

Imperial College of Science, Technology and Medicine
Department of Electrical and Electronic Engineering

Deep Joint Source-Channel Coding for Semantic Communication

Haotian Wu

Supervised by Prof. Deniz Gündüz and Prof. Krystian Mikolajczyk

Submitted in part fulfillment of the requirements for the degree of PhD in Electrical and
Electronic Engineering and the Diploma of Imperial College London

June 2025

Abstract

The evolution of artificial intelligence applications has exponentially increased the demand for efficient and low-latency transmission of large data volumes, posing significant challenges to conventional wireless communication systems. While conventional systems, known for simplicity and elegance, have been extensively studied with a focus on reliable bit sequence transmission without considering the content semantics and communication objectives, they can yield sub-optimal results in practical scenarios characterized by finite block lengths and rapidly varying wireless channel conditions. Next-generation communication systems aim to address these limitations by incorporating message semantics and specific communication objectives into their design, a paradigm known as semantic communication. One of the most promising advancements in this area is the deep joint source and channel coding (DeepJSCC), a pioneering deep learning-based technique that combines source compression and error correction into a unified process. This approach allows end-to-end optimization and promises significant improvements in bandwidth efficiency and reliability, thereby challenging Shannon’s separation theorem.

This thesis delves into the transformative potential of DeepJSCC in the realm of semantic communications, with a particular focus on developing practical DeepJSCC-based transceivers for image transmission and edge inference tasks. By exploring the DeepJSCC for image transmission across three canonical channel models: orthogonal frequency division multiplexing (OFDM), multiple-input and multiple-output (MIMO), and feedback channels, this thesis showcases DeepJSCC’s superiority, adaptability, efficiency, and robustness in diverse semantic communication scenarios. Furthermore, this thesis extends the application of DeepJSCC to edge inference tasks, showcasing its ability to optimize different communication objectives. The comprehensive evaluation of the proposed DeepJSCC frameworks across diverse tasks and scenarios underscores its versatility, establishing it as a highly promising alternative to traditional separation-based algorithms and a potential solution for next-generation semantic communication systems.

Statement of Originality

I declare that the work included in this thesis is solely my own work other than where appropriately referenced.

Haotian Wu

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

First of all, I am deeply grateful to my supervisors, Deniz and Krystian, for giving me the opportunity to be their student and for their attentive guidance throughout my PhD. Beyond their invaluable academic support, they have set an example with their integrity, rigor, and dedication. Their constant encouragement to pursue solid, meaningful work has not only strengthened my research but also inspired me to embrace challenges. I would also like to thank my examiners, Paolo and Pier Luigi, for organizing my viva and for providing constructive feedback and thoughtful suggestions.

From the very beginning of my academic journey, our senior lab members, Yulin and Qianqian, supported me with great care and generosity. Their kindness left a lasting impression on me and motivated me to follow their example in supporting our junior group members in the future. Later, I was fortunate to meet a group of cool friends. Shout out to my incredible teammates in the IPC Lab: Arpan, Mohammad, Meng, Burak, Nitish, Emre, Selim, Marcello, Yassine, Szymon, Anastasiia, Yuxuan, Gongpu, Ruiqing, Yukuan, Jiancheng, Jiangjing, Wenjing, Bingxuan, Li, Chenghong, Edoardo, Taowen, Weichen, and Gen. I'm also grateful for the amazing company from the MatchLab members: Axel, Michal, Mikolaj, Adrian, Dylan, Roy, Benedikt, Ranran, Maojun, Junpeng, Ye, and Weixun. Thank you all for making this journey so much more enjoyable, and for the many memorable moments we shared.

Finally, I would like to thank my family, to whom this dissertation is dedicated. To my parents, Hongzhen and Chengfeng: thank you for always being there for me. To my life partner, Yuxin: thank you for your unwavering support and for always encouraging me to pursue what I love, even when it meant taking risks. Special thanks to my mother-in-law, Xingfen, whose care has been invaluable to our family. Her warmth and tenacity continue to inspire me. To our beloved son, Yining: thank you for coming into our lives. I imagine you'll be mentioned in many acknowledgments in the future, but as your father, I'm proud that this is the very first. Lastly, to myself: thank you for taking on the challenge and choosing a different path. I still think back to the moment one afternoon when our group member said to me, "The best time to plant a tree was ten years ago. The second-best time is now." May you continue to pursue solid and meaningful work in the future.

