Wireless Image Transmission Using Deep Source Channel Coding With Attention Modules

Jialong Xu, *Student Member, IEEE,* Bo Ai, *Fellow, IEEE,* Wei Chen, *Senior Member, IEEE,* Ang Yang, Peng Sun, Miguel Rodrigues, *Senior Member, IEEE*

***Abstract*—Recent research on joint source-channel coding (JSCC) for wireless communications has achieved great success owing to the employment of deep learning (DL). However, the existing work on DL based JSCC usually trains the designed network to operate under a specific signal-to-noise ratio (SNR) regime, without taking into account that the SNR level during the deployment stage may differ from that during the training stage. A number of networks are required to cover the scenario with a broad range of SNRs, which is computational inefficiency (in the training stage) and requires large storage. To overcome these drawbacks our paper proposes a novel method called Attention DL based JSCC (ADJSCC) that can successfully operate with different SNR levels during transmission. This design is inspired by the resource assignment strategy in traditional JSCC, which dynamically adjusts the compression ratio in source coding and the channel coding rate according to the channel SNR. This is achieved by resorting to attention mechanisms because these are able to allocate computing resources to more critical tasks. Instead of applying the resource allocation strategy in traditional JSCC, the ADJSCC uses the channel-wise soft attention to scaling features according to SNR conditions. We compare the ADJSCC method with the state-of-the-art DL based JSCC method through extensive experiments to demonstrate its adaptability, robustness and versatility. Compared with the existing methods, the proposed method takes less storage and is more robust in the presence of channel mismatch.**

This work is supported by the National Key R&D Program of China (2018YFE0207600, 2020YFB1807201); the Key-Area Research and Development Program of Guangdong Province (2019B010157002); the Natural Science Foundation of China (61911530216, 61961130391, U1834210); the Beijing Natural Science Foundation (L202019); the State Key Laboratory of Rail Traffic Control and Safety (RCS2021ZZ004, RCS2020ZT010) of Beijing Jiaotong University; NSFC Outstanding Youth Foundation under Grant 61725101; the Royal Society Newton Advanced Fellowship under Grant NA191006; vivo research grant.*(corresponding authors: Bo Ai; Wei Chen).*

Jialong Xu is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: jialongxu@bjtu.edu.cn).

Bo Ai is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China, also with Frontiers Science Center for Smart High-speed Railway System, Beijing 100044, China, also with Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518055, China, and also with Henan Joint International Research Laboratory of Intelligent Networking and Data Analysis, Zhengzhou University, Zhengzhou 450001, China (e-mail: boai@bjtu.edu.cn).

Wei Chen is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China, and also with Frontiers Science Center for Smart High-speed Railway System, Beijing 100044, China (e-mail: weich@bjtu.edu.cn).

Ang Yang and Peng Sun are with vivo Communication Research Institute, Beijing 100015, China (e-mail: ang.yang@vivo.com; sunpeng@vivo.com).

Miguel Rodrigues is with the Department of Electronic and Electrical Engineering, University College London, London, WC1E 7JE, U.K. (e-mail: m.rodrigues@ucl.ac.uk).

***Index Terms*—Joint source-channel coding, deep learning, deep neural network, attention mechanism.**

I. Introduction

FROM the first generation to the fifth generation of mobile communication systems, one traditionally adopts a modular approach to design a communications system (e.g., the transmitter is typically divided into source coder, channel coder and modulation module). Indeed, Shannon's separation theorem [1] showcases that a transceiver can be decomposed into a source coding and a channel coding without loss of optimality under certain conditions. However, the theorem assumes that the codeword lengths can be arbitrarily large. This is not realistic in various wireless settings due to latency considerations, such as in emerging wireless enabled applications like autonomous driving, smart manufacturing and telemedicine.

Shannon's groundbreaking work in 1948—which asserts that the fundamental problem of communication is that of reproducing at sink either exactly or approximately a message—does not explicitly dictate the use of separate source-channel coding (SSCC). In fact, Shannon states that if natural redundancy of the source is matched to the statistical characteristics of the channel input, to combat channel noise, there is no need to remove source redundancy [2]. From then on, joint source-channel coding (JSCC) had become a research hotspot. Gallager gave a mathematical expression of the lower bound of the lossless joint source-channel coding [3]. The expression clearly shows the code length of JSCC is shorter than that of SSCC in the same error performance. Csiszár adopted random coding method for the discrete memoryless system containing discrete memoryless source and discrete memoryless channel to give the lower bound and the upper bound expressions of error exponent of JSCC [4]. Zhong et al. studied error exponents of a point-to-point communication system [5] and a two-user asymmetric communication system [6], and systematically compared the error exponent between SSCC and JSCC. Moreover, the research community has also proposed various concrete JSCC designs. One JSCC strategy leverages SSCC techniques by applying resource assignment [7], information interaction [8] and unequal error protection [9] to adapt to different signal-to-noise ratios (SNRs). The other JSCC strategy integrates source coding and channel coding as a single process to optimize the communication system [10].

More recently, in view of its impressive performance in domains such as computer vision [11], speech processing

[12] and natural language processing [13], researchers have also been using deep learning (DL) approaches to support source or channel coding. For example, in the source coding domain, Google research has demonstrated that DL can lead to outstanding image compression results [14]–[17]. Toderici et al. [14] and Ballé et al. [15] demonstrated the DL based autoencoder [18] can achieve better compression performance than JPEG and JPEG2000, respectively. Minnen et al. further improved the method of [15] that completely surpasses the hand-engineered image codec BPG in [16] and [17]. There are also some other works. To improve the quality of image compression, Jiang et al. [19], Mishra et al. [20] and Zhao et al. [21] proposed to combine existing codecs (e.g., JPEG, JPEG2000 and BPG), wavelet transform and multiple description coding with autoencoders, respectively.

In turn, in the channel coding domain, O'Shea et al. were the first to construct an end-to-end autoencoder with performance close to the Hamming code [22]. Considering high-speed mobile channel, Xu et al. introduced measured channel data into the autoencoder, evaluated its performance in real channels and demonstrated the advantage of the improved autoencoder in channel mismatch condition [23]. Jiang et al. proposed Low-latency Efficient Adaptive Robust Neural in [24] and TurboAE code in [25] to demonstrate the superiority of channel coding based on DL in comparison with conventional channel coding techniques, in the regime of short-medium block-length. Other approaches adopted to support effective wireless communications have resorted to reinforcement learning and generative adversarial networks [26], [27].

These successes have in turn also motivate the use of DL based JSCC. For example, [28] has proposed DL based JSCC for the transmission of text over wireless communications channels for the first time. [29] designed a joint source-channel encoder and a joint source-channel decoder for image source. This DL based JSCC scheme outperforms SSCC schemes combining JPEG or JPEG2000 with capacity-achieving channel codes. [30] proposed three hierarchical DL based JSCC schemes leveraging the joint source-channel encoder and joint source-channel decoder of [29] as basic structures for successive refinement of images. [31] proposed a DL based JSCC scheme that incorporates channel output symbols to the transmission system, which further improves the performance of DL based JSCC. Considering a discrete channel, [32] proposed a new discrete variational autoencoder model named Neural Error Correcting and Source Trimming based on the maximization of the mutual information between the source and noisy received codeword. Variational inference for Monte Carlo objectives [33] is used to circumvent the non-differential network introduced by the discrete channel. For processed sources, [34] used the joint source-channel encoder for dimension reduction and the joint source-channel decoder for feature recovery to transmit the person's features extracted by the neural network based on ResNet-50 architecture [35]. For the extension of JSCC, [36] proposed a joint transmission-recognition scheme that builds the encoder combining the feature extraction with the joint source-channel encoder and the decoder combining recognition with the joint source-channel decoder to transmit the encoded feature wirelessly to the server for recognition tasks.

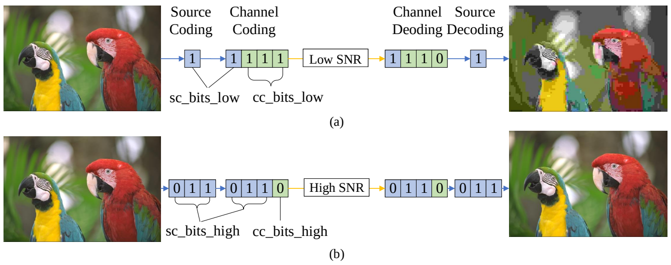


Fig. 1. Illustration of traditional JSCC. (a) In the low SNR regime, more bits (cc_bits_low) are allocated for channel coding and fewer bits (sc_bits_low) are allocated for source coding, (b) In the high SNR regime, fewer bits (cc_bits_high) are allocated for channel coding and more bits (sc_bits_high) are allocated for source coding.

