the channel.

The first DeepJSCC scheme for image transmission was presented in [1], which was shown to outperform the concatenation of the better portable graphics (BPG) image compression codec with low-density parity-check (LDPC) codes. It was later extended to adaptive channel bandwidth scenario in [5] and [7], and to the transmission over multi-path fading channels in [2] and [21]. Its feasibility in practical systems has also been evaluated, including building prototypes [8], [22], or its implementation with constrained channel inputs [23], or under peak-to-average power ratio constraints [24]. However, existing DeepJSCC schemes solely consider single-antenna transmitters and receivers. While there is a growing literature successfully employing DNNs for various multiple-input multiple-output (MIMO)-related tasks in wireless communications, such as detection, channel estimation, or beamforming [25]–[27], no previous work has so far applied DeepJSCC

TABLE I
OVERVIEW OF ALTERNATIVE END-TO-END MIMO COMMUNICATION PIPELINES.

Potential Schemes	End-to-end AE [30]	DAE benchmarks [31]	SVD-DAE [32]	VST [36]	Our work
Input source	One-hot input	One-hot input	Bits	Images	Images
CSIR scenarios	✓	✓	✗	✓	✓
CSIT scenarios	✓	✓	✓	✗	✓
Antenna flexibility	✗	✗	✗	✗	✓
Channel adaptability	✗	✗	✓	✓	✓
Channel robustness	✓	✗	✗	✗	✓
Metrics	Bit error rate	Block error rate	Bit error rate	PSNR	PSNR & LPIPS
Main baseline	SVD+QPSK+ equal power	STBC*/SVD + GS+ bit/power loading	Plain-DAE	BPG+LDPC	BPG/DL-based compression+Capacity
Results	DAE outperforms/matches the baseline for CSIR; DAE outperforms baseline at some specific SNRs for CSIT.†			Method outperforms the baseline	Method outperforms the baseline
Main insights	Pioneered the DAE-based MIMO	Benchmark end-to-end MIMO systems	Evaluate the model-driven DAE	Semantic coding over CQI ‡	First unified DeepJSCC scheme over MIMO

∗ STBC represents space-time block codes; GS represents geometrically-shaped signal constellations; QPSK is quadrature phase shift keying modulation; SVD is singular value decomposition precoding operation.
† For the CSIT, DAE-based models can outperform baseline at some specific SNRs, but cannot perform better than properly chosen benchmarks [31].
‡ CQI represents channel quality indicators, e.g., the SNR value, which needs to be assumed accessible for the transmitter in the open-loop MIMO system.

to the more challenging MIMO scenario, with the exception of our initial results in [28]. JSCC over MIMO channels was studied in [29] from a theoretical perspective, providing theoretical bounds for the distortion exponent as a function of the bandwidth ratio considering Gaussian sources. It is challenging to design a practical JSCC scheme for MIMO channels, where the model needs to learn how to map the input signal to multiple antennas, and then to recover the input from multiple channel outputs, where each input-output antenna pair experiences a distinct channel condition.

Advantages of DeepJSCC and end-to-end MIMO schemes in practical MIMO transmission scenarios have prompted the investigation of DeepJSCC over MIMO systems. The first deep autoencoder (DAE) based end-to-end MIMO communication method was introduced in [30]. In the subsequent work [31], the authors systematically establish the symbol error rate benchmarks for MIMO channels by evaluating several DAE-based models with channel state information (CSI). A singular-value decomposition (SVD) based DAE is proposed in [32] to achieve competitive performance for the closed-loop MIMO system with CSI at the transmitter (CSIT). These works indicate the potential of end-to-end MIMO communication schemes to enhance transmission quality when coupled with advanced deep learning (DL) technologies. However, we remark that the aforementioned works focus on the transmission of bits/one-hot vectors for specific signal-to-noise ratio (SNR) values, and do not consider the contextual features of the input signals, also referred to as 'semantics' in the literature [33], [34], and the model's adaptability to varying channel conditions, as outlined in Table I. Two recent studies that are more relevant to this work are [35] and VST [36], where [35] theoretically analyzes the excess distortion exponent for semantic-aware MIMO systems, while the authors in [36] propose a semantically coded transmission scheme utilizing MIMO, assuming that CSI is available at the receiver (CSIR).

To the best of our knowledge, prior research has not delved into the specific exploration of adaptive JSCC MIMO systems designed for wireless image transmission, encompassing both scenarios with open-loop MIMO with only receiver-side CSI (CSIR) and closed-loop MIMO with both transmitter and receiver side CSI (CSIT). Such a system would be capable of leveraging the semantics of the source signal and channel conditions simultaneously, and efficiently managing both fluctuating channel conditions and potential channel estimation errors, all within a cohesive framework.

The differences from earlier studies and the ensuing challenges we aim to address in this paper can be outlined across four key points: performance improvement, model generalizability, channel adaptability, and channel robustness. These facets become particularly important to ensure the framework's effectiveness and practicality.

- Performance improvement: Prior DL-based designs for MIMO systems [30]–[32], have mostly focused on symbol detection in a digital communication framework. Moreover, even the state-of-art DL-based solution [31] cannot outperform a properly selected traditional model-driven solution, which underscores the inherent difficulty of learning effective communication strategies in MIMO systems. Another important aspect of the performance is the memory and computational complexity, which further motivates competitive and practical end-to-end architectures.
- Model generalizability: Most of the proposed schemes [32], [36] have been designed specifically for open-loop or closed-loop systems. Moreover, they require retraining to be used in different antenna configurations. The impracticality of customizing distinct models tailored for specific scenarios is apparent, prompting our search for a unified DeepJSCC solution. Specifically, our ambition is to design a unique DeepJSCC scheme which can generalize to both CSIR and CSIT scenarios, and varying antenna configurations.
- Channel adaptability: The adaptability to diverse channel conditions constitutes a pivotal attribute of DL-based communication schemes [3], [32]. The separate training of neural networks for different channel conditions necessitates massive storage space for multiple networks

corresponding to distinct channel conditions. Previous investigations of channel adaptability in DeepJSCC within single input single output (SISO) system focus on incorporating the SNR as an additional network input [3]. However, transferring these approaches to MIMO channels remains challenging, where the model needs to simultaneously account for the coupled channel inputs and the channel matrix. Consequently, we employ model-driven techniques to disentangle the channels and introduce an innovative paradigm, integrating the self-attention mechanism and CSI heatmap, to learn and adapt to the channel conditions effectively.

- Channel Robustness: CSIR and CSIT are typically acquired through channel estimation and feedback, respectively, which can introduce estimation errors that can lead to reduction in the reliability of the transmitted information [37]. Authors in [38] demonstrated that, in the presence of imperfect CSIT and minimal noise, MIMO performance can be enhanced through power allocation compared to systems without CSIT. However, an increase in noise level can significantly deteriorate the performance. Meanwhile, the joint design of the system with channel estimation errors enables model robustness against channel estimation errors [27], [30]. This observation motivates us to incorporate imperfect CSI into the proposed DeepJSCC framework.

This paper presents an innovative DeepJSCC scheme meticulously tailored for MIMO image transmission to address the above practical concerns. Our approach introduces a unified vision transformer (ViT)-based DeepJSCC scheme, named DeepJSCC-MIMO, which accommodates both scenarios with CSIR and CSIT. Inspired by the attention mechanism in developing flexible communication schemes [3], [21], [39]–[41], we leverage the self-attention mechanism inherent in ViT for adaptive wireless image transmission. Specifically, we represent the channel conditions with a channel heatmap and adapt the JSCC encoding and decoding parameters according to this heatmap. Our approach can learn global attention parameters between the source image and the channel conditions across all the intermediate layers of the DeepJSCC encoder and decoder. Intuitively, we expect this design to simultaneously learn channel symbol mapping and power allocation, considering different channel conditions, channel estimation errors, and different antenna number configurations. Therefore, unlike most existing literature on DNN-aided MIMO communications, we eliminate the need for training separate DNN models for different transceiver settings and channel conditions. The proposed DeepJSCC-MIMO is a practical and unified scheme with robustness to channel estimation errors.

Our main contributions can be summarized as follows:

- This paper presents the first DeepJSCC scheme over the MIMO system for image transmission. The key innovation lies in utilizing a ViT-based model, which adeptly exploits both the semantic features of the image and the CSI through a self-attention mechanism. This design of jointly leveraging attention over both the source and channel aspects can be readily applied to other emerging communication systems concerning semantic sources.
- DeepJSCC-MIMO is a comprehensive and versatile solution that can be seamlessly applied to both open-loop and closed-loop MIMO systems. Importantly, DeepJSCC-MIMO demonstrates remarkable flexibility and adaptability by efficiently accommodating diverse channel conditions and antenna number configurations. This is achieved without the necessity for retraining, thanks to the utilization of the channel heatmap and the self-attention mechanism. Furthermore, the DeepJSCC-MIMO scheme exhibits significant resilience to channel estimation errors.
- Extensive numerical evaluations validate the superiority of the proposed model, showcasing significant enhancements in both distortion quality and perceptual quality across a wide range of channel conditions and bandwidth ratios when compared to traditional separate source and channel coding schemes.

II. System Model

We consider an $M \times M$ MIMO system, where an M-antenna transmitter aims to deliver an image $\boldsymbol{S} \in \mathbb{R}^{h\times w\times 3}$ to an M-antenna receiver (h and w denote the height and width of the image, while 3 refers to the color channels R, G, and B). The transmitter encodes the image into a vector of channel symbols $\boldsymbol{X} \in \mathbb{C}^{M\times k}$, where k denotes the number of channel uses dedicated to transmitting one image. We define the *bandwidth ratio* as $R \triangleq k/n$, representing the average number of available channel symbols per source dimension, where $n = 3hw$ is the number of source symbols. The transmitted signal $\boldsymbol{X}$ is subject to a power constraint P_s as:

$$\frac{1}{Mk}\|\boldsymbol{X}\|_F^2 \le P_s, \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, and we set $P_s = 1$ without loss of generality.

The channel model can be written as:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{W}, \tag{2}$$

where $\boldsymbol{X} \in \mathbb{C}^{M\times k}$ and $\boldsymbol{Y} \in \mathbb{C}^{M\times k}$ denote the channel input and output matrices, respectively, while $\boldsymbol{W} \in \mathbb{C}^{M\times k}$ is the additive white Gaussian noise (AWGN) term, whose entries $W[i,j] \sim \mathcal{CN}(0,\sigma_w^2)$ follow an independent and identically distributed (i.i.d.) complex Gaussian distribution with zero mean and variance σ_w^2. The entries of the channel gain matrix $\boldsymbol{H} \in \mathbb{C}^{M\times M}$ also follow a complex Gaussian distribution with zero mean and variance σ_h^2, i.e., $\boldsymbol{H}[i,j] \sim \mathcal{CN}(0,\sigma_h^2)$. We consider a block-fading channel model, in which the channel matrix $\boldsymbol{H}$ remains constant for k channel uses, corresponding to the transmission of one image, and takes an independent realization in the next block.

Given the channel output $\boldsymbol{Y}$, the receiver reconstructs the source image as $\hat{\boldsymbol{S}} \in \mathbb{R}^{h\times w\times 3}$. The reconstruction quality is quantified through the peak signal-to-noise ratio (PSNR) and the learned perceptual image patch similarity (LPIPS) metrics [42]. The PSNR metric serves as an indicator of image distortion at the per-pixel level, defined as:

$$\mathrm{PSNR} \triangleq 10\log_{10}\frac{\|\boldsymbol{S}\|_\infty^2}{\mathrm{MSE}(\boldsymbol{S},\hat{\boldsymbol{S}})}\ (\mathrm{dB}), \tag{3}$$

where $\|\cdot\|_\infty$ is the infinity norm and $\text{MSE}(\boldsymbol{S},\hat{\boldsymbol{S}}) \triangleq \frac{1}{3hw}\|\boldsymbol{S}-\hat{\boldsymbol{S}}\|_2^2$ is the mean squared error (MSE) between $\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$. LPIPS, on the other hand, computes the dissimilarity between the feature vectors of the input and reconstructed images, where a lower LPIPS score implies a better perceptual quality, defined as:

$$\text{LPIPS}(\boldsymbol{S},\hat{\boldsymbol{S}}) \triangleq \sum_L \frac{1}{H_l W_l}\sum_{h,w}\|\boldsymbol{w_l}\odot(\boldsymbol{y^l}-\hat{\boldsymbol{y}}^{\boldsymbol{l}})\|_2^2, \quad (4)$$

where $\boldsymbol{y^l}, \hat{\boldsymbol{y}}^{\boldsymbol{l}} \in \mathbb{R}^{H_l\times W_l\times C_l}$ are the intermediate features derived from the l-th layer of the given network (VGG in our case) composed of a total of L layers. H_l, W_l, and C_l are the intermediate feature map's height, width, and channel dimensions, $\boldsymbol{w^l} \in \mathbb{R}^{C_l}$ is the weight vector, and $\odot$ is the channel-wise feature multiplication operation [42].