List of Acronyms

AE	Auto-encoder
AI	Artificial Intelligence
AR	Augmented Reality
AWGN	Additive White Gaussian Noise
BLER	Block Error Rates
BPG	Better Portable Graphics
CNNs	Convolutional Neural Networks
CPU	Central Processing Unit
CP	Cyclic Prefix
CSI	Channel State Information
DAE	Deep Auto-encoder
DeepJSCC	Deep Joint Source-Channel Coding
DDIB	Deterministic Distributed Information Bottleneck
DL	Deep Learning
DNNs	Deep Nerual Networks
FC	Fully Connected
FFT	Fast Fourier Transform
FLOPs	Floating-point Operations
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things

JSCC Joint Source-Channel Coding

LDPC Low-Density Parity-Check

LLMs large language models

LPIPS Learned Perceptual Image Patch Similarity

LS Least Square

MAC Multiple Access Channel

mAP mean Average Precision

MIMO Multiple-Input and Multiple-Output

ML Machine Learning

MLP Multi Layer Perceptron

MMSE Minimum Mean Square Error

MSA Multi-head Self-Attention

NOMA Non-Orthogonal Multiple Access

OFDM Orthogonal Frequency-Division Multiplexing

OMA Orthogonal Multiple Access

OTA Over-The-Air

PReLU Parametric Rectified Linear Unit

PSNR Peak Signal to Noise Ratio

R-D Rate-Distortion

re-ID Re-IDentification

ReLU Rectified Linear Unit

SA Self-Attention

SNR Signal-to-Noise Ratio

SOTA State-Of-The-Art

SVD Singular Value Decomposition

TDMA Time Division Multiple Access

ViT Vision Transformer

VR Virtual Reality

ZF Zero-Forcing

Contents

Abstract	3
Acknowledgements	9
List of Tables	21
List of Figures	23
1 Introduction	31
1.1 Challenges and motivations	38
1.2 Outline and contributions	39
1.3 Publications	41
2 Background and related works	44
2.1 From Shannon’s separation theorem to Deep JSCC	44
2.2 From technical to semantic communication	50
3 DeepJSCC for adaptive image transmission over OFDM channels	56
3.1 Introduction	56

3.2	System Model	57
3.3	Proposed method	60
3.3.1	Channel-wise attention module	61
3.3.2	Spatial attention module	62
3.3.3	Design interpretation	62
3.4	Training and evaluation of the CA-JSCC	64
3.4.1	Channel-gain adaptability	65
3.4.2	SNR adaptability	66
3.4.3	Impact of CSI estimation errors	68
3.4.4	Ablation study	68
3.5	Conclusion	71
4	DeepJSCC for adaptive image transmission over MIMO channels	72
4.1	Introduction	72
4.2	System Model	75
4.2.1	Open-loop MIMO with CSIR	77
4.2.2	Closed-loop MIMO with CSIT	78
4.3	Proposed method	80
4.3.1	Image-to-sequence transformation	81
4.3.2	Channel heatmap construction	82
4.3.3	ViT encoding	83
4.3.4	ViT decoding	84

4.3.5	Loss function	88
4.4	Training and evaluation	88
4.4.1	Experimental setup	88
4.4.2	Open-Loop MIMO system with CSIR	91
4.4.3	Closed-Loop MIMO system with CSIT	96
4.5	Conclusion	105
4.5.1	Future work	105
5	DeeJSCC for feedback-aided semantic communication	106
5.1	Introduction	106
5.2	System Model	110
5.3	JSCCformer-f Architecture	112
5.3.1	JSCCformer-f encoder	113
5.3.2	JSCCformer-f decoder	117
5.3.3	Loss function	119
5.4	Training and evaluation	119
5.4.1	Transmission performance	120
5.4.2	Adaptability	127
5.4.3	Model efficiency	130
5.4.4	Variable Rate Transmission	131
5.4.5	JSCCformer-f for broadcast channels	133
5.5	Conclusion	135

5.5.1	Future work	136
6	DeepJSCC empowered edge inference for semantic communication	137
6.1	Introduction	137
6.2	Related work	142
6.3	System model	144
6.4	Methodology	147
6.4.1	Overall workflow	149
6.4.2	JSCC transmission	150
6.4.3	The CL module	151
6.4.4	Training strategies	155
6.4.5	Design interpretation	156
6.5	Training and evaluation	158
6.5.1	Experimental setup	159
6.5.2	CL-SC scheme in OMA transmission	160
6.5.3	CL-SC scheme in NOMA transmission	162
6.5.4	CL-SC scheme for $M=3$	166
6.5.5	Result visualization	167
6.6	Conclusion	168
6.6.1	Future work	168
7	Conclusion	170
7.1	Future work	171

List of Tables

4.1	Overview of alternative end-to-end MIMO communication pipelines.	74
4.2	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.	92
4.3	Comparisons of DeepJSCC-MIMO-universal and BPG-LDPC schemes with sphere decoding operations under channel estimation errors over the Kodak dataset for $R = 1/24$	94
4.4	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	95
4.5	Ablation study of DeepJSCC-MIMO-universal with/without channel heatmap over the CIFAR10 dataset with CSIR.	100
4.6	Ablation study of DeepJSCC-MIMO-universal with/without channel heatmap over the CIFAR10 dataset with CSIT.	100
4.7	Ablation study of DeepJSCC-MIMO-universal scheme with/without channel heatmap on the CIFAR10 dataset across various SNRs in the fast fading channel when $R = 1/48$	101
5.1	Performance of models over the Kodak dataset.	125

5.2	Ablation study over different training strategies (fixed SNR or random SNR), where $m = 4$ and $R = 1/3$	127
5.3	Ablation study of the Siamese layer on the CIFAR10 dataset, $R = 1/6$, $m = 2$. . .	129
5.4	The number of parameters (millions), FLOPs (G), and coding time (ms) for different models; bold figures correspond to the minimal values among different models for each m value.	130
6.1	Top1-accuracy of different schemes across various SNR values, where $B = 60$ and $M = 3$	167