However, the existing DL based JSCC methods assume that the channel conditions—notably the SNR—used to optimize the networks are the same as the channel conditions experienced during network deployment in the actual communications system. However, as shown in [29] and [37], a mismatch between the channel conditions used during the network optimization and network deployment stage can lead to serious performance degradation, limiting the advantages of DL based JSCC techniques. In principle, one could train multiple networks suited to a range of SNR that would then be selected during transmission/reception depending on the exact SNR condition, but this would lead to transceivers exhibiting considerable computational/storage requirements limiting its applicability in resource-constrained application such as IoT. In summary, in view of the fact that wireless channels can experience varying SNR levels, one needs to develop new DL based JSCC methods capable of adapting to varying channel conditions. In this paper, we design a new DL based JSCC method—leveraging traditional JSCC design principles—that can operate successfully over a wide range of SNRs. The inspiration of our method originates from the resource assignment strategy adopted by traditional JSCC illustrated in Fig. 1. Specifically, when the channel exhibits bad condition, for the same image, more bits are allocated to the channel encoder and fewer bits are allocated to the source encoder. The increased number of bits allocated to the channel encoder improve the redundancy to combat the intense channel noise. Conversely, when the channel is in good condition, for the same image, fewer bits are given to the channel encoder and more bits are given to the source encoder. The increased source bits are used to improve image quality. Some cross-layer optimization for image/video streaming approaches drawing on traditional concatenated source channel coders have demonstrated to be effective as shown in [38], [39]. In our proposed method, channel-wise soft attention network is used to replace the artificially designed resource allocation strategy to dynamically adjust the compression ratio in source coding and the channel coding rate according to the range of SNR. Compared with [29] where neither the transmitter nor the receiver has channel SNR knowledge, our proposed method leads to a large performance improvement. Compared with [31] where one has only access to delayed SNR knowledge based on feedback from receiver to transmitter, our proposed method uses attention mechanism as a resource allocation approach to allocate different contributions for intermediate features in the coding process according to channel SNR; it also feedback channel information more efficiently because it relays channel SNR back to transmitter rather than the channel output. Another advantage of our proposed method is that our proposed method is more robust than the method proposed in [29] in the presence of channel mismatch. The most important contribution of our work is that we have successfully explored a way to design the JSCC base DL method with the help of traditional JSCC design principles in the development of wireless communications.

The rest of this work is organized as follows. In Section II, we introduce the system model. Then, the proposed method is presented in Section III. In Section IV, we offer a series of simulation results to showcase the performance of our method. Section V is dedicated to the evaluation of the performance and the storage of the proposed method compared with other DL based JSCC schemes. Finally, the paper is concluded in Section VI.

II. System Model

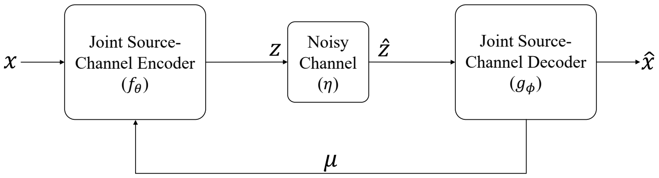

x
Joint Source-Channel Encoder (f_θ)
z
Noisy Channel (η)
$\hat{z}$
Joint Source-Channel Decoder (g_ϕ)
$\hat{x}$
μ

Fig. 2. The model of the point-to-point image transmission system with SNR feedback.

Consider a point-to-point image transmission system with SNR feedback as shown in Fig. 2. Channel SNR is known both at the joint source-channel encoder and the joint source-channel decoder. An input image of size H(height)×W(weight)×C(channel) is represented by a vector $\boldsymbol{x} \in \mathbb{R}^n$, where n = H×W×C and $\mathbb{R}$ denotes the set of real numbers. The joint source-channel encoder encodes $\boldsymbol{x}$ and the feedback SNR μ, and the encoding function $f_\theta : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{C}^k$ leads to a vector of complex-valued channel input symbols $\boldsymbol{z} \in \mathbb{C}^k$. The encoding process can be expressed as:

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

where k is the size of channel input symbols, θ is the parameter set of the joint source-channel encoder, $\mu \in \mathbb{R}$ is the channel SNR that can be estimated at the joint source-channel decoder and fed back to the joint source-channel encoder, and $\mathbb{C}$ denotes the set of complex numbers. The encoder maps the n-dimensional vector of real-valued image $\boldsymbol{x}$ to a k-dimensional vector of complex-valued channel input samples $\boldsymbol{z}$. To satisfy the average power constraint at the joint source-channel encoder, $\frac{1}{k}\mathbb{E}(\boldsymbol{z}\boldsymbol{z}^*) \leq 1$ is also imposed, where $\boldsymbol{z}^*$ denotes the conjugate transpose of $\boldsymbol{z}$.

The encoded symbols $\boldsymbol{z}$ are transmitted over a noisy channel represented by the function $\eta : \mathbb{C}^k \rightarrow \mathbb{C}^k$. AWGN channel is considered in our work. The channel output symbols $\hat{\boldsymbol{z}} \in \mathbb{C}^k$ received by the joint source-channel decoder are expressed as:

$$\hat{\boldsymbol{z}} = \eta(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{\omega}, \tag{2}$$

where the vector $\boldsymbol{\omega} \in \mathbb{C}^k$ consists of independent and identically distributed (i.i.d) samples with the distribution $\mathbb{CN}(0, \sigma^2\boldsymbol{I})$. σ^2 is the noise power and $\mathbb{CN}(\cdot,\cdot)$ denotes a circularly symmetric complex Gaussian distribution. The proposed method can be applied to other differentiable channels. Consider the following fading channel:

$$\hat{\boldsymbol{z}} = h\boldsymbol{z} + \boldsymbol{\omega}, \tag{3}$$

where $h \in \mathbb{C}$ is the channel gain. By applying equalization at the receiver, the above model can be represented as Eq. (2), while the noise has a different distribution. The study on non-differentiable channel is out of the scope of this paper.

The joint source-channel decoder uses a decoding function $g_\phi : \mathbb{C}^k \times \mathbb{R} \rightarrow \mathbb{R}^n$ to map $\hat{\boldsymbol{z}}$ and μ as follows:

$$\hat{\boldsymbol{x}} = g_\phi(\hat{\boldsymbol{z}}, \mu) = g_\phi(\eta(f_\theta(\boldsymbol{x}, \mu)), \mu), \tag{4}$$

where $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ is an estimation of the original image $\boldsymbol{x}$, ϕ is the parameter set of the joint source-channel decoder. The distortion between the original image $\boldsymbol{x}$ and the reconstructed image $\hat{\boldsymbol{x}}$ is expressed as:

$$d(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2, \tag{5}$$

where x_i and $\hat{x}_i$ represents the intensity of the color component of each pixel corresponding to $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively.

Akin to [29]–[31], we call the image size n, the channel input size k and $R = k/n$ as the source bandwidth, the channel bandwidth and bandwidth ratio, respectively. Under a certain R, the goal is to determine the encoder and decoder parameters θ^* and ϕ^* that minimize the expected distortion as follows:

$$(\theta^*, \phi^*) = \underset{\theta,\phi}{\arg\min}\, \mathbb{E}_{p(\mu)}\mathbb{E}_{p(\boldsymbol{x},\hat{\boldsymbol{x}})}[d(\boldsymbol{x}, \hat{\boldsymbol{x}})], \tag{6}$$

where θ^* is the optimal encoder parameters, ϕ^* is the optimal decoder parameters, $p(\boldsymbol{x}, \hat{\boldsymbol{x}})$ represents the joint probability distribution of the original image $\boldsymbol{x}$ and the reconstructed image $\hat{\boldsymbol{x}}$, and $p(\mu)$ represents the probability distribution of the SNR. We will model the encoder and decoders using deep neural networks.

III. PROPOSED METHOD

The majority of existing DL based JSCC approaches are designed to operate under specific SNR [29]–[31], [34], [36]. There are also some recent JSCC based DL techniques that operate under a range of SNRs but these involve optimizing a series of networks for specific SNRs during the training stage and using a specific network adequate for current SNR conditions during the testing stage. This leads to serious drawbacks, including higher computational requirements during the training stage and higher storage demands during the testing stage.

Our goal is to design a single network for joint source-channel coding that can adapt to a wide range of SNR conditions. The proposed method is motivated by the resource assignment strategy in traditional concatenated source channel coders [40] which concatenate the source encoder and the channel encoder, and adjust the compression ratio and the channel coding rate according to the SNR to achieve the optimal quality of reconstructed images under the limited bandwidth. To transmit an image with some fixed bandwidth resource in the low SNR regime, one compresses the source more aggressively but simultaneously increases the channel coding rate in order to combat channel induced errors. On the other hand, in the high SNR regime, one compresses the source less aggressively but decreases the channel coding rate. This approach allows for nearly optimal transmission under a constant rate. Unfortunately, the existing DL based JSCC methods do not support a flexible network structure that can automatically change and adapt to the channel state.

We address this challenge by employing attention mechanism, which is a technique of DL widely used in natural language processing [41]–[43] and computer vision [44]–[47]. Such mechanisms use an additional neural network that can rigidly select certain features or assign different weights to different features in the original neural network. Our approach—so-called Attention DL based JSCC (ADJSCC)—has been constructed to specifically implement joint source-channel coding.

The architecture of ADJSCC has two parts, i.e., the neural encoder at the transmitter and the neural decoder at the receiver. The neural encoder usually consists of multiple nonlinear layers. The first few layers of the neural encoder can be seen as the source encoder, and the remaining layers of the neural encoder are regarded as the channel encoder. The input and output of the neural encoder in our proposed method are source values and channel symbols, respectively. Thus, it performs the source encoder function and the channel encoder function. The neural decoder in turn perform the reverse operations. Our approach exhibits two additional features: (a) first, one can adapt the compression rate as a function of the SNR conditions by allowing the source encoder to output more or less symbols; (b) second, one can also dynamically adjust the size of the sub-networks associated with the source encoder and channel encoder using attention mechanisms, which is akin to the adaptation of the source compression ratio and the channel coding rate according to the SNR.

We now describe in detail the proposed ADJCC design. Both the proposed encoder and decoder are constructed using two types of modules, i.e., a feature learning (FL) module and an attention feature (AF) module. FL modules and AF modules are connected alternately as shown in the upper part of Fig. 3. FL module learns features from the input of the FL module, and AF module takes the SNR and the output of the FL module as the input and produces a sequence of scaling parameters. The product of the outputs of the FL module and the AF module can be seen as a filtered version of the FL module output. The design of the FL module based on convolution layers has been studied in several works [29]–[32]. Therefore we focus on the design of AF module.