As shown in Fig. 1, there can be two approaches to solve this problem, i.e., the traditional separate source and channel coding scheme and the JSCC scheme. This paper considers both the CSIR (open-loop MIMO) and CSIT (closed-loop MIMO) scenarios for each transmission scheme.

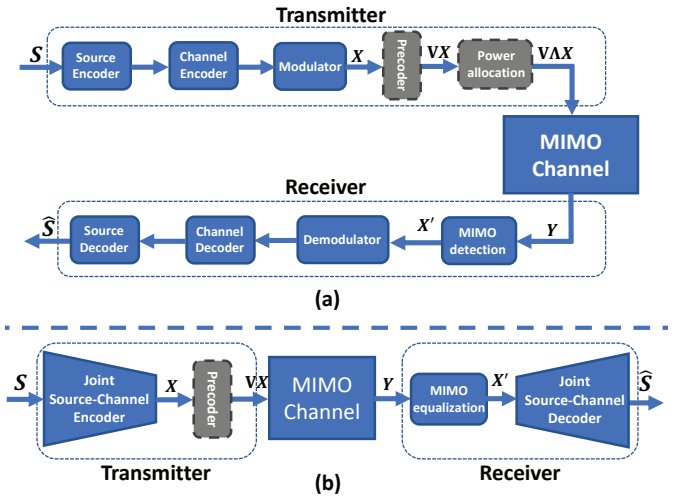


Fig. 1. Block diagram of the MIMO image transmission system: (a) conventional separate source-channel coding scheme and (b) DeepJSCC scheme, where the gray blocks with dashed lines are the additional operations for the closed-loop MIMO system.

A. Open-loop MIMO with CSIR

In an open-loop MIMO system, CSI is only available at the receiver, enabling it to equalize the transmitted signals and subsequently decode the image.

1) Separate source and channel coding scheme: In an open-loop MIMO system with separate source and channel coders, the transmitter follows a sequential process, applying source coding, channel coding, and the modulation in order. We note that, due to the lack of CSI, the transmitter has to choose the compression and channel coding rates and the constellation independently of the current channel realization.

Given the received signal $\boldsymbol{Y}$ and the CSI, the receiver decodes the source signal as $\hat{\boldsymbol{S}}$ by sequentially performing the MIMO detection, demodulation, channel decoding, and source decoding operations.

2) DeepJSCC scheme: Different from the traditional separation-based scheme, DL-based JSCC scheme uses a joint source-channel encoder at the transmitter, denoted as $f_{\boldsymbol{\theta}}$, to directly map the source signal $\boldsymbol{S}$ into the channel input matrix $\boldsymbol{X}$. The encoder $f_{\boldsymbol{\theta}}$ is trained to achieve the best end-to-end performance across diverse channel realizations in an average sense since the transmitter does not have access to CSI. This process can be represented as:

$$\boldsymbol{X} = f_{\boldsymbol{\theta}}(\boldsymbol{S}). \quad (5)$$

At the receiver, a MIMO equalization algorithm, such as zero-forcing (ZF) or minimum mean square error (MMSE), is employed first to exploit the CSI to decouple the entangled signal $\boldsymbol{Y}$ as $\boldsymbol{X'}$. Subsequently, a JSCC decoder, referred to as $f_{\boldsymbol{\phi}}$, is employed to reconstruct the source signal based on the available CSI represented by $(\boldsymbol{H}, \sigma_w^2)$, along with $\boldsymbol{X'}$. This can be expressed as:

$$\hat{\boldsymbol{S}} = f_{\boldsymbol{\phi}}(\boldsymbol{X'}, \boldsymbol{H}, \sigma_w^2). \quad (6)$$

An alternative approach is to directly learn the decoding process using the CSI without explicitly performing channel equalization. This involves training the model to minimize the distortion in an end-to-end manner. However, in order to achieve better transmission quality, we adopt a model-driven approach that explicitly performs channel equalization prior to the decoding process, leveraging the domain knowledge to simplify the learning process, similarly to [2], [27].

B. Closed-loop MIMO with CSIT

Within a closed-loop MIMO system, the CSI is accessible to both the transmitter and receiver, which allows them to apply pre-coding and power allocation at the transmitter, and MIMO equalization at the receiver, thereby improving the image transmission quality.

1) Separate source and channel coding scheme: The transmitter sequentially performs source coding, channel coding, and modulation to generate the channel input matrix $\boldsymbol{X}$, the elements of which are constellations with average power normalized to 1. Additional operations at the transmitter with CSIT are precoding and power allocation, which can boost the communication rate. Specifically, given the CSI, we first decompose the channel matrix $\boldsymbol{H}$ by singular-value decomposition (SVD), yielding $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{H}}$, where $\boldsymbol{U} \in \mathbb{C}^{M\times M}$ and $\boldsymbol{V} \in \mathbb{C}^{M\times M}$ are unitary matrices, $\boldsymbol{V}^H$ is the conjugate transpose of $\boldsymbol{V}$, and $\boldsymbol{\Sigma}$ is a diagonal matrix whose singular values are in descending order. We denote $\boldsymbol{\Sigma}$ by $\text{diag}(s_1, s_2, \ldots, s_M)$, where $s_1 \geq s_2 \geq \cdots \geq s_M$.

We denote the power allocation matrix by $\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is diagonal, with its diagonal elements being the power allocation weights for the signal streams of separate antennas. $\boldsymbol{\Lambda}$ can be derived using the standard water-filling algorithm [43]. With power allocation and SVD precoding (precoding $\boldsymbol{X}$ into $\boldsymbol{V}\boldsymbol{X}$), Eqn. (2) can be rewritten as:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{X} + \boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{X} + \boldsymbol{W}. \quad (7)$$

Multiplying both sides of Eqn. (7) by $\boldsymbol{U}^H$ (MIMO detection) gives us:

$$\boldsymbol{X'} = \boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{X} + \boldsymbol{U}^H\boldsymbol{W}. \quad (8)$$

SVD-based precoding converts the MIMO channel into a set of parallel subchannels with different SNRs. In particular, the SNR of the i-th subchannel is determined by the i-th singular value of s_i and the i-th power allocation coefficient of $\boldsymbol{\Lambda}$.

Given $\boldsymbol{X'} \in \mathbb{C}^{M\times k}$, the receiver performs demodulation, channel decoding, and source decoding sequentially to reconstruct the source image as $\hat{\boldsymbol{S}}$.

2) DeepJSCC scheme: For the DeepJSCC scheme, we exploit DL technologies to parameterize the encoder and decoder functions, which are trained jointly on an image dataset and the channel model as in Eqn. (2). Let us denote the DeepJSCC encoder and decoder by $f_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{\phi}}$, respectively, where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ denote the network parameters. We have

$$\boldsymbol{X} = f_{\boldsymbol{\theta}}(\boldsymbol{S}, \boldsymbol{H}, \sigma_w^2). \tag{9}$$

Unlike the separate source-channel coding scheme, the transmitter does away with power allocation. Instead, we leverage the DeepJSCC encoder to perform feature extraction and channel symbol mapping, and power allocation, all at once. Intuitively, the DNN is expected to transmit critical features over subchannels with higher SNRs, thereby improving the transmission performance.

We note here that one option is to train the encoder/decoder networks directly, hoping they will learn to exploit the spatial degrees-of-freedom that the MIMO channel provides. We will instead follow the model-driven approach, where we will exploit the SVD and first convert the MIMO channel into subchannels. The received signal can then be written as

$$\boldsymbol{Y} = \boldsymbol{HVX} + \boldsymbol{W} = \boldsymbol{U\Sigma X} + \boldsymbol{W}. \tag{10}$$

To simplify the training, we apply MIMO equalization at the receiver by left multiplying both sides of Eqn. (10) by $\boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{H}$ to obtain

$$\boldsymbol{X'} = \boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{H}\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W'}, \tag{11}$$

where $\boldsymbol{W}' \triangleq \boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{W} \in \mathbb{C}^{M\times k}$ is the equivalent noise term, $\boldsymbol{\Sigma}^{\dagger}$ is the Moore-Penrose inverse of matrix $\boldsymbol{\Sigma}$, and $\boldsymbol{U}^{H}$ is the conjugate transpose of matrix $\boldsymbol{U}$.

Finally, we feed both $\boldsymbol{X'}$ and the CSI into the DeepJSCC decoder to recover the image as

$$\hat{\boldsymbol{S}} = f_{\boldsymbol{\phi}}(\boldsymbol{X'}, \boldsymbol{H}, \sigma_w^2). \tag{12}$$

In summary, our DeepJSCC framework integrates some well-established algorithms, such as SVD-based precoding and equalization operations, from previous model-driven MIMO solutions. The objective is to improve the end-to-end performance of DL-based JSCC approach by exploiting domain knowledge from conventional MIMO design.

III. Proposed Method

This section presents a novel DeepJSCC architecture called DeepJSCC-MIMO, which aims to enable efficient image transmission in MIMO systems. The pipeline of our DeepJSCC-MIMO scheme is depicted in Fig. 2 and outlined in Algorithm 1. In this scheme, DeepJSCC-MIMO employs a pair of ViT-based encoder $f_{\boldsymbol{\theta}}$ and decoder $f_{\boldsymbol{\phi}}$ to encode and decode channel symbols, utilizing the inherent self-attention mechanism

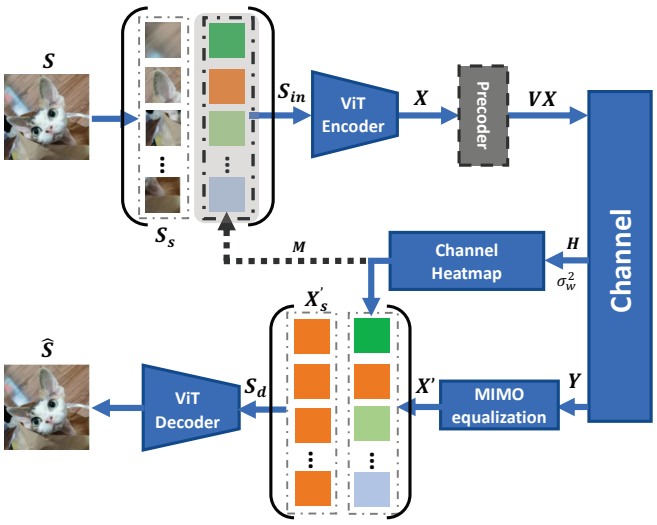


Fig. 2. The pipeline of the DeepJSCC-MIMO scheme, where the source image $\boldsymbol{S}$ is encoded by a ViT-encoder and reconstructed by a ViT-decoder as $\hat{\boldsymbol{S}}$. Precoding operation in dashed line will be performed if the CSI is available at the transmitter, and the CSI $(\boldsymbol{H}, \sigma_w^2)$ is fed to the encoder, if available, and decoder in the form of a "heatmap" $\boldsymbol{M}$ to facilitate the JSCC encoding/decoding process.

over both source features and channel conditions. Further elaborations on the inner structures of these ViTs are provided in Fig. 3. In the following, we detail the pipeline of DeepJSCC-MIMO in five main steps: image-to-sequence transformation, channel heatmap construction, ViT encoding, ViT decoding, and the loss function.

A. Image-to-sequence transformation

To construct the input of DeepJSCC-MIMO, we first convert the three-dimensional input image $\boldsymbol{S}$ into a sequence of vectors, denoted by $\boldsymbol{S_s} = Seq(\boldsymbol{S})$. Specifically, given a source image $\boldsymbol{S} \in \mathbb{R}^{h\times w\times 3}$, we divide $\boldsymbol{S}$ into a grid of $p\times p$ patches, and reshape each patch into a vector of dimension $\mathbb{R}^{\frac{3hw}{p^2}}$. In this way, $\boldsymbol{S}$ is converted to $\boldsymbol{S_s} \in \mathbb{R}^{l\times c}$, where $l = p^2$ is the sequence length and $c \triangleq \frac{3hw}{p^2}$ is the dimension of each vector in the sequence.