List of Figures

1.1	Illustration of wireless edge applications.	32
1.2	Alternative wireless transmission pipelines.	33
1.3	An example of DeepJSCC-empowered semantic communication system with multiple communication objectives, where the whole transmitter can be viewed as a DeepJSCC encoder.	36
1.4	Illustration of DeepJSCC and semantic communication: (a) DeepJSCC design accounting for source and channel characteristics to enhance semantic communication. (b) Key considerations in designing DeepJSCC for semantic communication.	37
2.1	A comparison between conventional separation-based scheme and JSCC scheme, where source $\mathbf{S}$ is transmitted as a channel input $\mathbf{X}$, and is reconstructed as $\hat{\mathbf{S}}$ based on the channel output $\mathbf{Y}$	45
2.2	Illustrations of Weaver’s three levels communication problems.	51
3.1	Block diagram of the wireless image transmission system: (a) the conventional separation-based scheme and (b) the deep learning based JSCC scheme.	58
3.2	Illustration of the proposed CA-JSCC scheme, where the encoder maps the input image $\mathbf{S}$ and estimated CSI $\hat{\mathbf{h}}$ to the channel input matrix $\mathbf{X}$, and the decoder reconstructs the source as $\hat{\mathbf{S}}$	59

3.3	Basic blocks of our dual-attention encoder and decoder architectures, where both the encoder and decoder are composed of spatial attention blocks and channel attention blocks arranged in a comb structure.	60
3.4	Visualization of FL process.	63
3.5	Visualization of CL process.	63
3.6	(a) Comparison of CA-JSCC scheme with Exp-JSCC when the values of the train and test SNRs are the same. (b) Ablation experiments for the attention strategy. (c) Visualization of the power allocation for the CA-JSCC and the Exp-JSCC.	65
3.7	Performance of CA-JSCC model compared with the Exp-JSCC model of different bandwidth ratios.	66
3.8	(a) Additional experiments with training on the ImageNet dataset tested on the Kodak dataset. (b) Comparison of CA-JSCC with different channel estimation methods.	67
3.9	Different structures of CA-JSCC.	68
3.10	Power allocation for CA-JSCC with different channel gain distribution.	69
3.11	Different structures trained with the random SNRs.	70
3.12	Power allocation for CA-JSCC-CH with different channel gain distribution. . . .	71
4.1	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.	77

4.2	The pipeline of the DeepJSCC-MIMO scheme, where the source image $\mathbf{S}$ is encoded by a ViT-encoder and reconstructed by a ViT-decoder as $\hat{\mathbf{S}}$. Precoding operation in dashed line will be performed if the CSI is available at the transmitter, and the CSI $(\mathbf{H}, \sigma_w^2)$ is fed to the encoder, if available, and decoder in the form of a “heatmap” $\mathbf{M}$ to facilitate the JSCC encoding/decoding process. .	81
4.3	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.	82
4.4	Structure of DL-aided channel equalization method, where the input at the i -th channel use is the concatenation of all elements within $\mathbf{H}$ and the channel output	85
4.5	Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the MIMO systems with CSIR.	89
4.6	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. .	90
4.7	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.	92
4.8	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.	96

4.9	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, MBT2018-Capacity, and Cheng2020-Capacity schemes, respectively. The last two columns demonstrate the patch recovered by the DeepJSCC-MIMO-universal and DeepJSCC-MIMO schemes.	98
4.10	Performance of DeepJSCC-MIMO and conventional separation-based scheme in the CelebA dataset when $R = 1/48$	99
4.11	Visualization of the channel heatmap and power allocation across the channel symbols, where antenna number is set as $M = 32$ and $R = 1/6$	102
4.12	Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the open-loop Kronecker MIMO channels with CSIR.	102
4.13	Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the closed-loop Kronecker MIMO channels with CSIT.	103
4.14	Performance of DeepJSCC-MIMO with different antennas number.	104
5.1	Illustration of alternative JSCC schemes for channels with feedback, where the solid lines represent forward channels, while the dotted lines represent feedback channels. Top: Illustration of the DeepJSCC-f scheme (with $m = 3$ blocks as an example), where multiple independent encoders and decoders are trained as the channel codes in each block. Bottom: Illustration of our proposed JSCCformer-f scheme, where a single unified encoder-decoder pair is employed at each block, significantly reducing the training and memory requirements as well as the coding complexity.	107