The ideal hard attention mechanism generates a mask containing elements equal to either 0 or 1 that changes the size of the effective (i.e., nonzero) features. However, the non-differentiable property of the loss in hard attention hinders the execution of the backpropagation algorithm in the training stage. Instead, soft attention is generally adopted instead of hard attention to facilitate backpropagation. The extracted features can be regarded as the signal components on the convolution kernel. The channel relationship are captured and different scaling parameters are generated for different channel features to increase or suppress their connection strength to the next layer. The aforementioned mechanism is channel-wise soft attention[1].

The architecture of AF module based on channel-wise soft attention is shown in the lower part of Fig. 3. Let $F^G = [F_1^G, F_2^G, \cdots, F_c^G] \in \mathbb{R}^{h\times w\times c}$ denote the features extracted by the FL module, where c is the number of the features and $h \times w$ is the size of each feature. Let also $F_A = [F_1^A, F_2^A, \cdots F_c^A] \in \mathbb{R}^{h\times w\times c}$ denote the scaled features produced by AF module. In summary, the AF module processes the features F^G using a global average pooling function. The results of the global average pooling is then concatenated with SNR to form the context information. The context information is fed into a full connected neural network that produces a scaling factor. The scaled features F^A are obtained by multiplying the features F^G and the scaling factor. This process results in different scaled features F^A depending on the exact SNR conditions.

We next elaborate further about the different components of the AF module: 1) context extraction; 2) factor prediction; and 3) feature recalibration.

1) Context Extraction: Context information includes channel SNR μ and feature information I^G. Image features are usually extracted by convolution kernels limited in a local receptive field. Hence these features cannot usually perceive the information out of this region especially using feature extraction with small kernel size. Global average pooling $G(\cdot)$ however can extract global information by averaging elements u_{jk} of F_i^G as follow:

$$I_i^G = G(F_i^G) = \frac{1}{h\times w}\sum_{j=1}^{h}\sum_{k=1}^{w} u_{jk} \in \mathbb{R}. \tag{7}$$

These global information features I_i^G are concatenated with the channel information SNR = μ to produce the context information I as follows:

$$I = (\mu, I_1^G, I_2^G, \cdots, I_c^G) \in \mathbb{R}^{c+1}. \tag{8}$$

2) Factor Prediction: We employ a factor prediction neural network $P_\omega(\cdot)$ to predict the scaling factor S based on the context information I. In order not to excessively increase the limit complexity, $P_\omega(\cdot)$ is a simple neural network consisting of two fully connected (FC) layers. The first FC layer contains a ReLu and the last FC layer contains a sigmoid in order to limit the output range to the interval (0,1). Therefore,

$$S = P_\omega(I) = \sigma(W_2\delta(W_1 I + b_1) + b_2) \in \mathbb{R}^c, \tag{9}$$

[1] The notion channel in "channel-wise soft attention" refers to the feature channel rather than the communications channel.

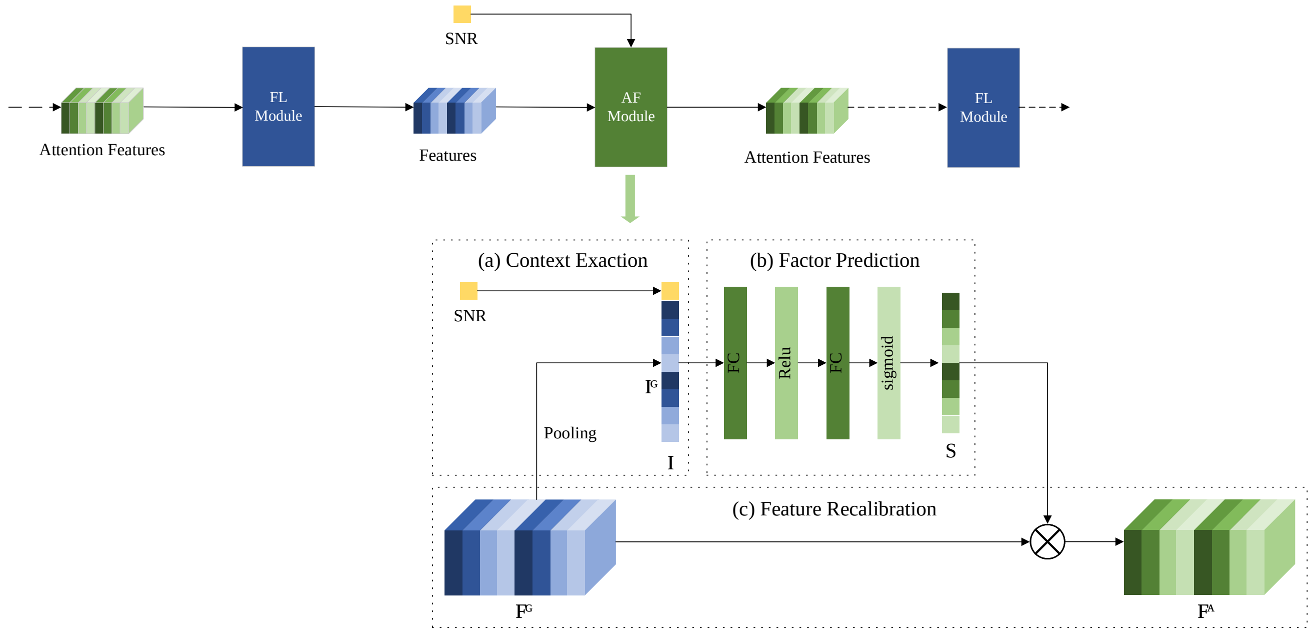


Fig. 3. The relationship between FL modules and AF modules in ADJSCC. FL Module based on convolution layers has been studied in several works [29]–[32]. AF module contains three parts: (a) Context Extraction, (b) Factor Prediction, (c) Feature Recalibration.

where W_1 and b_1 refer to the weights and the biases of the first FC layer, W_2 and b_2 refer to the weights and the biases of the second FC layer, and δ and σ represent the activation function ReLu and sigmoid, respectively. We let ω = (W_1, b_1, W_2, b_2) denote the parameter set of the factor prediction neural network.

3) Feature Recalibration: Finally, feature recalibration produces a feature map F^A based on the feature map F^G as follows:

$$F_i^A = R_e(F_i^G, S_i) = S_i \cdot F_i^G, i = 1, 2, \cdots, c, \tag{10}$$

where F_i^A denotes the i-th element in F^A, F_i^G denotes the i-th element in F^G, and S_i denotes the i-th element of S. The operation of our AF module is described in Algorithm 1. Different from the attention mechanism used in computer vision, we use SNR as the wireless channel information combined with the image inherent features to compute channel-wise attention. Once again, our motivation derives from the fact that the number of bits that one ought to allocate to source coding and channel coding should depend on the channel SNR.

Algorithm 1 AF module

```
Input: the features $F^G$, the SNR information $\mu$
Output: the attention features $F^A$
calculate the size of the features: $(height, width, channel) = \text{size}(F^G)$
calculate the global information features: $I^G = \text{GlobalAveragePooling}(F^G)$
calculate the context information $I = \text{concatenate}(\mu, I^G)$
calculate the scaling factor $S = P_\omega(I)$
convert the scaling factor $S$ to the channel-wise scaling factor $S_i, i = 1, 2, \cdots, c$
convert the features $F^G$ to the channel-wise feature $F_i^G, i = 1, 2, \cdots, c$
for i = 0 : 1 : c do
    channel-wise attention feature: $F_i^A = S_i \cdot F_i^G$
end for
convert the channel-wise attention feature $F_i^A, i = 1, 2, \cdots, c$ to the attention features $F^A$
return $F^A$
```

IV. Simulation Results

The first DL based JSCC architecture proposed by [29] consists of convolutional modules for image source. Each module consists of a convolution layer followed by a parametric ReLU (PReLU) every layer except the last one or a sigmoid activation function in the last layer. It has led to comparable performance to the standard SSCC scheme (JPEG/JPEG2000 + LDPC). [31] further improved the performance by introducing the generalized divisive normalization (GDN) as a normalization method and widening the channel of the convolution layer for each module. To prove the efficiency of our proposed ADJSCC scheme, we adopt the state-of-the-art DL based JSCC architecture used in [31] as the basic DL based JSCC (BDJSCC) architecture as shown in Fig. 4. The layers of the BDJSCC Encoder except the normalization layer, the reshape layer and the power normalization layer are divided into five modules. Each of the first four modules consists of a convolution layer, a GDN layer [48] and a PReLU layer [49]. The fifth module only consists of a convolution layer and a GDN layer. Similarly, the layers of the BDJSCC Decoder except for the normalization layer and the reshape layer are also divided into five modules. The first four modules have the same construction consisting of a transposed convolution layer, a GDN layer and a PReLU layer. The only difference between

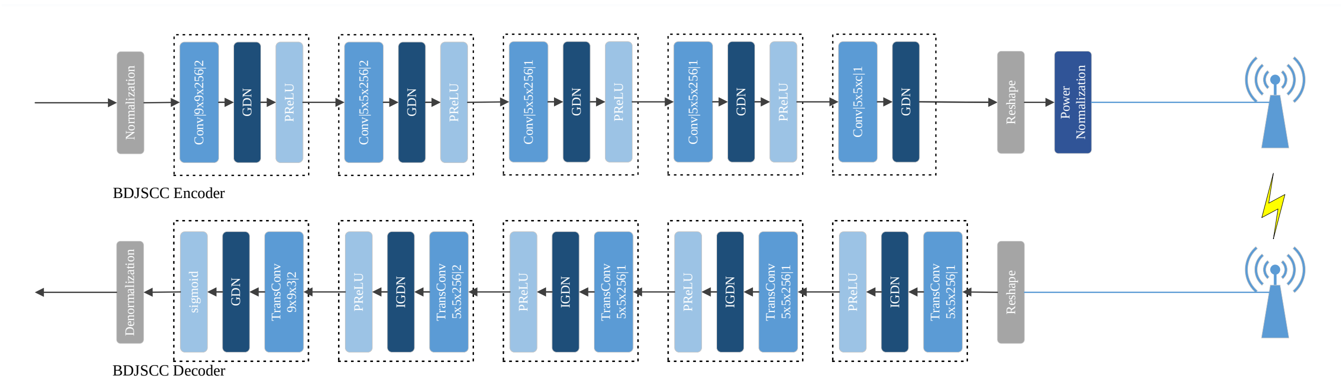


Fig. 4. The architecture of BDJSCC used in [31]. Convolution and transposed convolution are parameterized by $F \times F \times K|S$, where F and K are the filter size and the number of filters, respectively. In the convolution layer, S represents downsampling strides. In the transposed convolution layer, S represents upsampling strides.