B. Channel heatmap construction

To enable efficient training, we construct a channel heatmap from CSI, indicating the effective noise variance faced by each channel symbol generated by ViT. Let us define $\boldsymbol{P_n} \in \mathbb{R}^{M\times k}$ as the average power of the equivalent additive noise term $\boldsymbol{W'}$. Specifically, for different scenarios, $\boldsymbol{P_n}$ is defined as:

$$\boldsymbol{P_n} \triangleq \begin{cases} \sigma_w^2(\boldsymbol{H_w} \odot \bar{\boldsymbol{H}}_{\boldsymbol{w}})\boldsymbol{J}_{M\times k}, & \text{MIMO with CSIR} \\ \sigma_w^2\boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{H}\boldsymbol{J}_{M\times k}, & \text{MIMO with CSIT,} \end{cases} \tag{13}$$

where $\odot$ denotes the Hadamard product and $\bar{\boldsymbol{H}}_{\boldsymbol{w}}$ is the element-wise conjugate of matrix $\boldsymbol{H_w}$, which is obtained from the zero-forcing operation as: $\boldsymbol{H_w} \triangleq (\boldsymbol{H^H H})^{-1}\boldsymbol{H^H}$, and $\boldsymbol{J}$ is the matrix of ones.

Then the heatmap $\boldsymbol{M} \in \mathcal{R}^{l\times\frac{2Mk}{l}}$ is constructed as follows:

$$\boldsymbol{M} \triangleq \texttt{reshape}\left(\texttt{concat}\left(\frac{1}{2}\boldsymbol{P_n}, \frac{1}{2}\boldsymbol{P_n}\right)\right), \tag{14}$$

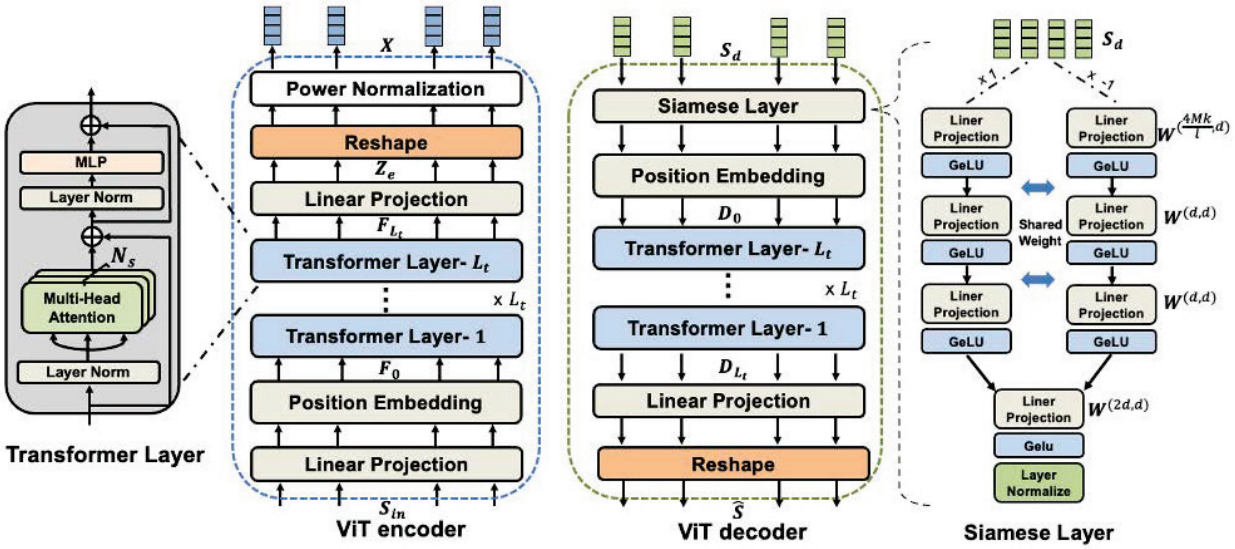


Fig. 3. The architecture of the ViT-based encoder and decoder, where both encoder and decoder comprise linear projection layers, a positional embedding layer, and multiple transformer layers.

where $\texttt{concat}(\cdot)$ and $\texttt{reshape}(\cdot)$ denote the concatenation and reshape operations, respectively. We concatenate two matrices $\frac{1}{2}\boldsymbol{P_n}$ to get the shape of $\mathbb{R}^{M\times 2k}$, and reshape it into $\boldsymbol{M} \in \mathcal{R}^{l\times\frac{2Mk}{l}}$, which represents the equivalent noise term faced by each real encoder output element.

We expect that feeding the CSI in the form of a heatmap will simplify the training process as the model only needs to focus on the 'additive' noise power faced by each channel symbol. As we will show later, this design enables our model to be used under various antenna numbers without retraining.

C. ViT encoding

For the two scenarios, the input sequences to our ViT encoder $\boldsymbol{S_{in}} \in \mathbb{R}^{l\times c_{in}}$ are given by:

$$\boldsymbol{S_{in}} = \begin{cases} \boldsymbol{S_s}, & \text{MIMO with CSIR} \\ \texttt{concat}(\boldsymbol{S_s}, \boldsymbol{M}), & \text{MIMO with CSIT}, \end{cases} \tag{15}$$

where the dimensions of the vectors in these sequences are $c_{in} = c$ and $c_{in} = c + \frac{2Mk}{l}$ for the MIMO with CSIR and MIMO with CSIT scenarios, separately.

The architecture of our encoder is shown in Fig. 3, which mainly consists of linear projection layers, a positional embedding layer, and several transformer layers.

Linear projection and positional embedding: $\boldsymbol{S_{in}}$ is first linearly projected by $\boldsymbol{W_0} \in \mathbb{R}^{c_{in}\times d}$ followed by a positional embedding operation $P_e(\cdot)$ to get the initial input $\boldsymbol{F_0} \in \mathbb{R}^{l\times d}$ for the following transformer layers:

$$\boldsymbol{F_0} = \boldsymbol{S_{in}}\boldsymbol{W_0} + P_e(\boldsymbol{p}), \tag{16}$$

where d is the output dimension of the hidden projection layer. Positional embedding $P_e(\cdot)$ serves the purpose of representing the spatial arrangement of sequences for better performance, where two typical methods are dense layer-based position embedding (DPE) [44] and conditional position embedding (CPE) [45]. In specific, our $P_e(\cdot)$ is implemented with a DPE that employs a dense layer to embed the index vector $\boldsymbol{p}$ of each patch into a d dimensional vector.

Transformer Layer: As shown in Fig. 3, the intermediate feature map $\boldsymbol{F_i}$ is generated by the i-th transformer layer by a multi-head self-attention (MSA) block and a multi-layer perceptron (MLP) layer as:

$$\boldsymbol{F_i} = MSA(\boldsymbol{F_{i-1}}) + MLP(MSA(\boldsymbol{F_{i-1}})), \tag{17}$$

where $\boldsymbol{F_i} \in \mathbb{R}^{l\times d}$ is the output sequence of the i-th transformer layer, GeLU activation and layer normalization operations are applied before each MSA and MLP block.

Each MSA block consists of N_s self-attention (SA) modules with a residual skip, which can be formulated as:

$$MSA(\boldsymbol{F_i}) = \boldsymbol{F_i} + [SA_1(\boldsymbol{F_i}), \cdots, SA_{N_s}(\boldsymbol{F_i})]\boldsymbol{W_i}, \tag{18}$$

where the output of all SA modules $SA(\boldsymbol{F_i}) \in \mathbb{R}^{l\times d_s}$ are concatenated for a linear projection $\boldsymbol{W_i} \in \mathbb{R}^{d_s N_s\times d}$, $d_s = d/N_s$ is the output dimension of each SA operation.

For each SA module, the operations are formulated as:

$$SA(\boldsymbol{F_{l-1}}) = softmax\left(\frac{\boldsymbol{q}\boldsymbol{k}^T}{\sqrt{d}}\right)\boldsymbol{v}, \tag{19}$$

where the $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v} \in \mathbb{R}^{l\times d_s}$ are the query, key, and value vectors generated through three linear projection layers $\boldsymbol{W_q}, \boldsymbol{W_k}, \boldsymbol{W_v} \in \mathbb{R}^{d\times d_s}$ as:

$$\boldsymbol{q} = \boldsymbol{F_{l-1}}\boldsymbol{W_q}, \quad \boldsymbol{k} = \boldsymbol{F_{l-1}}\boldsymbol{W_k}, \quad \boldsymbol{v} = \boldsymbol{F_{l-1}}\boldsymbol{W_v}. \tag{20}$$

Linear projection and power normalization: After L_t transformer layers, we apply a linear projection $\boldsymbol{W_c} \in \mathbb{R}^{d\times\frac{2Mk}{l}}$ to map the output of the transformer layers $\boldsymbol{F_{L_t}}$ to the channel symbols as:

$$\boldsymbol{Z_e} = \boldsymbol{F_{L_t}}\boldsymbol{W_c}, \tag{21}$$

where $\boldsymbol{Z_e} \in \mathbb{R}^{l\times\frac{2Mk}{l}}$ is then reshaped and normalized to satisfy the power constraints to form the complex channel input symbols $\boldsymbol{X} \in \mathbb{C}^{M\times k}$.

D. ViT decoding

Channel equalization: To simplify the decoding process, we perform an equalization operation on the channel output $\boldsymbol{Y}$ to decouple the transmitted data of each antenna as $\boldsymbol{X'} \in \mathbb{C}^{M\times k}$ before feeding the signal into the ViT decoder. We apply different equalization methods for the MIMO systems with CSIR and with CSIT:

$$\boldsymbol{X'} = \begin{cases} \boldsymbol{H_w Y} + f_{res}(\boldsymbol{Y}, \boldsymbol{H}), & \text{MIMO with CSIR} \\ \boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{Y}, & \text{MIMO with CSIT.} \end{cases} \tag{22}$$

To perform MIMO equalization in an open-loop MIMO system with CSIR, we designed a DL-aided equalization method to retrieve the transmitted signal $\boldsymbol{X'}$ from $\boldsymbol{Y}$. Specifically, we add a residual block, denoted as $f_{res}(\cdot)$, to learn the compensation after the zero-forcing (ZF) channel equalization operation based on the received signals and CSI, given as:

$$\boldsymbol{X'} = \boldsymbol{H_w Y} + f_{res}(\boldsymbol{Y}, \boldsymbol{H}) \tag{23}$$

$$= \boldsymbol{X} + \boldsymbol{H_w W} + f_{res}(\boldsymbol{Y}, \boldsymbol{H}), \tag{24}$$

where $\boldsymbol{W'} \triangleq \boldsymbol{H_w W} \in \mathbb{C}^{M\times k}$ is the equivalent additive noise term from the channel. The residual operation $f_{res}(\cdot)$ encompasses two linear layers, parameterized with $\boldsymbol{W_{r1}} \in \mathbb{R}^{(2M^2+2M)\times 128}$ and $\boldsymbol{W_{r2}} \in \mathbb{R}^{128\times 2M}$, and a PReLU activation function, formulated as:

$$f_{res}(\boldsymbol{H}, \boldsymbol{Y}) \triangleq \text{PReLU}(\boldsymbol{f_{res_{in}}} \boldsymbol{W_{r1}}) \boldsymbol{W_{r2}} \tag{25}$$

where the elements of $\boldsymbol{H}$ and $\boldsymbol{Y}$ in each time slot are concatenated to form an input $\boldsymbol{f_{res_{in}}} \triangleq \texttt{concat}(\boldsymbol{H}, \boldsymbol{Y}) \in \mathbb{R}^{2M^2+2M}$ for the residual block. We jointly optimize this residual block with the whole pipeline and expect it to compensate for ZF equalization when faced with channel estimation errors. Note that we can also apply other MIMO equalization methods. However, we have observed that a simple ZF estimation method, together with a learned compensation, is sufficiently effective. Moreover, the ZF method converts the MIMO channel into equivalent parallel sub-channels, which suits our approach of applying the self-attention mechanism over different sub-channels and source signals.

To perform MIMO equalization in a closed-loop MIMO system with CSIT, we perform SVD decomposition and precoding as in Eqn. (10). At the receiver, we perform equalization as in Eqn. (11) to get:

$$\boldsymbol{X'} = \boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{W}, \tag{26}$$

which converts the MIMO channel into a set of parallel subchannels.

In summary, the proposed DeepJSCC-MIMO scheme obtains disentangled channel output $\boldsymbol{X'}$ via channel equalization procedures, which is shown to improve decoding performance, as demonstrated in the appendix of [46].