5.2	The pipeline of our JSCCformer-f scheme. In the i -th transmission block, the ViT-based encoder E_θ encodes the channel symbols $\mathbf{X}_i$ based on the input image and the channel feedback signals received until that time, where $\hat{\mathbf{Y}}_{i-1}$ is the noisy channel feedback for the $i - 1$ -th transmission and $\mathbf{Z}_{i-1}$ is its embedding. Note that $(\hat{\mathbf{Y}}_i, \dots, \hat{\mathbf{Y}}_m)$ and $(\mathbf{Z}_i, \dots, \mathbf{Z}_{m-1})$ are padded with zeros as they correspond to future channel feedback signals.	113
5.3	The architecture of the encoder and decoder, where a symmetric structure is designed to encode the input sequence and reconstruct the source signal.	116
5.4	Performance of different schemes at various SNR values and bandwidth ratios with noiseless feedback, where models in subfigures (a) and (b) undergo assessment within the AWGN channel, and models in subfigure (c) are evaluated in a Rayleigh Fading channel.	120
5.5	(a) Performance comparison of different models versus bandwidth ratio R in AWGN channel when SNR= 10 dB with noiseless feedback. (b) Performance of models over different block numbers and SNRs in AWGN channel with $R = 1/2$, where the optimal block number is highlighted with a red star.	122
5.6	Performance of JSCCformer-f over noisy feedback link in AWGN channel, where $R = 1/6$, $m = 2$	123
5.7	LPIPS performance of different schemes, where models are trained with $m = 2$	124
5.8	Visual comparisons of images transmitted by DeepJSCC-f, JSCCformer-f (lite), and JSCCformer-f over the AWGN channel, where the model is trained on the ImageNet dataset and validated on the Kodak dataset at various SNR values and $R = 1/12$	125
5.9	Visualization of the reconstructed image with JSCCformer-f after each interaction for an input image from the Kodak dataset, where the model is trained on the ImageNet dataset with $R = 1/12$, and $SNR = -2$ dB.	126

5.10	Visualization of the reconstructed image with JSCCformer-f after each interaction for images from the CelebA dataset, where the model is trained on the ImageNet dataset with $R = 1/12$, and $SNR = -2$ dB.	127
5.11	(a) Performance of JSCCformer-f (lite) compared with other schemes in AWGN channel, where $R = 1/6$, and $m = 2$. (b) Performance of JSCCformer-f (lite) trained with random SNRs over noisy feedback link in the AWGN channel, where $R = 1/6$ and $m = 2$	128
5.12	Target PSNR analysis of different schemes for variable rate transmission. (a) The cumulative distribution function over various bandwidths given a specific target PSNR. (b) The average bandwidth ratio to achieve a specific target PSNR.	133
5.13	Schematic illustration of extending our JSCCformer-f scheme into broadcast channels.	133
5.14	(a) PSNR performance region from different schemes, where $R = 1/6$, $\mu_1 = 4$ dB and $\mu_2 = 7$ dB . (b) Average PSNR performance of JSCCformer-f (broadcast) for different channel SNR pairs.	134
6.1	(a) Collaborative edge retrieval task. (b) CL for collaborative edge retrieval: the dotted and solid rectangles represent the information contained in each observation and the shared identity-related information between two observations, respectively; the common identity-related features are represented by the orange points, while the inconsistent features are represented by other colors. We expect CL module to help $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ to be more discriminative with maximum semantic consistency and identity-related features.	138

6.2	(a) Schematic of the proposed CL-SC scheme. This framework aims to refine the discriminative power of noisy received features by optimizing their representations in the semantic feature space. The optimization process specifically focuses on minimizing the feature distance between samples that share identical identities, while simultaneously maximizing the distance between samples that have different identities, enhancing the robustness and accuracy of identity matching.	
	(b) Detailed architecture of the JSCC encoder and decoder used in our system.	145
6.3	Contrastive learning module for M devices, where $(k, k, c s)$ means convolutional operation with (k, k) kernel, out channel number c , stride s , and sg means stop-gradient operation.	149
6.4	Design interpretation of our cross-view CL module, where the dotted rectangles denote the information (feature space) contained in the decoded view $\mathbf{z}_{\theta_1}$ and $\mathbf{z}_{\xi_2}$, the solid rectangle denotes the overlapped information (identity-related semantics). The blue notation means the operations after exchanging the input views for the online and target networks during the training phase.	156
6.5	Top-1 retrieval accuracy of the different schemes for the OMA transmission over the AWGN and Fading channel across various SNR and bandwidth values.	160
6.6	Over-the-air superposition of feature vectors from a cluster increases the contrast between different class clusters, where $\mathbf{x}_1, \mathbf{x}_2$ are two data points belonging to the same cluster, while $\mathbf{x}_3$ and $\mathbf{x}_4$ correspond to two other data points in another cluster.	163
6.7	Top-1 retrieval accuracy of the different schemes for the NOMA transmission over the AWGN and Fading channel across various SNR and bandwidth values.	164
6.8	Performance comparison between OMA and NOMA schemes for the CL-SC.	166
6.9	Visualization of the features from two-source JSCC and CL-SC module, where $B = 64$ and $SNR = 0$ dB.	167

Chapter 1

Introduction

The rapid expansion of wireless communication technologies has revolutionized the way information is transmitted and received. The exponential growth of wireless applications, such as the Internet of Things (IoT) [1], virtual reality (VR) [2], real-time streaming [3] and edge intelligence [4], has significantly amplified the demand for reliable and efficient transmission of large data volumes, as depicted in Fig. 1.1. Transmitting such vast amounts of data over wireless channels with low latency and high reliability poses substantial challenges, primarily due to bandwidth constraints, the data-intensive nature, and the inherent variability of wireless channels. Addressing these challenges requires the development of advanced techniques such as source compression [5, 6, 7], channel coding [8, 9], and transceiver optimization for specific downstream tasks [10].