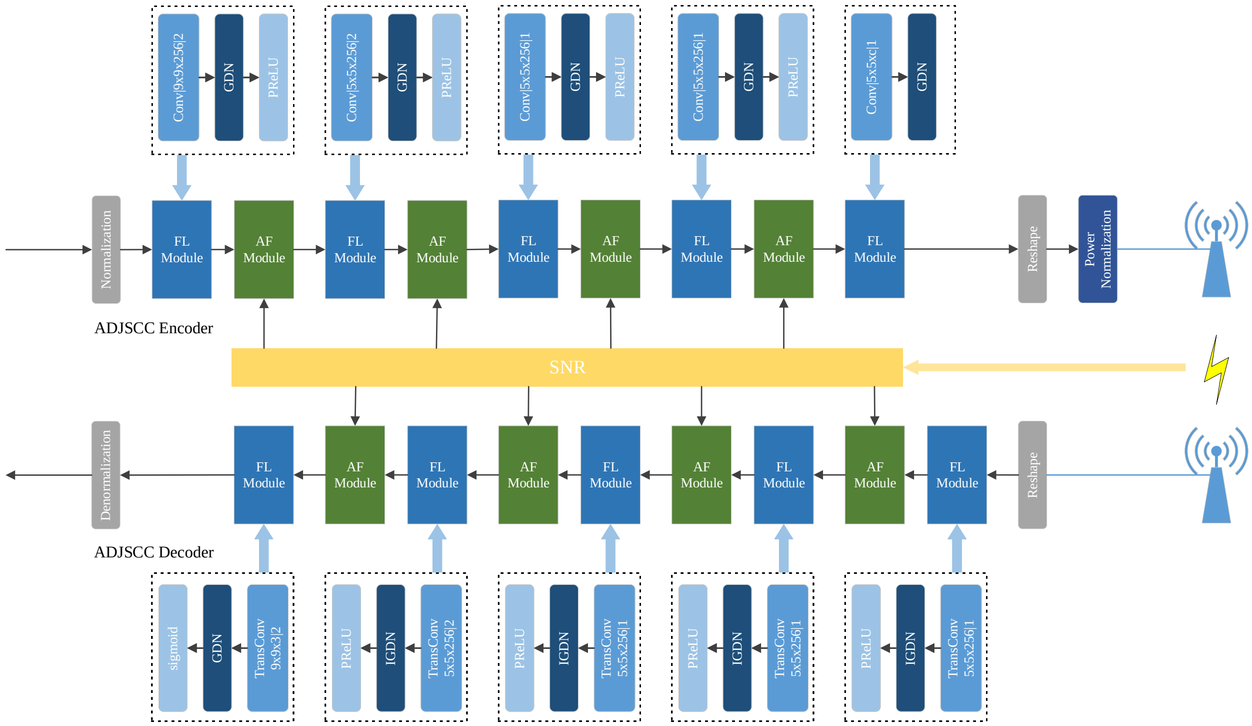


Fig. 5. The architecture of our proposed ADJSCC. The FL modules of ADJSCC consists of a convolution layer, a GDN layer and a PReLU (or sigmoid) layer. Each FL module is followed by an AF module except the last FL module in the encoder and the decoder. The SNR coming from channel feedback is another input of the AF module.

the last module and the first four modules is that the sigmoid layer replaces the PReLU layer. The notation $F \times F \times K|S$ in a convolution/transposed convolution layer denotes that it has K filters with size F and stride down/up S.

The corresponding ADJSCC architecture[2] is shown in Fig. 5. Five modules in BDJSCC Encoder are considered as five FL modules, each of which is followed by an AF module except the last FL module. The output of the previous FL module is one input of the present AF module and the output of the present AF module is the input of the next FL module. The other input of the AF module is the SNR level associated with the communications channel. The ADJSCC Decoder is also constructed similarly. By changing the output channel size c in the last convolution layer of the encoder, different bandwidth ratios can be obtained.

To compare with the existing method proposed in [31], we comply with the loss function and the metric used in [31], namely, the average mean squared error (MSE) and the average peak SNR (PSNR). The average MSE over N transmitted images is defined as follows:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{x}}^{(i)}), \tag{11}$$

[2] Source codes for constructing the ADJSCC architecture and the BDJSCC architecture are available at: https://github.com/alexxu1988/ADJSCC.

where $\boldsymbol{x}^{(i)}$ and $\hat{\boldsymbol{x}}^{(i)}$ represent the i-th original image and the corresponding reconstructed image, respectively, $d(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{x}}^{(i)})$ is the MSE defined in Eq. (5), and N is the number of image samples. In practice, Eq. (11) insteads of Eq. (6) by assuming a given distribution of SNR and equally transmitting images in the dataset. In turn, the PSNR is defined as follows:

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{MAX}^2}{\mathrm{MSE}}(\mathrm{dB}), \tag{12}$$

where MAX is the maximum possible value of the image pixels (e.g., MAX is 255 for the 8-bit color image). The PSNR is firstly calculated for each image and then averaged over all the tested images.

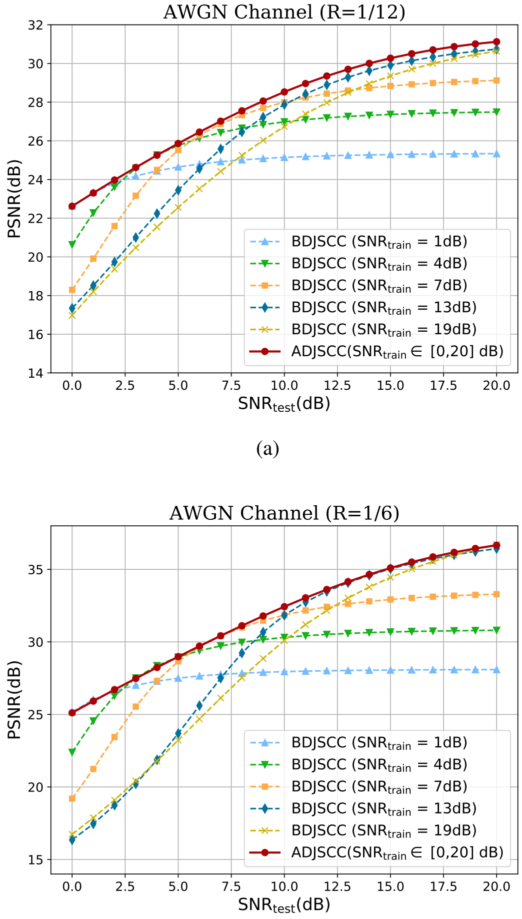
We use Tensorflow [50] and its high-level API Keras to implement the BDJSCC and ADJSCC[3] models. Consistently with the work [31], Adam optimizer with a learning rate of 10^{-4} and the batch size with 128 is chosen to optimize the models. To measure the training efficiency, we let the training epochs to be euqal to 1280. Unless stated otherwise, CIFAR-10 [51] is used for training and evaluating the BDJSCC and ADJSCC models. The CIFAR-10 dataset consists of 60000 $32 \times 32 \times 3$ color images associated with 10 classes where each class has 6000 images. Note however we are not concerned with the class of each image because our goal is to reconstruct the original data from the received one with minimum distortion only. Training dataset and test dataset contain 50000 images and 10000 images, respectively.

The BDJSCC method is trained at a specific SNR. In order to adapt to dynamic channel conditions, the ADJSCC method is trained under a uniform distribution within the SNR range [0, 20] dB. Both the performance of the ADJSCC method and the BDJSCC method are evaluated under the specific SNR. Each image in the test dataset is transmitted 10 times to alleviate the effect caused by the randomness of the channel noise. All of our experiments are performed on a Linux server with twelve octa-core Intel(R) Xeon(R) Silver 4110 CPUs and sixteen GTX 1080Ti GPU. Each experiment runs on six CPU cores and a GPU.

A. ADJSCC Adaptability Experiments

We first consider the performance of our proposed ADJSCC on the AWGN channel in Eq. (2). In the following experiment, the ADJSCC model is trained under the uniform distribution of $\mathrm{SNR}_{\mathrm{train}}$ from 0 dB to 20 dB, and all of the BDJSCC models are trained at specific $\mathrm{SNR}_{\mathrm{train}}$ = 1dB, 4dB, 7dB, 13dB, 19dB, respectively, which are adopted in [29]. We however evaluate the performance of both ADJSCC and BDJSCC models at specific $\mathrm{SNR}_{\mathrm{test}} \in$ [0,20] dB.