Given the equalized signal $\boldsymbol{X'}$, which is reshaped into $\boldsymbol{X_s'} \in \mathbb{R}^{l\times \frac{2Mk}{l}}$, and the noise heatmap $\boldsymbol{M}$, a ViT-based decoder $f_{\boldsymbol{\phi}}$ is designed to recover the source image as $\hat{\boldsymbol{S}} = f_{\boldsymbol{\phi}}(\boldsymbol{X_s'}, \boldsymbol{M})$. Our ViT-based decoder consists of a Siamese layer, positional embedding layer, transformer layer, and linear projection layer, elaborated upon below:

Algorithm 1 Algorithm of the DeepJSCC-MIMO scheme

For the transmitter:
Input: $\boldsymbol{S} \in \mathbb{R}^{h\times w\times 3}$, $\boldsymbol{M} \in \mathbb{R}^{l\times \frac{2Mk}{l}}$ (MIMO with CSIT)
Output: $\boldsymbol{X} \in \mathbb{C}^{M\times k}$
$\boldsymbol{S_s} = Seq(\boldsymbol{S}) \in \mathbb{R}^{l\times c}$ ▷ Image Sequentialization
if *MIMO with CSIR* **then**
  $\boldsymbol{S_{in}} = \boldsymbol{S_s}$
  $\boldsymbol{X} = f_\theta(\boldsymbol{S_{in}}) \in \mathbb{C}^{M\times k}$ ▷ ViT encoding
else if *MIMO with CSIT* **then**
  $\boldsymbol{S_{in}} = concat(\boldsymbol{S_s}, \boldsymbol{M})$
  $\boldsymbol{X} = f_\theta(\boldsymbol{S_{in}}) \in \mathbb{C}^{M\times k}$ ▷ ViT encoding
  $\boldsymbol{H} = \boldsymbol{U\Sigma V^H}$ ▷ SVD decomposition
  $\boldsymbol{X} = \boldsymbol{VX}$ ▷ Precoding
end if
Channel model: $\boldsymbol{Y} = \boldsymbol{HX} + \boldsymbol{W}$
For the receiver:
Input: $\boldsymbol{Y} \in \mathbb{C}^{M\times k}$, $\boldsymbol{H} = \boldsymbol{U\Sigma V^H}$, $\boldsymbol{M}$
Output: $\hat{\boldsymbol{S}} \in \mathbb{R}^{h\times w\times 3}$
if *MIMO with CSIR* **then**
  $\boldsymbol{X'} = \boldsymbol{H_w Y} + f_{res}(\boldsymbol{Y}, \boldsymbol{H})$ ▷ DL-aided equalization
else if *MIMO with CSIT* **then**
  $\boldsymbol{X'} = \boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^{\boldsymbol{H}}\boldsymbol{Y}$ ▷ SVD-based equalization
end if
$\boldsymbol{X'} \in \mathbb{C}^{M\times k} \rightarrow \boldsymbol{X_s'} \in \mathbb{R}^{l\times \frac{2MK}{l}}$ ▷ Reshape
$\hat{\boldsymbol{S}} = f_{\boldsymbol{\phi}}(concat(\boldsymbol{X_s'}, \boldsymbol{M}))$ ▷ ViT decoding

Siamese layer and positional embedding: We design a weight-shared Siamese layer, denoted as Siam$(\cdot)$, consisting of several linear projection layers and GeLU activation functions. To form the input of the Siamese layer $\boldsymbol{S_d}$, we concatenate $\boldsymbol{X_s'}$ and $\boldsymbol{M}$ as:

$$\boldsymbol{S_d} = \texttt{concat}(\boldsymbol{X_s'}, \boldsymbol{M}) \in \mathbb{R}^{l\times \frac{4Mk}{l}}. \tag{27}$$

$\boldsymbol{S_d}$ multiplied by 1 and -1 are fed into several linear projection layers and GeLU functions, as illustrated in Fig. 3. In doing so, our networks are tasked with handling the positive and the negative noise realizations through two parallel branches, and subsequently, the resultant features are aggregated by a linear layer to obtain the final output. We expect that these GeLU functions and linear projection layers can learn to truncate excessive noise realizations to bootstrap the performance [39]. To introduce the positional information in the decoding process, we use the same positional embedding layer as in the Eqn. (16) to get the output $\boldsymbol{D_0} \in \mathbb{R}^{l\times d}$:

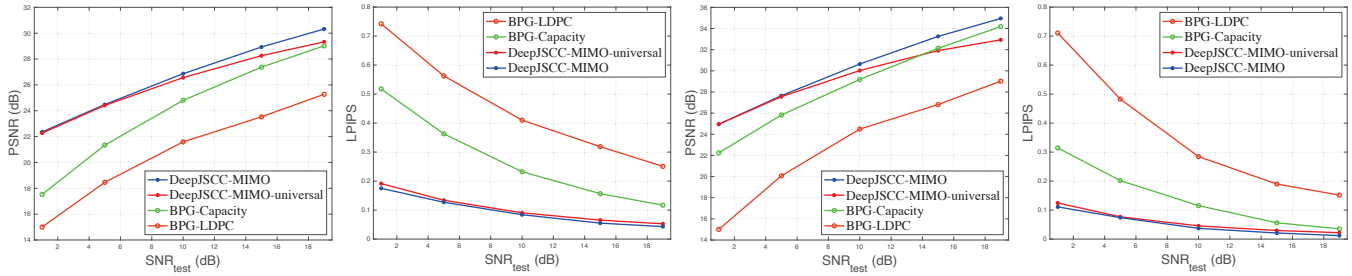
$$\boldsymbol{D_0} = \text{Siam}(\boldsymbol{S_d}) + P_e(\boldsymbol{p}). \tag{28}$$

Transformer layer: After the Siamese layer and positional embedding, $\boldsymbol{D_0}$ is passed through L_t transformer layers, where

$$\boldsymbol{D_l} = MSA(\boldsymbol{D_{l-1}}) + MLP(MSA(\boldsymbol{D_{l-1}})), \tag{29}$$

where $\boldsymbol{D_i} \in \mathbb{R}^{l\times d}$ is the output of the i-th transformer layer at the decoder, and the MSA and MLP blocks share the same structure as those in Eqn. (17).

Linear Projection: Given the output of the L_t-th transformer layer $\boldsymbol{D_{L_t}}$, we apply a linear projection $\boldsymbol{W_{out}} \in \mathbb{R}^{d\times c}$,



(a) $R = 1/24 - PSNR$ (b) $R = 1/24 - LPIPS$ (c) $R = 1/12 - PSNR$ (d) $R = 1/12 - LPIPS$

Fig. 4. Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the MIMO systems with CSIR.

and then reshape the output into a matrix of size $\mathbb{R}^{h\times w\times 3}$ to reconstruct the input image as:

$$\hat{\boldsymbol{S}} = \texttt{reshape}(\boldsymbol{D}_{\boldsymbol{L_t}}\boldsymbol{W}_{\boldsymbol{out}}). \tag{30}$$

E. Loss function

The encoder and decoder are optimized jointly to minimize the loss function

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) = \text{MSE}(\boldsymbol{S},\hat{\boldsymbol{S}}) + \lambda\cdot\text{LPIPS}(\boldsymbol{S},\hat{\boldsymbol{S}}), \tag{31}$$

where we introduced an additional LPIPS term [9] with $\lambda = 0.1$, which is set into 0 when optimizing the PSNR metric. We train the model to search for the optimal parameters $(\boldsymbol{\theta}^*,\boldsymbol{\phi}^*)$ with a minimal $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ as: $(\boldsymbol{\theta}^*,\boldsymbol{\phi}^*) = \arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}}\mathbb{E}\big[\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})\big]$, where the expectation is taken over the image and channel datasets.

IV. Training and Evaluation

This section conducts a set of numerical experiments to evaluate the performance of our DeepJSCC-MIMO in various bandwidth and SNR scenarios. Unless stated otherwise, we consider a 2×2 MIMO system and the CIFAR10 dataset, which has 50000 images with dimensions $3\times 32\times 32$ (color, height, width) as the training dataset and 10000 images as the test dataset. We employ both the practical separation-based codes and theoretical bounds (assuming capacity-achieving channel codes) as benchmarks.

A. Experimental setup

For the practical separation-based coding scheme, we use the BPG codec [47] as well as the state-of-the-art neural compression algorithms [48]–[52] as the source coding method. For channel coding, we consider LDPC codes at rates $(1/2, 2/3, 3/4, 5/6)$ and various constellation sizes. A lower LDPC coding rate can yield improved channel coding results but at the cost of a reduced compression rate with notable performance reduction, especially in very low coding rate regimes and short block length scenarios. In particular, we adopt WiFi (IEEE 802.11n) LDPC code construction, featuring block lengths of 648, 1296, and 1944 bits and 4-QAM, 16-QAM, and 64-QAM constellations. For each channel condition and scenario, an exhaustive search for all samples is conducted across all combinations of channel coding rates and constellation orders, each associated with its respective compression rates. The best performances are subsequently identified and employed as the benchmark for the given channel condition and scenario. Although exploring a larger set of code rate and constellation combinations could potentially enhance performance, the improvement is not expected to be significant and the overall performance is inherently constrained by the capacity-achieving separation-based scheme, as elaborated subsequently.

For the theoretical bound, we assume capacity-achieving channel codes for transmission, denoted as the BPG-Capacity scheme. For the MIMO system with CSIT, we apply the water-filling algorithm for power allocation among the parallel channels formed after Eqn. 8. For the open-looop MIMO system with CSIR, we assume that the transmitter knows the instantaneous capacity of each channel block without power allocation, and can select the best compression rate. Note that, in practice, the transmitter does not have access to the CSI, and would probably employ a conservative compression rate to avoid a potential outage.

Our model was implemented in Pytorch with two GTX 3090Ti GPUs. We use a learning rate of $5e^{-5}$ and a batch size of 128 with an Adam optimizer. Models were trained until the performance on a validation set stopped improving. Considering the model complexity and the performance, we set $p = 8$, $l = 64$, $c = 48$ for image vectorization, and $L_t = 8$, $d = 256$, and $N_s = 8$ for each transformer layer of the ViT.

Each element of $\boldsymbol{H}$ is sampled from a complex Gaussian process as $H[i,j]\sim\mathcal{CN}(0,1)$, where we set $\sigma_h^2 = 1$. Results for a more general correlated MIMO channel are presented in the appendix of [46]. To measure the channel quality, we define the channel SNR, denoted by μ, as:

$$\mu \triangleq 10\log_{10}\frac{\mathbb{E}_{\boldsymbol{H},\boldsymbol{X}}[\|\boldsymbol{H}\boldsymbol{X}\|_F^2]}{\mathbb{E}_{\boldsymbol{W}}[\|\boldsymbol{W}\|_F^2]}\,(\text{dB}) = 10\log_{10}\frac{M}{\sigma_w^2}. \tag{32}$$

B. Open-Loop MIMO system with CSIR

We first evaluate DeepJSCC-MIMO in the open-loop MIMO system with CSIR. Specifically, we compare DeepJSCC-MIMO, trained at specific SNRs, with the BPG-Capacity and BPG-LDPC schemes at different SNR values and bandwidth ratios. For the BPG-LDPC scheme, we apply the sphere decoding (SD) algorithm [53].