Traditional Shannon separation theorem One starting point is reconsidering the Shannon separation theorem in practical scenarios. The Shannon separation theorem [11] states that source and channel coding can be optimized independently without losing efficacy in the asymptotic limit of infinite block-length for ergodic source and channel distributions. Building on Shannon’s theorem, research into traditional separation-based communication systems has predominantly focused on two critical areas: source coding and channel coding, as illustrated in Fig. 1.2. The former aims to compress the source data into fewer bits, while the latter seeks

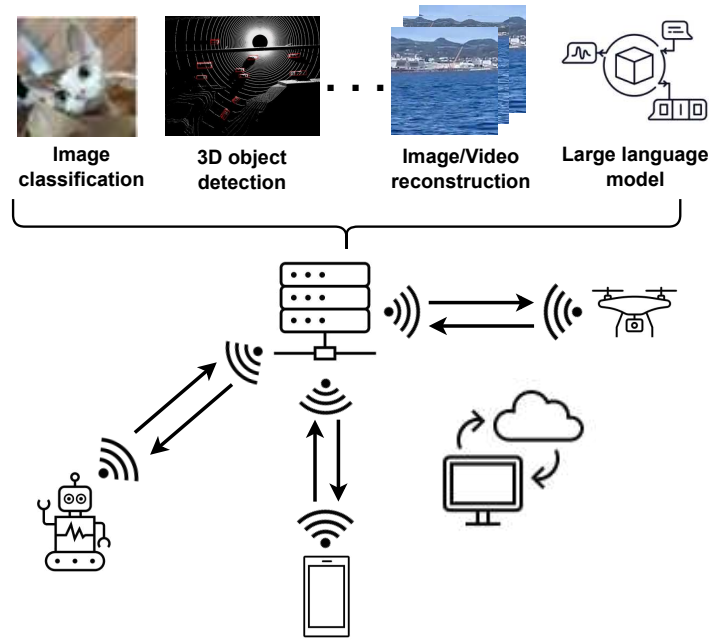


Figure 1.1: Illustration of wireless edge applications.

to maintain robustness against channel noise at a lower coding rate. Both areas have been extensively investigated, especially in the past decades, due to the increasing prominence and advancements in machine learning.

While the Shannon separation theorem has long been a cornerstone of digital communications, its applicability in real-world scenarios is significantly limited. In modern wireless systems, characterized by finite block lengths and dynamically changing channels, this elegant theoretical principle can result in sub-optimal performance [12, 13]. These limitations prompt a reevaluation of conventional methodologies, suggesting that an integrated source and channel coding approach might better suit contemporary needs. The advent of joint source-channel coding (JSCC) and its further innovation through deep learning [14, 15, 16] exemplify these considerations. By intelligently combining the tasks of compression and error correction, JSCC approaches aim to mitigate the inefficiencies brought about by the theorem's assumptions, addressing the critical issues such as cliff degradation and sub-optimality that can significantly impact system performance under varying channel conditions.

Joint source and channel coding As shown in Fig. 1.2, unlike separation-based schemes that independently execute source and channel coding, a classical JSCC approach [13] directly