Fig. 6 compares the ADJSCC method with the BDJSCC method at bandwidth ratios R = 1/12, 1/6, respectively. In Fig. 6(a) with R = 1/12, the performance of the ADJSCC model is better than the performance of any BJSCC model trained at the specific $\mathrm{SNR}_{\mathrm{train}}$. With the increase of the $\mathrm{SNR}_{\mathrm{test}}$, the ADJSCC model brings a gradually increased performance, outperforming the BDJSCC model ($\mathrm{SNR}_{\mathrm{train}}$ = 1dB) by a margin of at most 6dB. With the decrease of the $\mathrm{SNR}_{\mathrm{test}}$, the ADJSCC model still outperforms the BDJSCC model ($\mathrm{SNR}_{\mathrm{train}}$ = 13dB or 19dB). It is also worth noting that the ADJSCC still outperforms the BDJSCC when $\mathrm{SNR}_{\mathrm{train}}$ = $\mathrm{SNR}_{\mathrm{test}}$—this is remarkable because ADJSCC is only trained under a certain SNR range. However, as $\mathrm{SNR}_{\mathrm{test}}$ deviates from $\mathrm{SNR}_{\mathrm{train}}$, the ADJSCC model tends to outperform considerably the BDJSCC one.



(a)

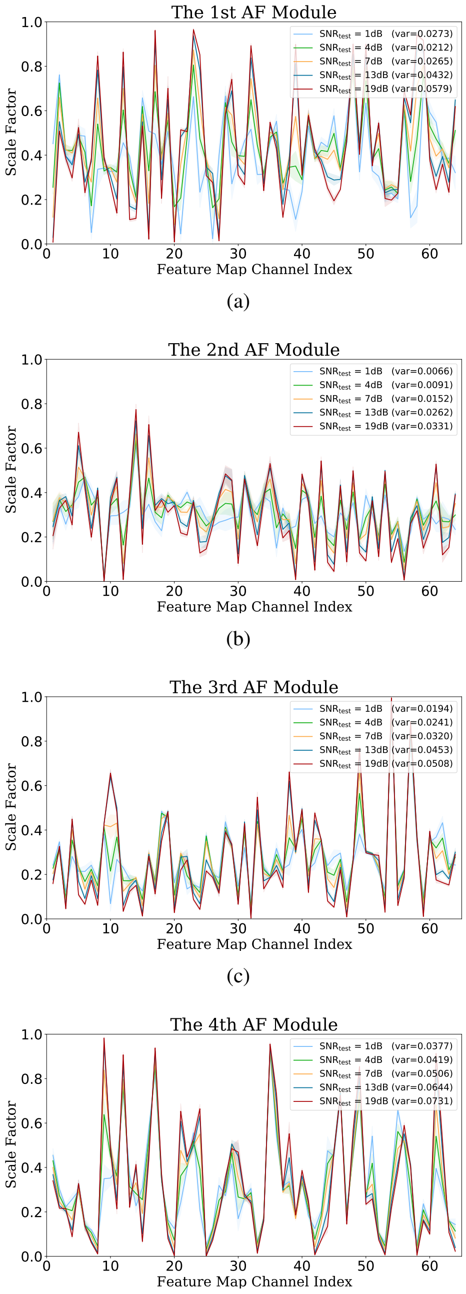
(b)

Fig. 6. Performance of ADJSCC and BDJSCC on CIFAR-10 test images. (a) R =1/12 and (b) R=1/6. The curve of ADJSCC is trained under the uniform distribution of SNR from 0dB to 20dB. Each curve of BDJSCC is trained at a specific SNR.

Fig. 6(b) with R = 1/6 reveals similar results to Fig. 6(a). However, with the increase of the bandwidth ratio, the performance gap between the ADJSCC model and the BDJSCC model ($\mathrm{SNR}_{\mathrm{train}}$ = 13dB or 19dB) at high $\mathrm{SNR}_{\mathrm{test}}$ regime almost disappears. Therefore, we conclude ADJSCC brings about higher performance gains than BDJSCC in the low bandwidth ratio regime. These results also suggest that using an ADJSCC is a much better strategy than using a collection of BDJSCCs where each BDJSCC is trained at a specific SNR level that are selected for transmission/reception depending on the actual channel conditions. Beyond the fact that such a strategy would lead to very complex transmitters and receivers,

[3] The following experiments demonstrate the ability of ADJSCC dealing with image source. With some appropriate modifications of the ADJSCC, the proposed mechanism can be applied to address other kinds of sources. However, it is out of our scope.

entailing considerable performance complexity, our results indicate that such a strategy does not lead to any performance gains over an ADJSCC strategy.



(a)

(b)

(c)

(d)

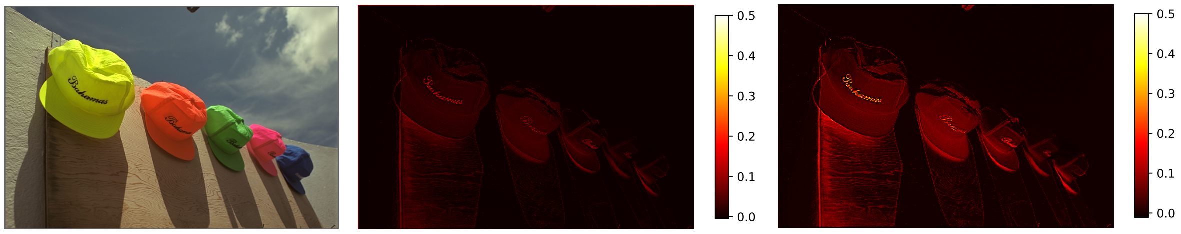
Fig. 7. The scaling factors of the first 64 channels in the encoder of ADJSCC on AWGN channel (R=1/6). (a) the scaling factors of the 1st AF module, (b)the scaling factors of the 2nd AF module, (c) the scaling factors of the 3rd AF module and (d) the scaling factors of the 4th AF module. Each curve of the scaling factors is evaluated at a specific SNR on CIFAR-10 test dataset. The solid line represents the mean values of the scaling factors and the translucent area around the solid line with the same color represents the standard derivation of the scaling factors.The var represents the variance of the mean values of the scaling factors on specific channel SNR.

The aforementioned performance evaluation of the ADJSCC model demonstrates that the ADJSCC model can accommodate for a range of SNR by using attention mechanism. We would like to understand how the AF module affects the features in practice. To provide a thoughtful understanding of the AF module, we study the scaling factors generated by the AF module for the features. Specifically, we transmit the test dataset of CIFAR-10 at specific $\mathrm{SNR}_{\mathrm{test}}$ and then compute the mean and standard deviation of the scaling factors for each AF module. We plot the mean and the standard deviation of the scaling factors in Fig. 7. The solid line represents the mean values of the scaling factors and the translucent area around the solid line with the same color represents the standard deviation of the scaling factors. In order to show the scaling factors for each AF Model more clearly, we select the scaling factors of the first 64 channels in the encoder. From top to bottom, the rows correspond to the order from the first AF module to the forth AF module in the encoder. Two observations are inferred from Fig. 7. First, the scaling factors across the channels fluctuate more drastically with the increase of the $\mathrm{SNR}_{\mathrm{test}}$. This suggests that the scaling factors are more sensitive to high $\mathrm{SNR}_{\mathrm{test}}$ than low $\mathrm{SNR}_{\mathrm{test}}$, which coincides with the intuition. When the $\mathrm{SNR}_{\mathrm{test}}$ is low, the channel noise is severe for each of the features. When the $\mathrm{SNR}_{\mathrm{test}}$ is high, some of the features have a better contribution to the performance than others. The scaling factors of good features should be increased to improve the performance and the scaling factors of bad features should be decreased to avoid resource occupation. Second, the difference between the curves at different $\mathrm{SNR}_{\mathrm{test}}$ gets smaller as the AF module gets deeper. This observation shows that the channel noise has a more significant impact on low-level features than high-level features. Low-level features concentrate on the pixel relationship of an image. In contrast, high-level features concentrate more on the semantic representation implied in an image than the pixel relationship. Compared with the low-level features, high-level features are more robust to the channel noise. Therefore in the fourth AF module shown in Fig. 7(d), the scaling factors at different $\mathrm{SNR}_{\mathrm{test}}$ are similar.

We also visualize the 23rd feature of the first AF module in the encoder at $\mathrm{SNR}_{\mathrm{test}}$ = 1dB shown in Fig. 8(b) and $\mathrm{SNR}_{\mathrm{test}}$ = 19dB shown in Fig. 8(c). The 32×32 images of CIFAR-10 dataset are too small to be recognized by human eyes. Thus we use a 512×768 image from Kodak dataset instead of the image from CIFAR-10 dataset. The detail of Kodak will be introduced in IV-C. Fig. 8(a) shows the original image. The comparison of Fig. 8(b) and Fig. 8(c) shows that when the channel exhibits high SNR, the information about caps should be enhanced to contribute more for the quality of the reconstructed image. It seems reasonable that the detailed information about the caps is easier to be disturbed as the channel gets worse. So when the channel exhibits high SNR, the detailed information about the caps should be decreased to save the transmit power for more robust information.

B. ADJSCC Robustness Experiments

We now study the robustness of the proposed AJSCC scheme in the presence of channel mismatch. Fig. 9 shows the mismatch performance of the ADJSCC model and the BDJSCC model under the AWGN channel with $R = 1/6$. Compared with the case without channel mismatch, the ADJSCC



(a) (b) (c)

Fig. 8. The original image and the heatmaps of the 23rd recalibrated feature in the first AM module of the encoder. (a) original image, (b) the heatmap at $\mathrm{SNR_{test}}$ = 1dB, (c) the heatmap at $\mathrm{SNR_{test}}$ = 19dB.