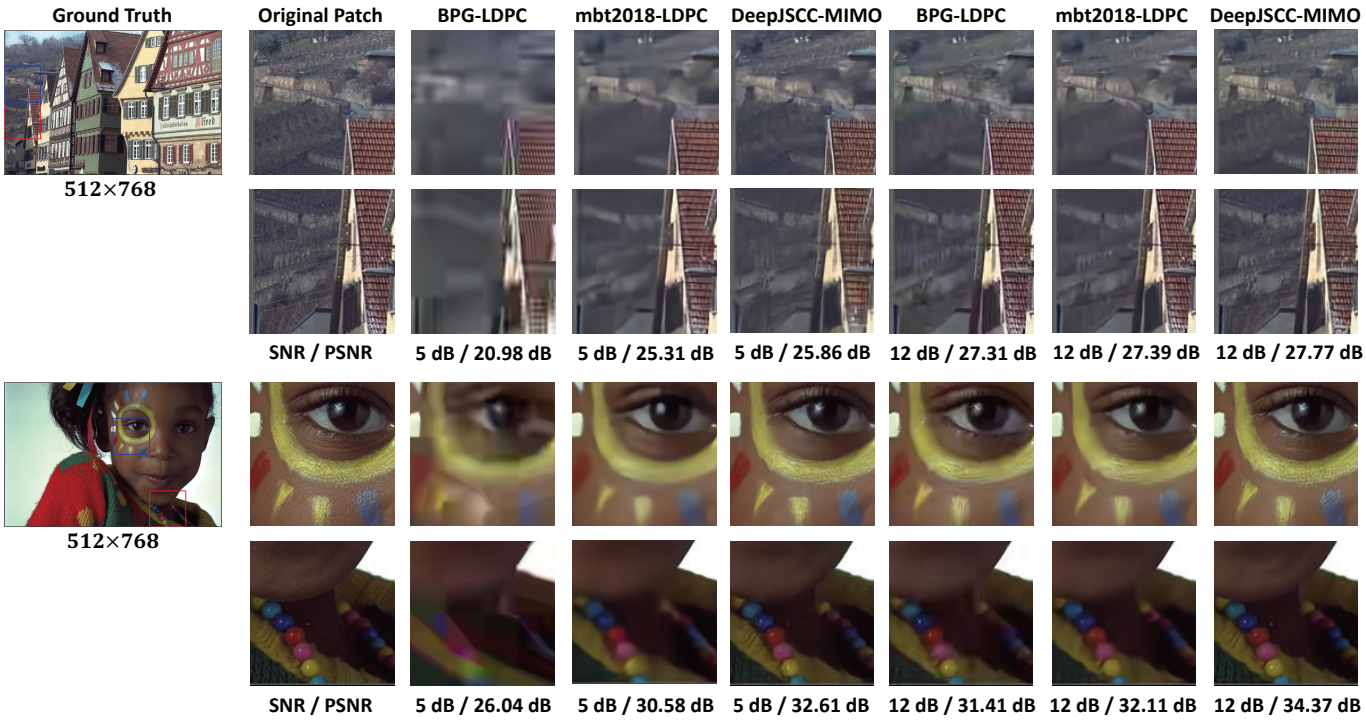


Fig. 5. Visual comparisons of images from the Kodak dataset transmitted over the open-loop MIMO channel at SNR 5 dB and 12 dB with $R = 1/24$. The first and second columns are the original image and the original patches of the bounding boxes.

1) General performance: Fig. 4a and Fig. 4c show the PSNR performance of the DeepJSCC-MIMO, BPG-Capacity, and BPG-LDPC schemes over the SNR values from 0 dB to 20 dB when the bandwidth ratio is $R = 1/24$ and $R = 1/12$. We can observe that our DeepJSCC-MIMO can generally outperform the BPG-Capacity and BPG-LDPC schemes in all SNR values. We emphasize that the BPG-Capacity performance shown in these figures is not achievable by practical channel coding schemes. Thus, it serves as an upper bound on any separation-based scheme that employs BPG for compression of the input image. We observe that DeepJSCC-MIMO provides significant improvements (at least 1.3 dB and 5.05 dB) for $R = 1/24$, compared with the BPG-Capacity scheme and the BPG-LDPC scheme, respectively. For $R = 1/12$, our DeepJSCC-MIMO can still outperform the BPG-Capacity scheme (at least 0.77 dB) and the BPG-LDPC scheme (at least 5.93 dB) in all SNRs. We observe that the gap between DeepJSCC-MIMO and BPG capacity is larger in the lower bandwidth ratio case.

We observe in Figs. 4b and 4d that DeepJSCC-MIMO achieves the best perceptual performance across a range of channel conditions and bandwidth ratios. Especially in poor SNR conditions, DeepJSCC-MIMO exhibits a substantial improvement over BPG-LDPC and BPG-Capacity, with a maximum LPIPS discrepancy of 0.59 and 0.35 at SNR 1dB for CIFAR10, respectively.

2) SNR-adaptability: Note that each point on the DeepJSCC-MIMO curve is obtained by training a separate encoder-decoder pair. We also consider a random SNR training strategy to evaluate the SNR adaptability of the DeepJSCC-MIMO scheme, denoted by DeepJSCC-MIMO-universal. We train the model with random SNR values uniformly sampled from $[0, 22]$ dB and test the well-trained model at different SNRs. The comparisons in Fig. 4 show that there is a slight performance degradation compared to training a separate DeepJSCC-MIMO encoder-decoder pair for each SNR (up to 0.99 dB for $R = 1/24$ and 2.02 dB for $R = 1/12$); however, the DeepJSCC-MIMO-universal brings significant advantage in terms of training complexity and storage requirements. We can conclude that the DeepJSCC-MIMO-universal scheme learns to adapt to different SNRs from the random training strategy at the expense of a slight loss in PSNR, which tends to increase with SNR.

TABLE II
COMPARISON OF DEEPJSCC-MIMO WITH SEPARATION-BASED BASELINES EMPLOYING DIFFERENT COMPRESSION ALGORITHMS, LDPC CODE AND SPHERE DECODING FOR THE OPEN-LOOP MIMO SYSTEM ON THE KODAK DATASET WHEN $SNR = 5$ DB.

Bandwidth ratio	1/24	1/12	1/8	1/6	1/4
BPG-LDPC [47]	24.05	25.50	27.02	28.21	29.70
MBT2018-LDPC [48]	24.48	26.03	27.64	28.91	30.39
VTM-LDPC [49]	24.81	26.31	27.91	29.15	30.68
ELIC-LDPC [50]	24.99	26.54	28.16	29.35	30.85
DeepJSCC-MIMO	**29.43**	**30.45**	**31.17**	**31.56**	**31.92**

3) High resolution datasets and different bandwidth ratios: To further study the effect of bandwidth ratio and channel SNR on the performance, and evaluate the model's generalizability, we train and test the DeepJSCC-MIMO scheme with various bandwidth ratios and SNRs individually over the Kodak dataset. Specifically, we train the models with randomly cropped 128×128 patches from the ImageNet dataset, and evaluate the well trained models on the Kodak dataset.

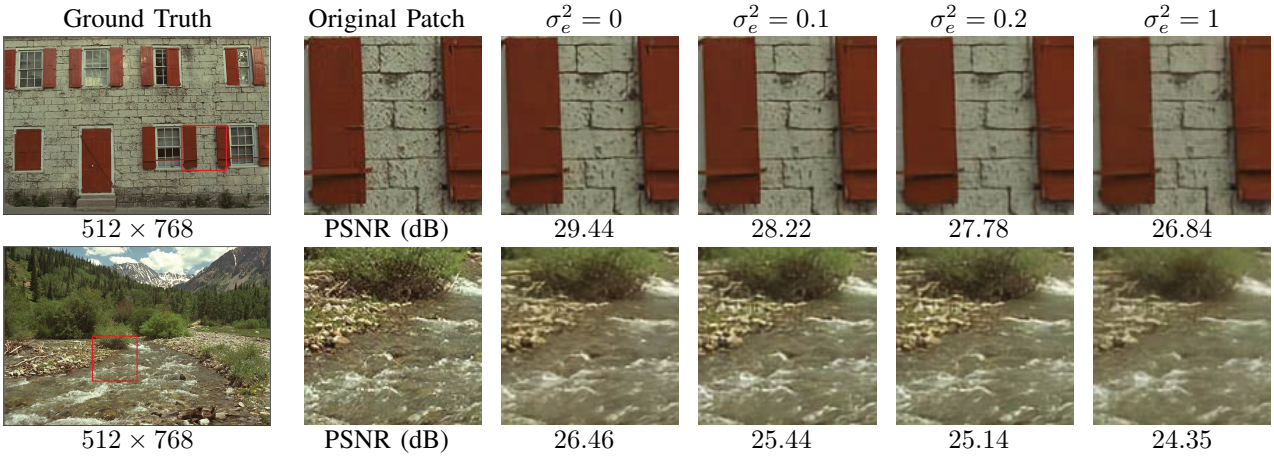


Fig. 6. Visual comparisons of images reconstructed by DeepJSCC-MIMO-universal in the presence of channel estimation errors, where the model is trained on the ImageNet dataset and validated on the Kodak dataset at SNR=12 dB and $R = 1/24$. The first and second columns are the source image and the image patch from the red bounding box of the original image. The third to sixth columns are the reconstructions of the same patches by DeepJSCC-MIMO universal with different channel estimation errors, respectively. The average PSNR metrics of the entire image are provided at the bottom of each visualized patch.

TABLE III
COMPARISONS OF DEEPJSCC-MIMO-UNIVERSAL AND BPG-LDPC SCHEMES WITH SPHERE DECODING OPERATIONS UNDER CHANNEL ESTIMATION ERRORS OVER THE KODAK DATASET FOR $R = 1/24$

Different SNR_{test} (dB)	5	7	9	12
DeepJSCC-MIMO ($\sigma_e^2 = 0$)	**29.38**	**30.08**	**30.80**	**31.62**
DeepJSCC-MIMO ($\sigma_e^2 = 0.2$)	29.18	29.86	30.53	31.33
DeepJSCC-MIMO ($\sigma_e^2 = 1$)	28.73	29.49	30.08	30.78
BPG-LDPC ($\sigma_e^2 = 0$)	24.05	27.23	28.92	30.27
BPG-LDPC ($\sigma_e^2 = 0.2$)	18.65	21.27	23.21	24.84
BPG-LDPC ($\sigma_e^2 = 1$)	15.01	15.37	15.72	16.03

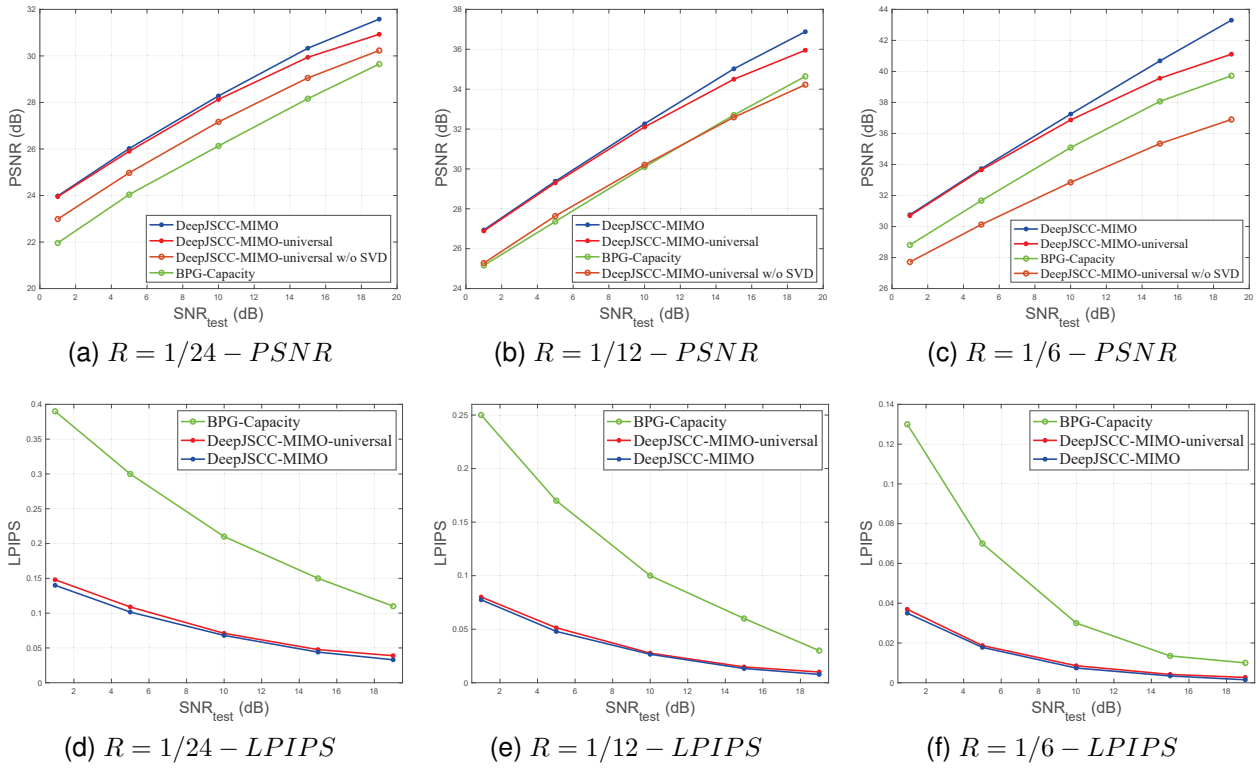
TABLE IV
THE NUMBER OF PARAMETERS (MILLIONS), FLOPS (G), PSNR PERFORMANCE (DB), AND ENCODING / DECODING TIME (MS) FOR DIFFERENT BACKBONES ACROSS VARIOUS BASIC BLOCK NUMBERS L_t.