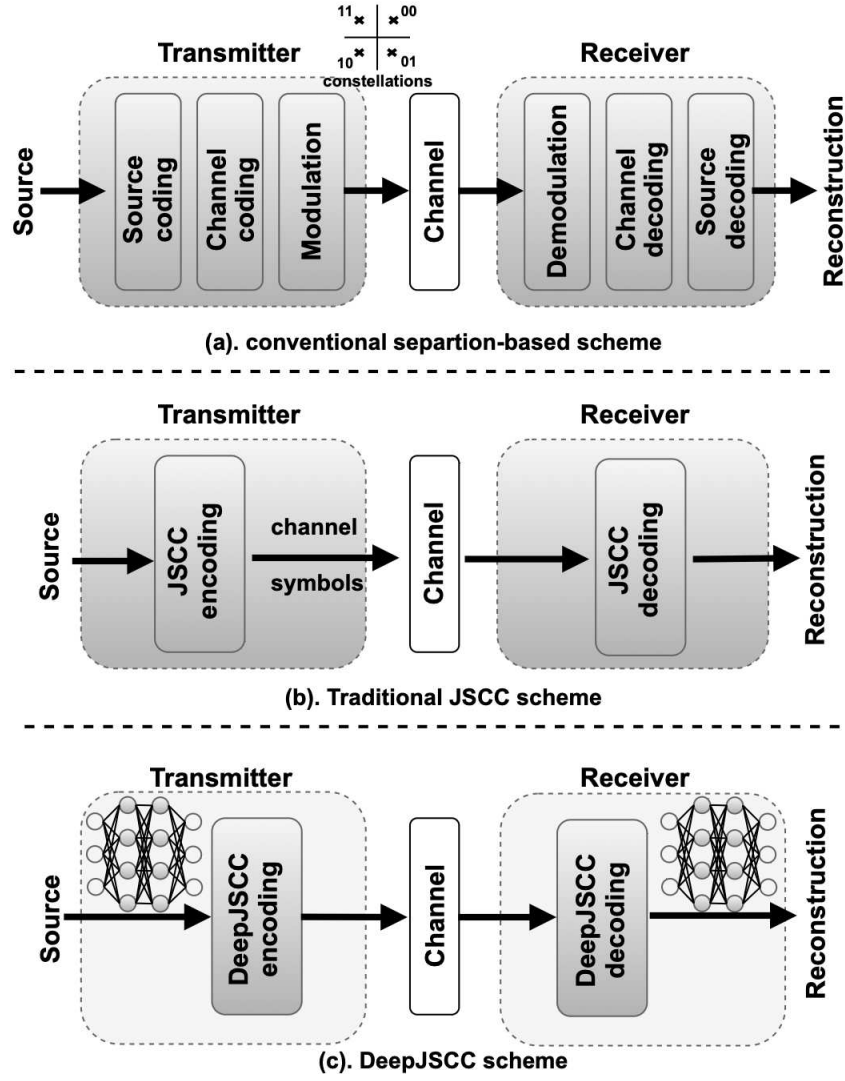


Figure 1.2: Alternative wireless transmission pipelines.

maps the source signal to the channel symbol at the transmitter. As a result, the receiver reconstructs the source signal directly from the received channel output. The primary advantages of JSCC-based systems include their ability to exhibit ‘graceful degradation,’ where the quality of reconstruction gradually diminishes as channel conditions deteriorate, and their superior performance under certain challenging channel conditions or in bandwidth-constrained scenarios [17, 18, 19].

While JSCC methods demonstrate advantages over traditional separation-based schemes in specific contexts, designing a practical JSCC scheme is extremely challenging. Most existing studies on JSCC either focus on theoretical scenarios, e.g., Gaussian or Gauss-Markov sources [20, 21, 22, 23], or jointly optimize parameters of separate source and channel codes [24, 25, 26,

27], 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 [28], particularly when dealing with practical sources such as images. Despite the significant computational and design complexity associated with optimizing source and channel coding configurations, to the best of the authors' knowledge, no classical JSCC schemes have been known that can perform on par with state-of-the-art separation-based codes.

The rise of artificial intelligence and DeepJSCC Simultaneously, artificial intelligence (AI) is increasingly recognized as a transformative force in improving connectivity, which is essential for developing the forthcoming 6G cellular network standard [29]. Notably, rapid advancements in AI technologies have demonstrated significant successes in the field of communication, offering innovative solutions across various domains such as source coding algorithms [5], channel state information (CSI) estimation [30], and channel coding techniques [31]. Leveraging these AI-driven methods holds considerable potential to substantially improve wireless communication systems' performance and efficiency, thereby enabling higher data rates, reduced latency, and greater efficiency across a broad spectrum of 6G applications.

Thanks to the remarkable advancements in AI, particularly in deep neural networks (DNNs) and deep learning (DL), significant progress has been made in the recent years in the design of practical and competitive DL-based joint source-channel coding (DeepJSCC) schemes [15, 32, 32, 33, 34, 35, 36, 37, 38]. As shown in Fig. 1.2, DeepJSCC schemes utilize an auto-encoder (AE) framework to directly map input source signals to channel symbols at the transmitter and vice versa at the receiver. This design eliminates the need for explicit source compression, error correction, or using bits as a standard representation unit. Instead, the DeepJSCC approach is optimized to directly generate channel symbols to enhance the overall communication performance across diverse source and channel distributions, which adopts an end-to-end training methodology without imposing constraints on the distribution of the source and the channel.

The first DeepJSCC was introduced for wireless image transmission in [15], which was shown

to outperform the concatenation of the better portable graphics (BPG) [39] image compression codec with low-density parity-check (LDPC) [40] codes. This was later extended to different communication channels, including orthogonal frequency division multiplexing (OFDM) [32, 41, 42], multiple-input and multiple-output (MIMO) [43, 44], relay [45], multiple-access [46], broadcasting [47], and feedback channels [35]. Moreover, the DeepJSCC framework is highly adaptable across a wide range of data modalities, which has demonstrated consistent transmission superiority in texts [14], videos [48], and speech [49]. The versatility and practicality of DeepJSCC, coupled with its effectiveness across diverse channel models and source modalities, positions it as a highly promising technology for next-generation communication systems [4, 50, 51].

Technical, semantic, and effective communication The foundational communication model, remaining widely utilized in contemporary research, was first introduced by Shannon in 1948 [11]. In this model, a transmitter encodes a message into channel symbols, which are sent through a communication channel and decoded by the receiver to reconstruct the original message. Based on this model, communication problems can be further categorized into three distinct levels [52]: Level A (technical), Level B (semantic), and Level C (effective). Level A, technical communication, concerns the precise transmission of communication symbols. Level B, semantic communication, ensures the intended semantic meaning is conveyed. Level C, effective communication, emphasizes the impact of the received message, aiming to influence the receiver in the desired manner.

Traditionally, research in existing communication has primarily focused on technical communication problems, ensuring the reliable and efficient delivery of data packets without considering the content of the packets or their impact on the receiver. This design can accurately transmit channel symbols, offering a universal solution applicable to diverse information sources. Such a technical communication approach is highly valued for its simplicity, adaptability, versatility, and compatibility with real-world systems.