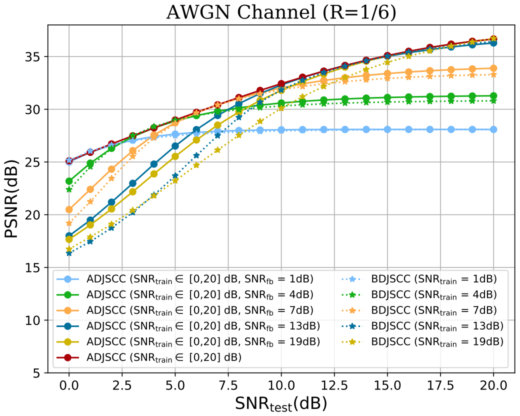


Fig. 9. Channel mismatch performance of ADJSCC and BDJSCC on CIFAR-10 test images.

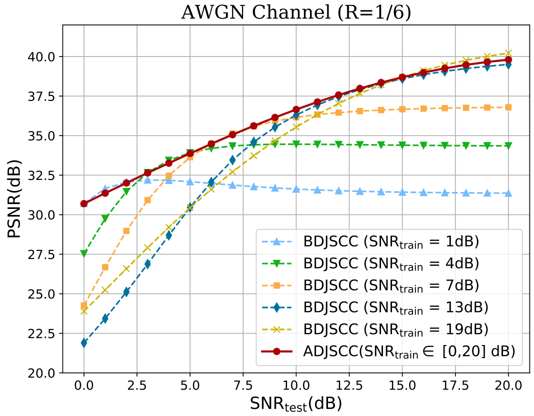


Fig. 10. Performance of ADJSCC and BDJSCC on Kodak dataset. ADJSCC and BDJSCC are trained on Imagenet dataset for bandwidth ratio R=1/6. The curve of ADJSCC is trained under the uniform distribution of SNR from 0dB to 20dB. Each curve of BDJSCC is trained at a specific SNR.

models have some performance loss in channel mismatch conditions. The ADJSCC models still outperforms the BDJSCC models in the case of channel mismatch. The ADJSCC model shows better robustness especially in high $\mathrm{SNR_{fb}}$ regimes. The gain of the ADJSCC against the BDJSCC is higher in the case when the feedback SNR is large.

C. ADJSCC Versatility

The effectiveness of ADJSCC has been demonstrated on CIFAR-10 dataset in Section IV-A. However, CIFAR-10 images exhibit low-resolution. Therefore, we also now test the performance of our proposed approach on higher resolution images as in [29] and [31]. It is worth noting that the duration of coded symbols transmitted by the transmitter need to be smaller than the coherence time to promise the constant SNR during the transmission for real wireless scenarios. For fading channels (e.g., Rayleigh channel) which suit for the large PHY block case, one can train the deep neural network with properly prepared channel.

We train ADJSCC on ImageNet dataset[4] [52] and evaluate ADJSCC on Kodak dataset. The ImageNet dataset contains various images of various sizes. We choose images with size larger than 128×128 and then randomly crop them to a size of 128×128 to generate the training dataset (About 5.8 million images satisfy this condition). Kodak dataset consists of 24 768×512 images. Two epochs with batch size of 16 and learning rate of 0.0001 is enough to make the ADJSCC model converge. The model is saved every 200 training batches. To average the random of the channel, each images in Kodak is transmitted 100 times to evaluate the ADJSCC performance. Note that it seems not rational to train our model on a dataset and evaluate our model on another dataset. However, training on a sufficiently complex dataset (e.g., ImageNet dataset) allows to perform well on other datasets (e.g., Kodak dataset). In addition, the size of evaluation dataset (768×512 in Kodak dataset) can be different from that of training dataset (128×128 in Kodak dataset) owes to the full convolutional architecture adopted in ADJSCC. Different from classical CNN networks[5], this full convolutional architecture only uses the convolutional layers to extract image features and uses transposed convolutional layers to restore the image, which allows different size of the input image as long as it meets the requirements of stride (the image size n must be a multiple of 4).

Fig. 10 shows the comparison of the ADJSCC model and the BDJSCC models on Kodak dataset. If we consider the BDJSCC models as a whole, the performance of the ADJSCC

[4] Until now, ImageNet dataset contains more than 14 million images with more than 21 thousand classes.

[5] Classical CNN networks (e.g., Alexnet) use the full connected layer after convolution layers to get fixed length feature vector for classification.

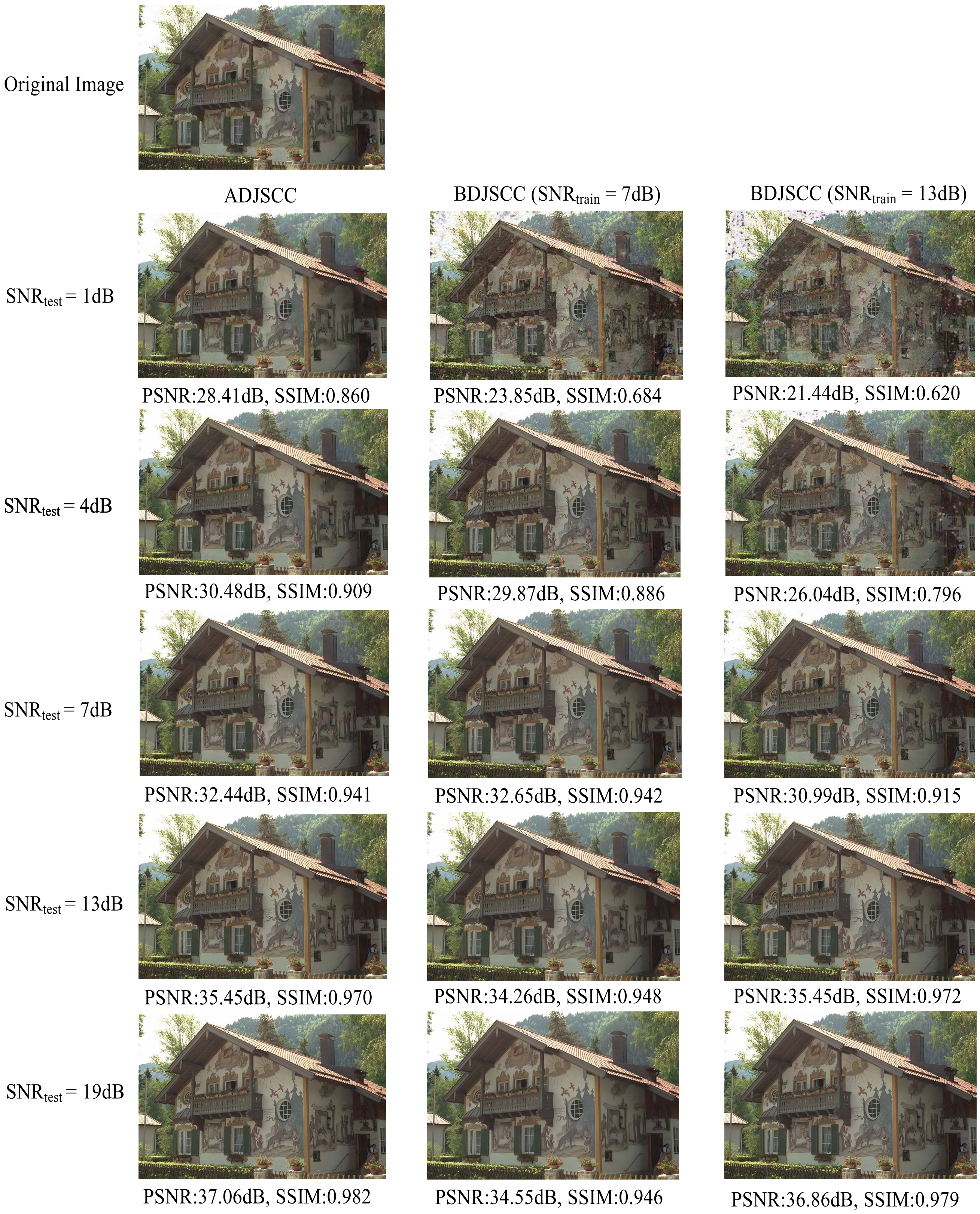


Fig. 11. Visual comparison between the ADJSCC model and the BDJSCC models ($\mathrm{SNR}_{\mathrm{train}} = 7\mathrm{dB}, 13\mathrm{dB}$) for the sample image of the Kodak dataset. We also used a perceptual metric—structural similarity index (SSIM)—to evaluate image quality.

model approaches that of the ensemble of BJSCC models when $\mathrm{SNR}_{\mathrm{test}} \leq 17\mathrm{dB}$ with negligible difference. Moreover, the performance of the ADJSCC model is 0.3dB lower than that of the ensemble of BJSCC models when $\mathrm{SNR}_{\mathrm{test}} >$

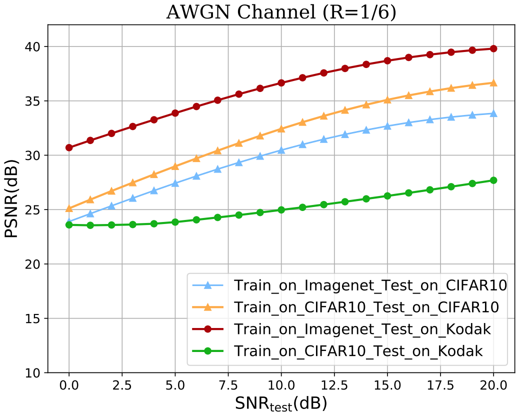


Fig. 12. Performance of the ADJSCC methods training on Imagenet or CIFAR-10 and testing on Kodak or CIFAR-10 for bandwidth ratio R=1/6. The ADJSCC methods are trained under the uniform distribution of SNR from 0dB to 20dB.