Number of basic blocks L_t		4	8	12
Parameters	Proposed method	**6.61**	**12.92**	**19.23**
	DeepJSCC-CNN	6.89	20.00	33.11
	DeepJSCC-Swin	8.89	15.21	21.52
FLOPs	Proposed method	**0.43**	**0.83**	**1.23**
	DeepJSCC-CNN	0.83	1.67	2.51
	DeepJSCC-Swin	1.22	2.23	3.24
Performance	Proposed method	**23.76**	**24.41**	**24.32**
	DeepJSCC-CNN	22.83	23.54	23.04
	DeepJSCC-Swin	22.67	23.61	23.22
Complexity	Proposed method	$\mathcal{O}(4ld^2 + 2l^2d)$		
	DeepJSCC-CNN	$\mathcal{O}(k^2F^2C_{in}C_{out})$		
	DeepJSCC-Swin	$\mathcal{O}(4ld^2 + 2M_w^2ld)$		
Encoding time / decoding time	BPG-LDPC (CPU)	37.25 / 71.28		
	Proposed method (GPU)	**4.82 / 5.16**		

In Fig. 5, we present visualization examples of images transmitted by DeepJSCC-MIMO and BPG-LDPC schemes at 5 dB and 12 dB when $R = 1/24$, where SNR values and PSNR of the entire image are provided at the bottom of each visualized patch. It can be observed that reconstructed by DeepJSCC-MIMO are qualitatively better with more detailed high-frequency features. The comparisons here can clearly show the advantages of the DeepJSCC-MIMO in high-resolution datasets. We also conclude that training on a sufficiently large dataset (ImageNet) can allow our DeepJSCC-MIMO scheme to perform well on a never-seen dataset (Kodak) for a wide range of bandwidth ratios.

We also evaluate the model across various bandwidth ratios and compression algorithms [48]–[52] when SNR=5 dB over the Kodak dataset, as shown in Table. II. We want to emphasize that the sphere decoding algorithm in separation-based schemes is a much better MIMO detection method with higher computation complexity, compared with ZF employed by DeepJSCC-MIMO. We can observe that the superiority of DeepJSCC-MIMO persists across various bandwidth ratios and compression algorithms, where the gap is even larger at low bandwidth ratios, which shows that DeepJSCC-MIMO outperforms traditional schemes despite using a weaker detection algorithm.

4) Robustness to channel estimation errors: We take the same approach as in [38] to evaluate the robustness of MIMO systems to channel estimation errors, where $\boldsymbol{H}$ is imperfectly estimated as $\hat{\boldsymbol{H}}$ at the receiver. Similar to the work in [38], [54], [55], let $\boldsymbol{E}_n \triangleq \boldsymbol{H} - \hat{\boldsymbol{H}}$, where the entries of $\boldsymbol{E}_n$ are zero-mean complex Gaussian with variance σ_e^2. Here σ_e^2 is the 'noise variance' term to capture the channel estimation quality, which can be appropriately selected depending on the channel conditions and channel estimation method. We evaluate the DeepJSCC-MIMO-universal and BPG-LDPC schemes with different $\sigma_e^2 \in \{0, 0.2, 1\}$ values over different SNR values on the Kodak dataset with $R = 1/24$. For the DeepJSCC-MIMO-universal scheme, we train and test the model both with imperfect $\hat{\boldsymbol{H}}$. The experimental results are shown in Table. III. As expected, for both schemes, the performance degrades as the channel estimation error increases. In specific, for $\sigma_e^2 = 0.2$, the performance loss of our DeepJSCC-MIMO scheme is up to 0.29 dB, compared with the performance loss up to 5.96 dB from the BPG-LDPC scheme. For $\sigma_e^2 = 1$, the BPG-LDPC scheme starts to fail to work. However, our DeepJSCC-MIMO scheme can still work despite experiencing a limited performance decrease of less than 0.84 dB, which mainly arises from noisy channel equalization operations, as



(a) $R = 1/24 - PSNR$ (b) $R = 1/12 - PSNR$ (c) $R = 1/6 - PSNR$

(d) $R = 1/24 - LPIPS$ (e) $R = 1/12 - LPIPS$ (f) $R = 1/6 - LPIPS$

Fig. 7. Performance of the proposed DeepJSCC-MIMO model compared with the BPG-Capacity benchmark at different channel SNR and bandwidth ratio scenarios for the closed-loop MIMO.

demonstrated in the appendix of [46].

To visualize the performance of DeepJSCC-MIMO-universal scheme in the presence of channel estimation errors, we present the reconstructed images with different channel estimation errors in Fig. 6. We can observe that the increasing channel estimation error is reflected in the gradual loss of high-frequency details during transmission. We want to emphasize that even with a significant channel estimation error ($\sigma_e^2 = 1$), DeepJSCC-MIMO continues to function, with only limited performance loss, compared with the significant performance loss with the traditional separation benchmark. We can conclude that the proposed scheme is more robust to channel estimation errors, as it can learn to compensate for it during transmission.

5) Model efficiency and other architectures: The proposed DeepJSCC-MIMO scheme is versatile and can be seamlessly integrated with various network architectures. Table IV presents the model size, floating point operations (FLOPs), PSNR performance, and computational complexity for alternative DeepJSCC architectures with different block numbers when transmitting a single sample from the CIFAR10 dataset under $R = 1/24$ and SNR=5 dB. L_t represents the number of basic blocks in different DeepJSCC architectures. Specially, each CNN block [2], [21] is characterized by kernel size k, feature size F, along with input / output channel numbers C_{in} / C_{out}. On the other hand, each Swin block within Swin-based DeepJSCC [4], [36] is characterized by the window size M_w, sequence length l, and hidden dimension d. We can observe from our experiments on CIFAR10 dataset that the proposed method maintains the lowest parameter and FLOP counts when achieving the best transmission performance, reducing the memory consumption and FLOP operations by up to 44.74% and 61.4%, respectively.

Considering computational complexity, the proposed DeepJSCC method and CNN-based DeepJSCC methods exhibit quadratic computation complexity concerning the sequence length or input size. This underscores the necessity for meticulous design of patch sizes, particularly when dealing with high-resolution images such as 2K images, or alternatively, Swin block can be employed as an optional solution. It is also noteworthy that the inherently parallelizable architecture of our approach renders it particularly efficient for deployment on GPU-like systems, as opposed to conventional methods typically tailored for optimization on CPUs. The DeepJSCC-MIMO scheme implemented on a GPU is significantly faster compared to the traditional scheme executed on a CPU with the iterative decoding algorithm.

In summary, the proposed ViT-based solution significantly reduces both the inference complexity and parameter count. Furthermore, it is feasible to retain the total coding time of DeepJSCC-MIMO within 10 ms, which can be further reduced by encompassing alternative hardware, different architectures, optimized implementations, and acceleration techniques [46].

C. Closed-Loop MIMO system with CSIT

In this section, we consider the closed-loop MIMO system with CSI at the transmitter as well as at the receiver. We evaluate the DeepJSCC-MIMO trained at specific SNRs with

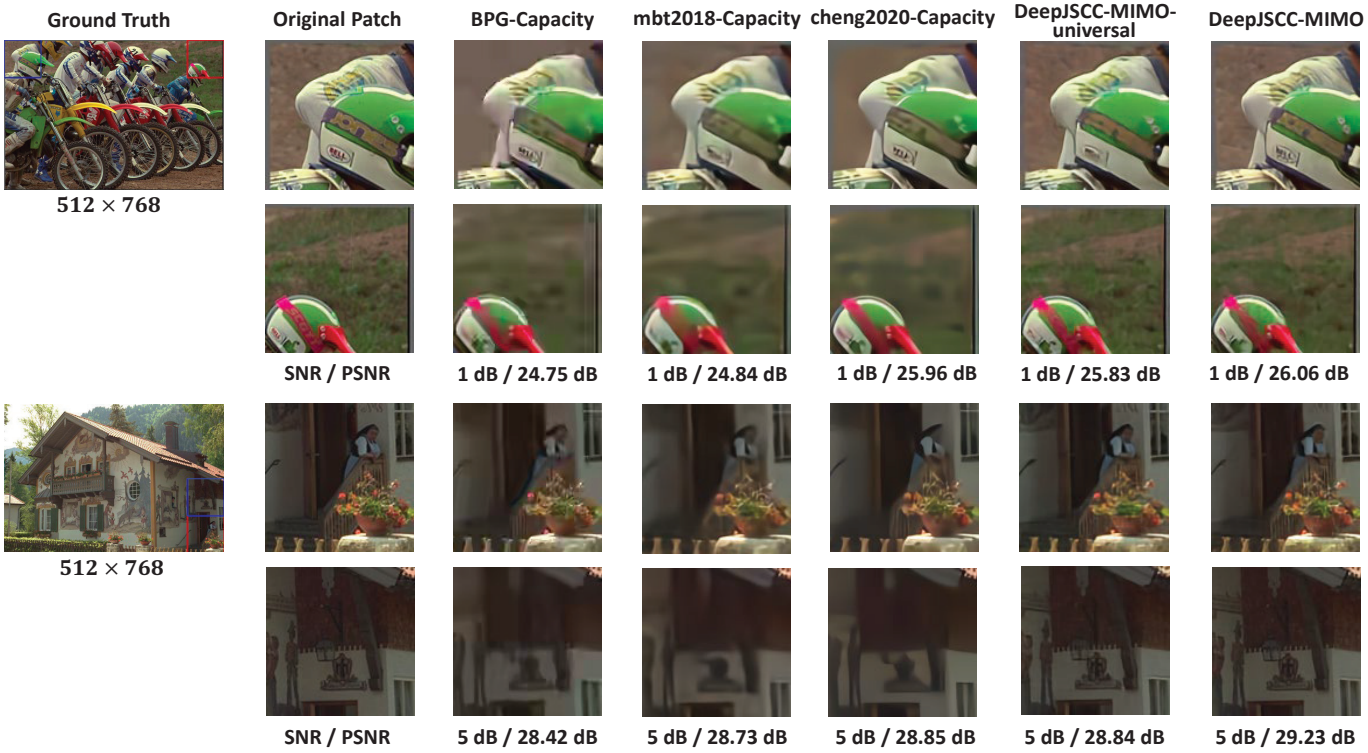


Fig. 8. Visual comparison of different schemes on two samples from the Kodak dataset at a bandwidth ratio $R = 1/24$ and different SNRs . The first and second columns are the original image and the original patch in the red and blue bounding boxes. The third, fourth, and fifth columns are the patch reconstructed by the BPG-Capacity [47], MBT2018-Capacity [48], and Cheng2020-Capacity [51] schemes, respectively. The last two columns demonstrate the patch recovered by the DeepJSCC-MIMO-universal and DeepJSCC-MIMO schemes.

$SNR_{test} \in [0, 20]$ dB in three bandwidth ratios $R = 1/24$, $R = 1/12$ and $R = 1/6$.

1) General performance: As before, we first evaluate the DeepJSCC-MIMO model under the setup where the training SNR, denoted by SNR_{train} matches the test SNR, SNR_{test}. In particular, we set $SNR_{train} = SNR_{test} \in \{1, 5, 10, 15, 19\}$ dB and the bandwidth ratio $R \in \{1/24, 1/12, 1/6\}$. The comparisons between DeepJSCC-MIMO and the BPG-Capacity benchmark is shown in Fig. 7.

In terms of the PSNR performance, DeepJSCC-MIMO outperforms the separation-based benchmark in all SNR and bandwidth-ratio scenarios. Specifically, we can see that DeepJSCC-MIMO outperforms the benchmark by at least 1.98 dB and 1.78 dB for $R = 1/24$ and $R = 1/12$, respectively. When $R = 1/6$, improvements of up to 3.58 dB can be observed. These significant improvements demonstrate the superiority of the DeepJSCC-MIMO scheme and its capacity in extracting and mapping image features to the available channels in an adaptive fashion. We observe in Figs. 7d, 7e and 7f that DeepJSCC-MIMO achieves the best perceptual performance across a wide range of channel conditions and bandwidth ratios, particularly in the low SNR regimes.