However, there has been a growing interest in moving beyond traditional technical communica-

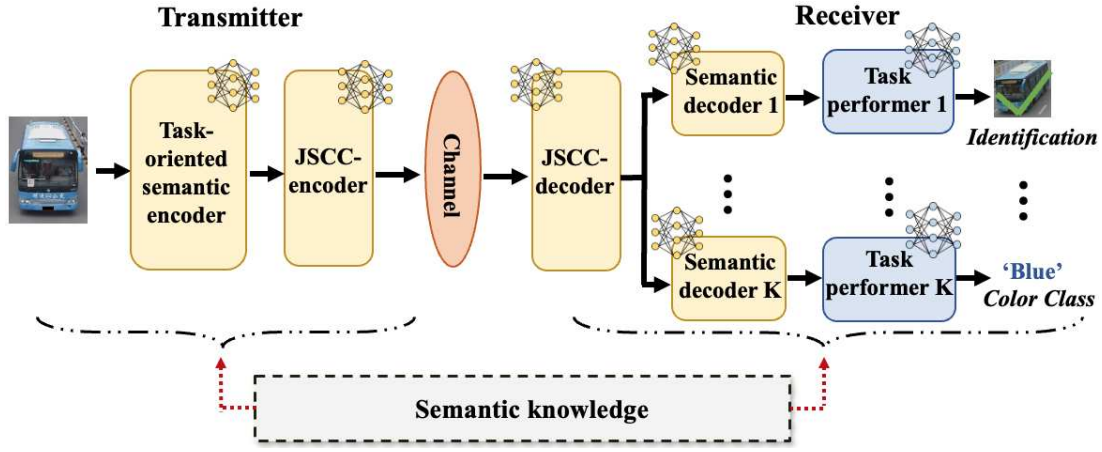


Figure 1.3: An example of DeepJSCC-empowered semantic communication system with multiple communication objectives, where the whole transmitter can be viewed as a DeepJSCC encoder.

tion systems into semantic communication systems in recent years. This shift is driven, in part, by the exponential increase in data traffic, the escalating demand for low-latency communication, and the potential performance improvement from fully utilizing the content semantics, all of which present significant challenges to conventional communication paradigms. Additionally, advancements in semantic-related metrics and theories and the innovative integration of various machine-learning techniques into semantic communication tasks have motivated the development of semantic communication. These advancements are especially relevant in the context of emerging AI applications, where the primary objective is to achieve task-specific performance or accurately convey the semantic features of the source for downstream tasks. Unlike traditional communication, which prioritizes high-fidelity message reconstruction, semantic communication emphasizes the accurate transmission of semantic meaning to enhance the overall functionality and efficiency of communication systems, particularly in goal-oriented, AI-driven applications.

DeepJSCC-empowered semantic communication Most semantic communication systems significantly benefit from the integration of deep learning technologies. The main reason is that directly extending theoretical findings from technical communication to semantic and effective communication levels can introduce substantial complexity and challenges. However, the advancements in DL for semantic tasks offer powerful tools for extracting source semantic or

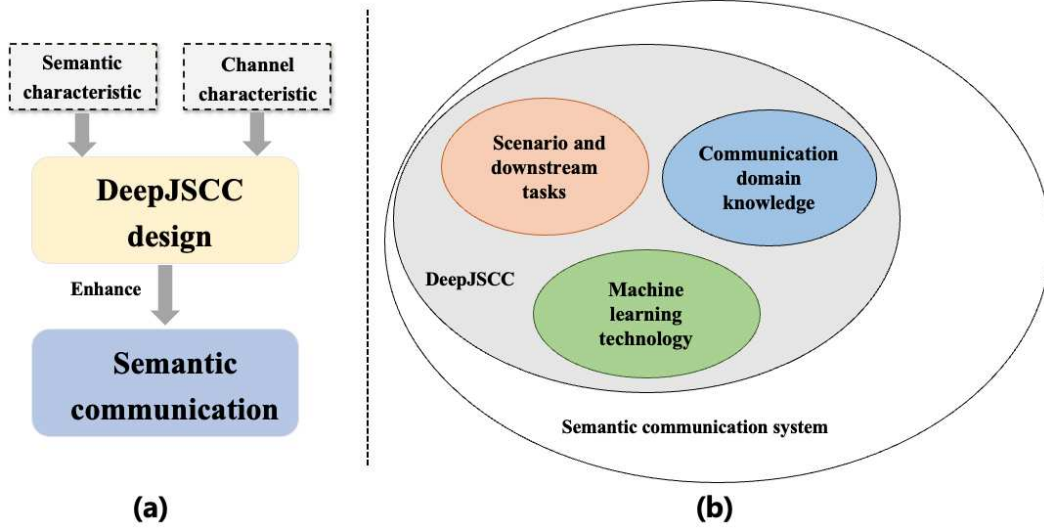


Figure 1.4: Illustration of DeepJSCC and semantic communication: (a) DeepJSCC design accounting for source and channel characteristics to enhance semantic communication. (b) Key considerations in designing DeepJSCC for semantic communication.

task-specific features customized for particular communication objectives, despite often lacking interpretability.