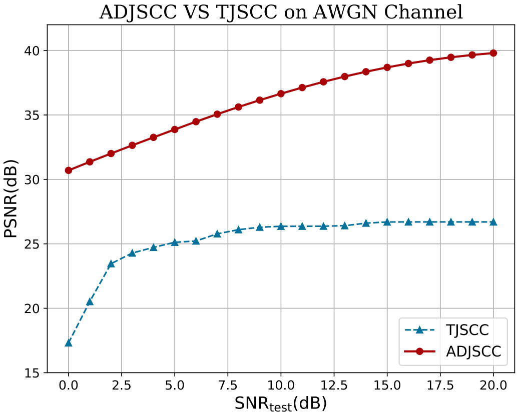


Fig. 13. Performance of ADJSCC and TJSCC on Kodak dataset. ADJSCC is trained on ImageNet dataset and with respect to the uniform distribution of SNR from 0dB to 20dB for bandwidth ratio R=1/6. The bandwidth ratio of the TJSCC $R_t = 2R = 1/3$.

17dB. Note however that the regime $\mathrm{SNR}_{\mathrm{test}} > 17\mathrm{dB}$ is associated with images with $\mathrm{PSNR} > 35\mathrm{dB}$ whose quality is virtually impossible to distinguish with human eyes[6].

We present a visual comparison between the ADJSCC model and the BDJSCC models for the sample image of the Kodak dataset in Fig. 11. Note that, even if the $\mathrm{SNR}_{\mathrm{test}}$ is the same, the reconstructed image is different at each time due to the randomness of the noise.

To further demonstrate the rationale of this experiment, we compare the performance of ADJSCC models trained on Imagenet or CIFAR-10 and tested on Kodak or CIFAR-10 in Fig. 12. The performance of the ADJSCC model trained on CIFAR-10 and tested on CIFAR-10 is better than the performance of the ADJSCC model trained on Imagenet and tested on CIFAR-10, which is consistent with the traditional requirement of DL that the test dataset should have the similar distribution with the training dataset to get good performance. Compared with the ADJSCC model trained on CIFAR-10 and tested on CIFAR-10, the ADJSCC model trained on Imagenet and tested on CIFAR-10 only has 1dB to 3dB performance loss in the range of SNR from 0dB to 20dB. However, the performance of the ADJSCC model trained on CIFAR-10 and tested on Kodak is much worse than the performance of the ADJSCC model trained on Imagenet and tested on Kodak. The performance loss caused by the image size mismatch is limited when the ADJSCC model is trained on a large number of higher resolution images (e.g., Imagenet), while there is a huge performance loss caused by the image size mismatch when the ADJSCC model is trained on a lower resolution images (e.g., CIFAR-10).

We finally compare our proposed method with the traditional JSCC (TJSCC) method proposed in [53], which is an unequal error protection scheme for lossy channels and fix packet size transmission. TJSCC uses Set Partitioning in Hierarchical Trees (SPIHT) coder as a source encoder to encode the original image to the embedded source bits. Cyclic redundancy check (CRC) bits are appended to the source bits to enhance the protection. Rate-compatible punctured convolutional (RCPC) codes are used as channel encoder and combined with fast unequal protection scheme to encode the embedded source bits to channel bits depending on the bandwidth ratio R_t, the rate-distortion of the original image and the channel status. BPSK modulation is assumed in TJSCC. Other settings are consistent with the setting in [53]. Because TJSCC is only fit for gray images, we separate the color images of Kodak into three channels. Each of the three channels is processed by TJSCC. The channel output symbols of ADJSCC are complex valued. However, the channel output symbols of TJSCC are real valued. For the sake of fairness, the bandwidth ratio of TJSCC is $R_t = 2R$, where R is the bandwidth ratio of ADJSCC. Fig. 13 shows the comparison of ADJSCC (trained on Imagenet dataset) and TJSCC on Kodak dataset. The performance improvement of TJSCC is fast from $\mathrm{SNR}_{\mathrm{test}} = 0\mathrm{dB}$ to $\mathrm{SNR}_{\mathrm{test}} = 2\mathrm{dB}$ and becomes slow from $\mathrm{SNR}_{\mathrm{test}} = 3\mathrm{dB}$ to $\mathrm{SNR}_{\mathrm{test}} = 15\mathrm{dB}$. When $\mathrm{SNR}_{\mathrm{test}} > 15\mathrm{dB}$, the performance of TJSCC is saturated. Compared with the TJSCC method, the ADJSCC method brings a large performance improvement. The minimum performance improvement is around 7dB at $\mathrm{SNR}_{\mathrm{test}} = 3\mathrm{dB}$ and the maximum performance improvement is around 13 dB at $\mathrm{SNR}_{\mathrm{test}} = 20\mathrm{dB}$. Besides the performance advantage, ADJSCC does not need to calculate the rate-distortion of the original image, resulting in a huge computational demand at the transmitter as shown in [53].

V. Storage Overhead and Computational Complexity

We finally calculate the storage overhead required by the BDJSCC model and the ADJSCC model with $R = 1/6$. The storage overhead of both ADJSCC and BDJSCC are independent of the image size. There are 10,690,351 total

[6] Note also that using an ensemble of BJSCC models in lieu of a single ADJSCC model is problematic because—as discussed earlier—it requires storing the different networks at transmitter and receiver, as well as switching between one or another depending on SNR conditions. This has considerably storage demands preventing the use of an ensemble of BJSCC models in various applications.

TABLE I
EVALUATION OF THE ADJSCC MODEL AND BDJSCC MODEL STRATEGIES

Strategy Name	Storage	PSNR
ADJSCC	41.04MB	**29.831dB**
BDJSCC-1	**40.78 MB**	24.474dB
BDJSCC-2	81.56MB	28.495dB
BDJSCC-5	203.9MB	29.694dB
BDJSCC-10	407.8MB	29.826dB

parameters in the BDJSCC model. The type of each parameter is float32, which requires 4 bytes. Hence the storage of the BDJSCC model is 10,690,351×4≈40.78 MB. The ADJSCC model has more parameters than the BDJSCC model because of the existing AF modules. The amount of the total parameters of the ADJSCC model is 10,758,191. The storage of the ADJSCC model is almost 41.04 MB. The ADJSCC model has 0.6% more parameters than the BDJSCC model, which needs a little additional storage.

However, we have already discussed that in practice one would desire to use an ensemble of BDJSCC models, each trained at a particular $\mathrm{SNR}_{\mathrm{train}}$ and tested at a $\mathrm{SNR}_{\mathrm{test}}$ = $\mathrm{SNR}_{\mathrm{train}}$. Let us denote by BDJSCC-1 as a strategy involving using one BJSCC model trained at $\mathrm{SNR}_{\mathrm{train}}$ = 10dB that is evaluated by the CIFAR-10 dataset with the assumed $\mathrm{SNR}_{\mathrm{test}}$. Let us in turn denote by BDJSCC-2 as a strategy involving using two BJSCC models trained at $\mathrm{SNR}_{\mathrm{train}}$ = 5dB, 15dB that are evaluated by the CIFAR-10 dataset with the assumed $\mathrm{SNR}_{\mathrm{test}}$. In particular, this strategy involves selecting between one BJSCC model trained at $\mathrm{SNR}_{\mathrm{train}}$ = 5dB and the other trained at $\mathrm{SNR}_{\mathrm{train}}$ = 15dB depending on the difference between $\mathrm{SNR}_{\mathrm{train}}$ and $\mathrm{SNR}_{\mathrm{test}}$. Finally, we let BDJSCC-5 denote a strategy involving using five models trained at $\mathrm{SNR}_{\mathrm{train}}$ = 2dB, 6dB, 10dB, 14dB, 18dB, respectively, and BDJSCC-10 the strategy involving using ten models trained at $\mathrm{SNR}_{\mathrm{train}}$ = 1dB, 3dB, 5dB, 7dB, 9dB, 11dB, 13dB, 15dB, 17dB, 19dB, respectively. Table I compares the performance/storage demands associated with each of these strategies. Note that BDJSCC-1 has a slightly lower storage overhead than ADJSCC, but it considerably underperforms ADJSCC by 5.3 dB. The BDJSCC-10 has similar performance with the ADJSCC. However, the storage overhead is 10 times that of the ADJSCC.

Then we evaluate the computational complexity of the ADJSCC model and the BDJSCC model on the Linux Server we mentioned in Section IV. With a training mini-batch of 128 images from CIFAR-10 training dataset, the mean training time of the ADJSCC model for a batch is almost 114 ms whereas the mean training time of the BDJSCC model for a batch is almost 110 ms. The inference time for the ADJSCC model takes 53 ms, compared to 49 ms for the BDJSCC model on CIFAR-10 dataset images. In summary, the training time of the ADJSCC model is 3.6% higher than that of the BDJSCC model and the inference time of the ADJSCC model is 8.1% higher than that of the BDJSCC model. However, note again that one would in practice have to use multiple BDJSCC models in order to achieve a performance comparable to the ADJSCC model, as discussed earlier, so, all in all, our proposed approach exhibits a much better computational/storage complexity that BDJSCC.

VI. CONCLUSION

In this work, we have proposed a novel ADJSCC method based on attention mechanisms that can adapt automatically to various channel conditions. It exhibits better performance, computational complexity, and storage complexity than existing approaches, making it an ideal candidate for JSCC in practical wireless communications scenarios.

We have proposed the ADJSCC method to be built upon two types of modules: the FL modules and the AF modules. The FL module is a general module that can utilize the existing module designed in the existing DJSCC work. The AF module takes the context information to generate the scaling factors by using the channel-wise soft attention and then recalibrates the channel-wise features. The motivation of our ADJSCC method originates from the resource assignment strategy in the traditional concatenated source channel coders.