2) SNR-adaptability: We also consider a random SNR training strategy to evaluate the channel adaptability of DeepJSCC-MIMO. We train the model with random SNR values uniformly sampled from $[0, 22]$ dB, denoted by DeepJSCC-MIMO-universal, and test it at different SNRs. The comparisons in Fig. 7 show that there is a slight performance degradation compared to training a separate DeepJSCC-MIMO encoder-decoder pair for each channel SNR (up to 0.65 dB, 0.93 dB, and 2.19 dB for $R = 1/24$, $R = 1/12$ and $R = 1/6$); however, DeepJSCC-MIMO-universal brings significant advantage in terms of training complexity and storage requirements.

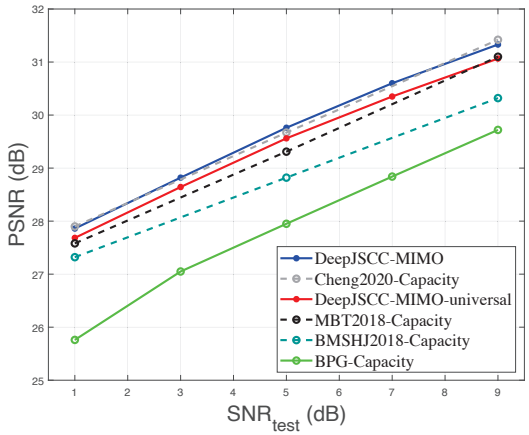


Fig. 9. Performance of DeepJSCC-MIMO and conventional separation-based scheme in the CelebA dataset when $R = 1/48$.

We can observe more performance loss in the high SNR regime. However, DeepJSCC-MIMO-universal still significantly outperforms the BPG-Capacity benchmark. We conclude that the DeepJSCC-MIMO-universal scheme can adapt to SNR variations with a slight loss in PSNR, especially in

TABLE V
ABLATION STUDY OF DEEPJSCC-MIMO-UNIVERSAL SCHEME WITH/WITHOUT CHANNEL HEATMAP ACROSS VARIOUS SNRS ON THE CIFAR10 DATASET WHEN $R = 1/12$.

Open-loop MIMO system with CSIR					
Models	1 dB	5 dB	10 dB	15 dB	19 dB
w/o heatmap	24.22	26.69	29.10	30.89	31.86
with heatmap	**24.95**	**27.56**	**30.02**	**31.91**	**32.93**
Closed-loop MIMO system with CSIT					
Models	1 dB	5 dB	10 dB	15 dB	19 dB
w/o heatmap	26.21	28.71	31.36	33.41	34.67
with heatmap	**26.89**	**29.30**	**32.10**	**34.50**	**35.95**

TABLE VI
ABLATION STUDY OF DEEPJSCC-MIMO-UNIVERSAL SCHEME WITH/WITHOUT CHANNEL HEATMAP ON THE CIFAR10 DATASET ACROSS VARIOUS SNRS IN THE FAST FADING CHANNEL WHEN $R = 1/48$.

Open-loop MIMO system with CSIR					
Models	1 dB	5 dB	10 dB	15 dB	19 dB
w/o heatmap	18.72	20.64	22.35	23.51	24.04
with heatmap	**19.24**	**21.23**	**23.11**	**24.47**	**25.17**
Closed-loop MIMO system with CSIT					
Models	1 dB	5 dB	10 dB	15 dB	19 dB
w/o heatmap	20.75	22.57	24.30	25.50	26.14
with heatmap	**21.56**	**23.44**	**25.23**	**26.67**	**27.30**

the high-SNR regime.

3) SVD ablation study: To evaluate the effectiveness of the SVD strategy in our scheme, we train the models of DeepJSCC-MIMO-universal without SVD decomposition-based precoding in different bandwidth ratios, denoted by DeepJSCC-MIMO-universal w/o SVD. Specifically, we directly feed the channel heatmap $\boldsymbol{M}$ into both transceivers and the performance is shown in Fig. 7. Compared with the DeepJSCC-MIMO-universal, we can observe that the performance of DeepJSCC-MIMO-universal w/o SVD has a gap of up to 0.97 dB, 1.91 dB and 4.21 dB for $R = 1/24$, $R = 1/12$, and $R = 1/6$, respectively. Although the networks can still learn to communicate over the MIMO channel, the SVD-based model-driven strategy can significantly simplify the training process and improve performance, illustrating the importance of exploiting domain knowledge in designing data-driven communication technologies.

4) High resolution datasets: To evaluate the model generalizability, we validate the DeepJSCC-MIMO on Kodak and CelebA datasets. Specifically, we train the DeepJSCC-MIMO and DeepJSCC-MIMO-universal schemes with randomly cropped 128×128 patches from the ImageNet dataset, and evaluate the well-trained models on the Kodak and CelebA datasets. The detailed visualization and the PSNR performance of the transmission, compared with BPG-Capacity, are shown in Fig. 8 and Fig. 9, respectively.

Similarly to the CIFAR dataset, from Fig. 9, we can observe that our DeepJSCC-MIMO can outperform BPG-Capacity at all test SNRs and generally provide higher gains at lower SNRs (up to 2.1 dB) in the CelebA dataset. There is a slight performance degradation of the DeepJSCC-MIMO-universal scheme (up to 0.27 dB) compared with DeepJSCC-MIMO trained at specific SNRs; however, the DeepJSCC-MIMO-universal can still outperform the BPG-Capacity scheme at all SNRs. Additionally, we consider state-of-the-art learning-based compression algorithms along with capacity-achieving codes and water-filling algorithm, denoted as Cheng2020-Capacity [51], MBT2018-Capacity [48], and BMSHJ2018-Capacity schemes [52]. The superior performance of Deep JSCC-MIMO is evident when compared to both the MBT2018-Capacity and BMSHJ2018-Capacity schemes. Furthermore, Deep-JSCC-MIMO achieves similar performance with the Cheng2020-Capacity scheme. It is also noteworthy that capacity-achieving channel codes are practically not possible, and require accurate channel estimation and sufficiently long block lengths to approximate in practice.

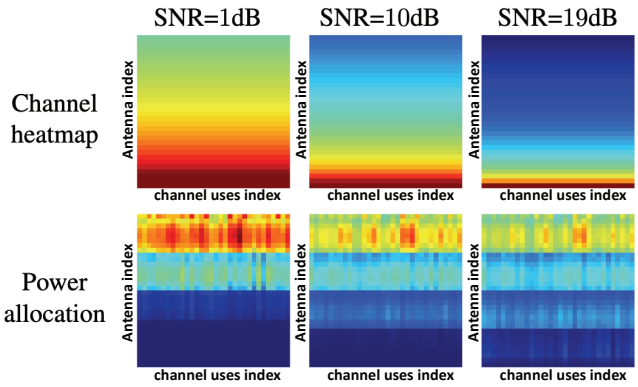


Fig. 10. Visualization of the channel heatmap and power allocation across the channel symbols, where antenna number is set as $M = 32$ and $R = 1/6$

We also visualize the results of the DeepJSCC-MIMO scheme on the Kadak dataset and compare them with those of the BPG-Capacity scheme in Fig. 8. We can observe that DeepJSCC-MIMO performs better in the low SNR regime, such as 1 and 5 dB, with more detailed high-frequency features. Comparing the performance of DeepJSCC-MIMO-universal and DeepJSCC-MIMO schemes, we can observe that the DeepJSCC-MIMO-universal can achieve comparable performance at the expense of a slight drop in distortion.

Considering that the BPG-Capacity performance shown in these figures is an upper bound on the performance of any separation-based scheme that employs the BPG codec, the comparison here clearly shows the benefits of the DeepJSCC-MIMO and DeepJSCC-MIMO-universal schemes in high-resolution datasets. We can also conclude that a sufficiently large dataset (ImageNet) can allow our DeepJSCC-MIMO scheme to perform well on other never-seen datasets (Kodak/CelebA) for a wide range of channel conditions (SNRs).

5) Channel heatmap ablation study: To assess the effectiveness of the channel heatmap, we conduct ablation studies over the DeepJSCC-MIMO-universal scheme, as outlined in Tables. V. Specifically, we can observe a maximum increase in PSNR of 1.07 dB and 1.28 dB for open-loop and closed-loop MIMO systems when $R = 1/12$, respectively. Our experimental findings demonstrate that integrating the channel heatmap is essential to enhance the performance of the DeepJSCC-MIMO-universal scheme in both open-loop and closed-loop MIMO systems across various SNRs and bandwidth ratios, as shown in the appendix of [46]. Similar findings can also

be observed in more challenging fast fading channels, where individual channel symbols experience fast fading, resulting in distinct channel heatmap in each time slot, as elaborated in Table. VI.

6) Visualization of the channel heatmap and power allocation: We visualize the channel heatmap and the power allocation of the channel input matrix for a closed-loop MIMO system in Fig. 10. Within our channel heatmap, we visualize the equivalent noise power experienced by each channel symbols, with the color red indicating higher noise power, and hence, a poorer channel condition. In the power allocation map, the color red denotes a higher allocation of power to the respective channel symbols. We can observe that for each SNR value, DeepJSCC-MIMO generally assigns more power to antennas with better channel conditions. Specifically, in the low SNR regime, DeepJSCC-MIMO concentrates most of the power on a few select antennas with exceptional quality. In certain extreme cases, DeepJSCC-MIMO refrains from allocating power to the sub-channels characterized by the worst channel conditions. As the SNR increases, we note that DeepJSCC-MIMO demonstrates a tendency to distribute power more evenly across a greater number of antennas, instead of concentrating on a limited set of antennas with the worst channel conditions.

7) Adaptability to the antenna numbers: In all the above experiments, we have the same number of $M = 2$ antennas at the transmitter and receiver. In this subsection, we investigate the generalizability of the DeepJSCC-MIMO architecture to different antenna numbers. In particular, we repeat the experiment of DeepJSCC-MIMO in Fig. 7b for different M values. The performance of our DeepJSCC-MIMO models with different antenna numbers $M \in \{2, 3, 4\}$ and channel SNRs $SNR_{test} \in \{1, 5, 10\}$ dB is shown in Fig. 11. We can observe that DeepJSCC-MIMO can generally achieve better performance with the increase in the number of antennas, thanks to its more flexible power allocation.

We also consider a random antenna number training strategy with a maximal antenna number setting M_{max}, denoted as 'adaptive-M'. We train the DeepJSCC-MIMO model with M uniformly sampled from $\{2, \ldots, M_{max}\}$ and test the well-trained model with different M values. Specifically, our encoder generates the channel symbols $\boldsymbol{X} \in \mathbb{C}^{M_{max} \times k}$, and only transmits first $\mathbb{C}^{M_i \times k}$ symbols for the $M = M_i$ antenna scenario. At the same time, the receiver pads the received signal into $\mathbb{C}^{M_{max} \times k}$ with zeros for decoding. The representation of CSI heatmaps differs across scenarios with distinct antenna configurations, serving as a guide for effectively encoding the channel symbols. We think these adaptive characteristics are attributed to the transformer's powerful learning and generalization capabilities. From experiments with $M_{max} = 4$, we observe in Fig. 12 that the adaptive-M strategy can achieve similar performance compared with models trained for specific M values: the performance loss is at most 0.6dB. We can conclude that DeepJSCC-MIMO is highly flexible and adaptive and can be applied to different number of antennas without retraining.

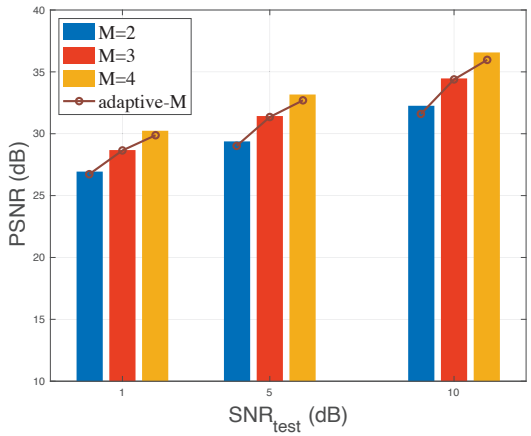


Fig. 11. Performance of DeepJSCC-MIMO with different antennas number.

V. Conclusion

We introduced the first DeepJSCC-enabled MIMO communication system designed for wireless image transmission. The proposed DeepJSCC-MIMO model utilizes a vision transformer-based architecture to exploit contextual semantic features and channel conditions through a self-attention mechanism. The results demonstrate significant improvements in transmission quality across a wide range of SNRs and bandwidth scenarios, with enhanced robustness to channel estimation errors. Moreover, DeepJSCC-MIMO is a unified and adaptable design capable of accommodating varying channel conditions and different antenna numbers without the need for retraining.