We have evaluated the proposed ADJSCC method on AWGN channel, channel mismatch and large dataset to demonstrate the adaptability, the robustness and the versatility of the ADJSCC method. Lastly, we have compared the storage overhead and computational complexity of the ADJSCC method with that of the BDJSCC method. To achieve the same performance (PSNR), ADJSCC only needs 10.06% of the storage required by BDJSCC-10 and 10.36% of the training time required by BDJSCC-10.

In future work, a potential direction is to extend the proposed method for high-definition images and real wireless channels. This will play an important part in promoting the DL based JSCC technology for practical wireless communication systems.

REFERENCES

[1] T. M. Cover, *Elements of information theory*. Hoboken, NJ, USA: Wiley, 1999.

[2] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[3] R. G. Gallager, *Information theory and reliable communication*. New York, NY, USA: Springer, 1968, vol. 2.

[4] I. Csiszár, "Joint source-channel error exponent," *Problems of Control & Information Theory*, vol. 9, pp. 315–328, 1980.

[5] Y. Zhong, F. Alajaji, and L. L. Campbell, "Joint source–channel coding error exponent for discrete communication systems with markovian memory," *IEEE transactions on information theory*, vol. 53, no. 12, pp. 4457–4472, 2007.

[6] ——, "Error exponents for asymmetric two-user discrete memoryless source-channel systems," in *2007 IEEE International Symposium on Information Theory*. IEEE, 2007, pp. 1736–1740.

[7] B. Belzer, J. D. Villasenor, and B. Girod, "Joint source channel coding of images with trellis coded quantization and convolutional codes," in *Proceedings., International Conference on Image Processing*, vol. 2. IEEE, 1995, pp. 85–88.

[8] S. Heinen and P. Vary, "Transactions papers source-optimized channel coding for digital transmission channels," *IEEE transactions on communications*, vol. 53, no. 4, pp. 592–600, 2005.

[9] J. Cai and C. W. Chen, "Robust joint source-channel coding for image transmission over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 962–966, 2000.

[10] T. Guionnet and C. Guillemot, "Joint source-channel decoding of quasiarithmetic codes," in *Data Compression Conference, 2004. Proceedings. DCC 2004*. IEEE, 2004, pp. 272–281.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[12] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.

[15] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[16] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 771–10 780.

[17] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," *arXiv preprint arXiv:2007.08739*, 2020.

[18] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[19] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 3007–3018, 2017.

[20] D. Mishra, S. K. Singh, and R. K. Singh, "Wavelet-based deep auto encoder-decoder (wdaed) based image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[21] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Multiple description convolutional neural networks for image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2494–2508, 2018.

[22] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[23] J. Xu, W. Chen, B. Ai, R. He, Y. Li, J. Wang, T. Juhana, and A. Kurniawan, "Performance evaluation of autoencoder for coding and modulation in wireless communications," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2019, pp. 1–6.

[24] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Learn codes: Inventing low-latency codes via recurrent neural networks," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[25] ——, "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels," in *Advances in Neural Information Processing Systems*, 2019, pp. 2758–2768.

[26] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 298–303.

[27] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–5.

[28] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2326–2330.

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

[30] D. B. Kurka and D. Gündüz, "Successive refinement of images with deep joint source-channel coding," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[31] ——, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[32] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, "Neural joint source-channel coding," in *International Conference on Machine Learning*, 2019, pp. 1182–1192.

[33] A. Mnih and D. J. Rezende, "Variational inference for monte carlo objectives," *arXiv preprint arXiv:1602.06725*, 2016.

[34] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Deep joint source-channel coding for wireless image retrieval," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5070–5074.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] C.-H. Lee, J.-W. Lin, P.-H. Chen, and Y.-C. Chang, "Deep learning-constructed joint transmission-recognition for internet of things," *IEEE Access*, vol. 7, pp. 76 547–76 561, 2019.

[37] D. Burth Kurka and D. Gündüz, "Joint source-channel coding of images with (not very) deep learning," in *International Zurich Seminar on Information and Communication (IZS 2020). Proceedings*. ETH Zurich, 2020, pp. 90–94.

[38] M. Loiacono, J. Johnson, J. Rosca, and W. Trappe, "Cross-layer link adaptation for wireless video," in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–6.

[39] P. Zhao, Y. Liu, J. Liu, A. Argyriou, and S. Ci, "Ssim-based error-resilient cross-layer optimization for wireless video streaming," *Signal Processing: Image Communication*, vol. 40, pp. 36–51, 2016.

[40] K. Sayood, H. H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Transactions on Communications*, vol. 48, no. 5, pp. 787–794, 2000.

[41] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[42] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[44] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[45] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.

[46] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[47] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[48] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *arXiv preprint arXiv:1511.06281*, 2015.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[50] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[51] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

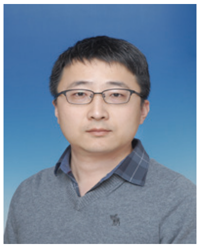
[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[53] N. Sprljan, M. Mrak, and E. Izquierdo, "A fast error protection scheme for transmission of embedded coded images over unreliable channels and fixed packet size," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 3. IEEE, 2005, pp. iii–741.

Jialong Xu (Student Member, IEEE) received the B.E. and M.S. degrees from Engineering University of PAP in 2009 and 2012 respectively. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China. His research interests include deep learning, wireless coding and information theory.

Bo Ai (Senior Member, IEEE) received the M.S. and Ph.D.degrees from Xidian University, Xian, China, in 2002 and 2004, respectively. He was with Tsinghua University, Beijing, China, where he was an Excellent Postdoctoral Research Fellow in 2007. He is currently a Professor and an Advisor of Ph.D.candidates with Beijing Jiaotong University, Beijing, where he is also the Deputy Director of the State Key Laboratory of Rail Traffic Control and Safety. He is also currently with the Engineering College, Armed Police Force, Xian. He has authored or coauthored six books and 270 scientific research papers, and holds 26 invention patents in his research areas. His interests include the research and applications of orthogonal frequency-division multiplexing techniques, high-power amplifier linearization techniques, radio propagation and channel modeling, global systems for mobile communications for railway systems, and long-term evolution for railway systems.

Dr. Ai is a Fellow of The Institution of Engineering and Technology. He was as a Co-chair or a Session/Track Chair for many international conferences such as the 9th International Heavy Haul Conference (2009); the 2011 IEEE International Conference on Intelligent Rail Transportation; HSRCom2011; the 2012 IEEE International Symposium on Consumer Electronics; the 2013 International Conference on Wireless, Mobile and Multimedia; IEEE Green HetNet 2013; and the IEEE 78th Vehicular Technology Conference (2014). He is an Associate Editor of IEEE TRANSACTIONS ON CONSUMER ELECTRONICS and an Editorial Committee Member of theWireless Personal Communications journal. He has received many awards such as the Qiushi Outstanding Youth Award by HongKong Qiushi Foundation, the New Century Talents by the Chinese Ministry of Education, the Zhan Tianyou Railway Science and Technology Award by the Chinese Ministry of Railways, and the Science and Technology New Star by the Beijing Municipal Science and Technology Commission.



Wei Chen (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in communications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from the University of Cambridge, Cambridge, U.K., in 2013. He was a Research Associate with the Computer Laboratory, University of Cambridge from 2013 to 2016. He is currently a Professor with Beijing Jiaotong University, Beijing. His current research interests include sparse representation, Bayesian inference, wireless communication systems and image processing. He was the recipient of the 2013 IET Wireless Sensor Systems Premium Award and the 2017 International Conference on Computer Vision (ICCV) Young Researcher Award.

Ang Yang received the B.S. and Ph.D. degree in electronic engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2009 and 2015, respectively. From 2015 to 2018, he worked as an algorithm engineer with Samsung Beijing Research Center. Since 2018, he has been a standard engineer with vivo Communication Research Institute. His research interests include massive multiple-input–multiple-output systems, millimeter wave and artificial intelligence.

Peng Sun received the B.S., M.S. and Ph.D. degree in electronic engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2006, 2009 and 2013 respectively. From 2012 to 2017 June, he worked as a PHY/MAC system engineer with Beijing Xinwei Telecommunication Company. Since 2017 July, he has been a standard expert with vivo Communication Research Institute. His research interests include system design and standardization for coding and modulation, MIMO, millimeter wave, artificial intelligence, etc.

Miguel Rodrigues (Senior Member, IEEE) received the Licenciatura degree in electrical and computer engineering from the University of Porto, Porto, Portugal, and the Ph.D. degree in electronic and electrical engineering from the University College London (UCL), London, U.K. He is currently a Professor of Information Theory and Processing, UCL, and a Turing Fellow with the Alan Turing Institute - the UK National Institute of Data Science and Artificial Intelligence. His research lies in the general areas of information theory, information processing, and machine learning. His work has led to more than 200 articles in leading journals and conferences in the field, a book on Information-Theoretic Methods in Data Science (Cambridge Univ. Press), and the IEEE Communications and Information Theory Societies Joint Paper Award 2011. He is an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY, and the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY. He was an Associate Editor for the IEEE COMMUNICATIONS LETTERS, and a Lead Guest Editor of the Special Issue on “Information-Theoretic Methods in Data Acquisition, Analysis, and Processing” of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He was a Co-Chair of the Technical Programme Committee of the IEEE Information Theory Workshop 2016, Cambridge, U.K. He is a member of the IEEE Signal Processing Society Technical Committee on “Signal Processing Theory and Methods”, and the EURASIP SAT on Signal and Data Analytics for Machine Learning (SiG-DML).