Moving forward, there are several potential directions for future exploration based on the findings of this paper:

- JSCC with space-time coding. In the open-loop scenario, full potential of the spatial and temporal dimensions in MIMO can be harnessed by integrating DeepJSCC with space-time codes [56]. This synergy can allow for even greater flexibility and performance enhancements, especially in the low-SNR regime.
- Variable length JSCC. Building upon the adaptability of ViT-based models to varying antenna numbers, future investigations can extend this concept to accommodate variable code lengths, where the channel resources are judiciously exploited depending on the channel state as well as the input signal.

References

[1] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. on Cognitive Comms. and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[2] M. Yang, C. Bian, and H.-S. Kim, "OFDM-guided deep joint source channel coding for wireless multipath fading channels," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 584–599, 2022.

[3] 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 Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2315–2328, 2022.

[4] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, "Nonlinear transform source-channel coding for semantic communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2300–2316, 2022.

[5] D. B. Kurka and D. Gündüz, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 8081–8095, 2021.

[6] C. Bian, Y. Shao, H. Wu, E. Ozfatura, and D. Gunduz, "Process-and-forward: Deep joint source-channel coding over cooperative relay networks," *arXiv preprint arXiv:2403.10613*, 2024.

[7] M. Yang and H.-S. Kim, "Deep joint source-channel coding for wireless image transmission with adaptive rate control," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5193–5197.

[8] H. Yoo, L. Dai, S. Kim, and C.-B. Chae, "On the role of ViT and CNN in semantic communications: Analysis and prototype validation," *IEEE Access*, vol. 11, pp. 71 528–71 541, 2023.

[9] E. Erdemir, T.-Y. Tung, P. L. Dragotti, and D. Gunduz, "Generative joint source-channel coding for semantic image transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 8, pp. 2645–2657, 2023.

[10] C. Bian, Y. Shao, H. Wu, and D. Gunduz, "Deep joint source-channel coding over cooperative relay networks," in *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE, 2024.

[11] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[12] F. Hekland, P. A. Floor, and T. A. Ramstad, "Shannon-kotel-nikov mappings in joint source-channel coding," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 94–105, 2009.

[13] F. J. Vazquez-Araujo, O. Fresnedo, L. Castedo, and J. Garcia-Frias, "Analog joint source-channel coding over MIMO channels," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, pp. 1–10, 2014.

[14] I. E. Aguerri and D. Gündüz, "Joint source-channel coding with time-varying channel and side-information," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 736–753, 2015.

[15] M. Vaezi and F. Labeau, "Distributed source-channel coding based on real-field BCH codes," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1171–1184, 2014.

[16] Z. Peng, Y.-F. Huang, and D. Costello, "Turbo codes for image transmission-a joint channel and source decoding approach," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 868–879, 2000.

[17] L. Pu, Z. Wu, A. Bilgin, M. W. Marcellin, and B. Vasic, "LDPC-based iterative joint source-channel decoding for JPEG2000," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 577–581, 2007.

[18] M. Fresia, F. Peréz-Cruz, H. V. Poor, and S. Verdú, "Joint source and channel coding," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 104–113, 2010.

[19] A. Golmohammadi and D. G. M. Mitchell, "Concatenated spatially coupled LDPC codes with sliding window decoding for joint source-channel coding," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 851–864, 2022.

[20] Z. Xuan and K. Narayanan, "Low-delay analog joint source-channel coding with deep learning," *IEEE Transactions on Communications*, vol. 71, no. 1, pp. 40–51, 2023.

[21] H. Wu, Y. Shao, K. Mikolajczyk, and D. Gündüz, "Channel-adaptive wireless image transmission with OFDM," *IEEE Wireless Communications Letters*, vol. 11, no. 11, pp. 2400–2404, 2022.

[22] T.-Y. Tung, D. B. Kurka, and D. Gündüz, "Deepjscc: The future of wireless video transmission," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Demo Session*, 2020.

[23] T.-Y. Tung, D. B. Kurka, M. Jankowski, and D. Gündüz, "DeepJSCC-Q: Constellation constrained deep joint source-channel coding," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 4, pp. 720–731, 2022.

[24] Y. Shao and D. Gunduz, "Semantic communications with discrete-time analog transmission: A PAPR perspective," *IEEE Wireless Communications Letters*, vol. 12, no. 3, pp. 510–514, 2023.

[25] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[26] M. B. Mashhadi and D. Gündüz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6315–6328, 2021.

[27] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[28] H. Wu, Y. Shao, C. Bian, K. Mikolajczyk, and D. Gündüz, "Vision transformer for adaptive image transmission over MIMO channels," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 3702–3707.

[29] D. Gunduz and E. Erkip, "Joint source–channel codes for MIMO block-fading channels," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 116–134, 2008.

[30] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *arXiv preprint arXiv:1707.07980*, 2017.

[31] J. Song, C. Häger, J. Schröder, T. J. O'Shea, E. Agrell, and H. Wymeersch, "Benchmarking and interpreting end-to-end learning of MIMO and multi-user communication," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 7287–7298, 2022.

[32] X. Zhang, M. Vaezi, and T. J. O'Shea, "SVD-embedded deep autoencoder for MIMO communications," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 5190–5195.

[33] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, 2023.

[34] W. F. Lo, N. Mital, H. Wu, and D. Gündüz, "Collaborative semantic communication for edge inference," *IEEE Wireless Communications Letters*, vol. 12, no. 7, pp. 1125–1129, 2023.

[35] Y. Shi, S. Shao, Y. Wu, W. Zhang, X.-G. Xia, and C. Xiao, "Excess distortion exponent analysis for semantic-aware MIMO communication systems," *IEEE Transactions on Wireless Communications*, vol. 22, no. 9, pp. 5927–5940, 2023.

[36] S. Yao, S. Wang, J. Dai, K. Niu, and P. Zhang, "Versatile semantic coded transmission over MIMO fading channels," *arXiv preprint arXiv:2210.16741*, 2022.

[37] J. Baltersee, G. Fock, and H. Meyr, "Achievable rate of MIMO channels with data-aided channel estimation and perfect interleaving," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 12, pp. 2358–2368, 2001.

[38] T. Yoo and A. Goldsmith, "Capacity and power allocation for fading MIMO channels with channel estimation error," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2203–2214, 2006.

[39] Y. Shao, E. Ozfatura, A. G. Perotti, B. M. Popović, and D. Gündüz, "Attentioncode: Ultra-reliable feedback codes for short-packet communications," *IEEE Transactions on Communications*, vol. 71, no. 8, pp. 4437–4452, 2023.

[40] E. Ozfatura, Y. Shao, A. G. Perotti, B. M. Popović, and D. Gündüz, "All you need is feedback: Communication with block attention feedback codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 3, pp. 587–602, 2022.

[41] H. Wu, Y. Shao, E. Ozfatura, K. Mikolajczyk, and D. Gündüz, "Transformer-aided wireless image transmission with channel feedback," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2024.

[42] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE/CVF Conf. on Comp. Vis. and Ptrn. Recog.*, 2018, pp. 586–595.

[43] G. Scutari, D. P. Palomar, and S. Barbarossa, "The MIMO iterative waterfilling algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 1917–1935, 2009.

[44] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[45] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9355–9366, 2021.

[46] H. Wu, Y. Shao, C. Bian, K. Mikolajczyk, and D. Gündüz, "Deep joint source-channel coding for adaptive image transmission over MIMO channels," *arXiv preprint arXiv:2309.00470*, 2023.

[47] "Better portable graphics 0.9.8," *URL https://bellard.org/bpg/*, 2018.

[48] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Neural Information Processing Systems*, 2018, pp. 10 794–10 803.

[49] "Versatile video coding reference software version 12.1 (vtm-12.1)." *https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM*, vol. 6, 2021.

[50] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *IEEE/CVF Conf. on Comp. Vis. and Ptrn. Recog.*, 2022.

[51] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *IEEE/CVF Conf. on Comp. Vis. and Ptrn. Recog.*, 2020.
[52] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Int'l Conf. on Learning Representations, ICLR*, 2018.
[53] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE Journal on selected areas in communications*, vol. 24, no. 3, pp. 491–503, 2006.
[54] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 951–963, 2003.
[55] R. Etkin and D. Tse, "Degrees of freedom in some underspread MIMO fading channels," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1576–1608, 2006.
[56] C. Bian, Y. Shao, H. Wu, and D. Gündüz, "Space-time design for deep joint source channel coding of images over MIMO channels," in *2023 IEEE 24th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2023, pp. 616–620.



Haotian Wu (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in Electrical Engineering from Zhejiang University, China, in 2017 and 2020, and a M.S. degree in Control Systems from Imperial College London in 2018. He is currently pursuing a Ph.D. degree at Imperial College London. His research interests include wireless communications and machine learning. He received the Best Paper Award at IEEE International Conference on Communications 2023.



Yulin Shao (Member, IEEE) received the B.S. and M.S. degrees in Communications and Information Engineering (Hons.) from Xidian University, China, in 2013 and 2016, and the Ph.D. degree in Information Engineering from the Chinese University of Hong Kong in 2020. He was a research assistant with the Institute of Network Coding (INC) from Mar. 2015 to Aug. 2016, a visiting scholar with the Research Laboratory of Electronics at Massachusetts Institute of Technology (MIT) from Sept. 2018 to Mar. 2019, a research associate with the Department of Electrical and Electronic Engineering at Imperial College London (ICL) from Jan. 2021 to Nov. 2022, and a Lecturer in Information Processing at the University of Exeter from Nov. 2022 to Aug. 2023. He is currently an Assistant Professor at the State Key Laboratory of Internet of Things for Smart City, University of Macau, and a Visiting Researcher at the Department of Electrical and Electronic Engineering, Imperial College London. His research interests include wireless communications and networking, machine learning, and stochastic control. He received the Best Paper Award at IEEE International Conference on Communications 2023. He is a Series Editor of IEEE Communications Magazine on "Artificial Intelligence and Data Science for Communications".



Chenghong Bian received the B.S. degree from the Physics Department, Tsinghua University, in 2020, and the M.S. degree from the EECS Department, University of Michigan, in 2022. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, Imperial College London. His research interests include semantic communications, channel coding, and deep learning. He received the Best Paper Award at IEEE International Conference on Communications 2023.



Krystian Mikolajczyk is a Professor in Computer Vision and Machine Learning at Imperial College London. He completed his PhD degree at the Institute National Polytechnique de Grenoble and held a number of research positions at INRIA, University of Oxford and Technical University of Darmstadt, as well as faculty positions at the University of Surrey, and Imperial College London. His main area of expertise is in multi view vision and recognition, in particular methods for image representation and learning. He has served in various roles at major international conferences co-chairing British Machine Vision Conference 2012, 2017 and IEEE International Conference on Advanced Video and Signal-Based Surveillance 2013. In 2014 he received Longuet-Higgins Prize awarded by the Technical Committee on Pattern Analysis and Machine Intelligence of the IEEE Computer Society.



Deniz Gündüz (Fellow, IEEE) received the B.S. degree in electrical and electronics engineering from METU, Turkey in 2002, and the M.S. and Ph.D. degrees in electrical engineering from NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively. Currently, he is a Professor of Information Processing in the Electrical and Electronic Engineering Department at Imperial College London, UK, where he also serves as the deputy head of the Intelligent Systems and Networks Group. In the past, he held various positions at the University of Modena and Reggio Emilia (part-time faculty member, 2019-22), University of Padova (visiting professor, 2018, 2020), Centre Tecnologic de Telecomunicacions de Catalunya (CTTC) (research associate, 2009-12), Princeton University (postdoctoral researcher, 2007-09, visiting researcher, 2009-11) and Stanford University (research assistant professor, 2007-09). His research interests lie in the areas of communications and information theory, machine learning, and privacy.

Deniz Gündüz is a Fellow of the IEEE. He is an elected member of the IEEE Signal Processing Society Signal Processing for Communications and Networking (SPCOM) and Machine Learning for Signal Processing (MLSP) Technical Committees. He serves as an Area Editor for the IEEE Transactions on Information Theory and IEEE Transactions on Communications. He is the recipient of the IEEE Communications Society Communication Theory Technical Committee (CTTC) Early Achievement Award in 2017, Starting (2016) and Consolidator (2022) and Proof-of-Concept (2023) Grants of the European Research Council (ERC), and has co-authored several award-winning papers, including the IEEE Communications Society - Young Author Best Paper Award (2022), and IEEE International Conference on Communications Best Paper Award (2023). He received the Imperial College London - President's Award for Excellence in Research Supervision in 2023.