In fact, numerous semantic communication systems [53, 54, 55] can be modeled within a general DeepJSCC framework. While various semantic communication frameworks can be developed for distinct goals and scenarios, DeepJSCC is a foundational technology in this field, as depicted in Fig. 1.3. It facilitates goal-oriented optimization of the entire communication pipeline without the necessity of explicitly defining semantic meaning. This intrinsic versatility makes it highly adaptable for various semantic communication systems, especially those involving AI applications. Thus, utilizing DeepJSCC as the foundational module for designing semantic communication systems becomes highly natural and appropriate. To achieve this, it is crucial to integrate domain-specific knowledge from the field of communication with the inherent characteristics of DeepJSCC. This integration facilitates the development of a model-driven system tailored for semantic communication, as shown in Fig. 1.4. For instance, in source transmission tasks, DeepJSCC can prioritize delivering essential semantic features over channels with better conditions [41, 56]. Another example is edge inference tasks, where incorporating DeepJSCC into the pipeline can significantly improve the task performance by ensuring that only robust and essential features are transmitted [57]. These enhancements are all achieved by employing

DeepJSCC to consider both the characteristics of the wireless channel and the specific objectives of the task.

This proposed design, integrating the advantages of DeepJSCC, environment-specific communication channels, and intended communication objectives, demonstrates the substantial potential for enhancing and optimizing next-generation semantic communication systems.

1.1 Challenges and motivations

While DeepJSCC is a sophisticated and versatile paradigm, designing a practical DeepJSCC-based transceiver for semantic communication systems poses significant challenges, particularly in terms of source and channel generalizability. Addressing these practical challenges forms the central motivation of this thesis.

Source generalizability Many deep learning models face overfitting and limited generalization issues regarding source generalizability, often due to their training on specific datasets. More recently, overfitted compression approaches [7, 58, 59, 60, 61], including our own work [7, 61], have introduced new paradigms to address these challenges by reducing dependence on training data. While promising, these methods require increased training effort and are not practical in real-time communication scenarios. Instead, this thesis delves into the meticulous design of model architectures, encompassing convolutional neural networks (CNNs) to transformers, and the exploration of training strategies, such as end-to-end training and contrastive learning. The emphasis here is on careful architectural design to enhance the robustness and generalizability of these models. Employing sufficiently large and diverse datasets can also significantly improve the robustness and generalizability of these models. This thesis presents several meticulously designed architectures for different communication scenarios, each demonstrating strong generalization capabilities and competitive performance, even on unseen datasets.

Channel generalizability In parallel, ensuring channel generalizability for DeepJSCC schemes is also essential for the effectiveness and practicality of semantic communication systems. One critical shortcoming of previous works is the lack of consideration for essential factors regarding channels, such as the ability to adapt to diverse channel models, varying channel conditions, and the channel estimation errors. To achieve adaptability across a range of channel models and channel conditions, DeepJSCC must incorporate corresponding extensions designed for specific scenarios, enabling the semantic communication systems to scale efficiently to practical applications. Addressing this challenge involves designing a robust architecture and integrating domain-specific knowledge in communication channels. Moreover, DeepJSCC is expected to demonstrate the potential to maintain performance, even in the presence of noisy CSI or imperfect side information, by employing specific deep learning mechanisms.

Motivations In light of the above research gap, this thesis provides a detailed exploration of how modern communication systems can overcome traditional constraints by leveraging DeepJSCC technologies. Specifically, we present practical DeepJSCC schemes across different scenarios, demonstrating the transformative impact of DeepJSCC techniques on communication performance. These scenarios encompass adaptive image transmission over OFDM and MIMO channels, communication problems in the presence of feedback channels, and a DeepJSCC-empowered edge inference scenario. Each scenario poses unique challenges that push the boundaries of conventional communication systems. By incorporating DeepJSCC, this thesis reveals its potential to enhance performance, highlighting its adaptability, efficiency, and resilience across diverse communication environments.

1.2 Outline and contributions

The outline and contributions of this thesis can be summarized as follows:

- **Chapter 1** provides a comprehensive overview of recent advancements in DeepJSCC and semantic communication, highlighting the motivation behind it and detailing our key

contributions.

- **Chapter 2** presents the foundational concepts, essential background knowledge, and a literature review on DeepJSCC and semantic communication.
- **Chapter 3** presents a channel-adaptive DeepJSCC scheme for wireless image transmission over OFDM channels. Specifically, we propose a dual-attention mechanism that leverages the estimated CSI to aid the allocation of features and power resources, thereby adapting effectively to time-varying channel conditions. Numerical experiments verify that the proposed method achieves the state-of-the-art performance across a range of SNR and bandwidth scenarios, and shows robust performance even with imperfect CSI.
- **Chapter 4** presents the first DeepJSCC scheme over the MIMO system, DeepJSCC-MIMO, for adaptive image transmission. The proposed DeepJSCC-MIMO architecture significantly surpasses traditional separation-based benchmarks in both open-loop and closed-loop MIMO systems by leveraging the channel heatmap design and employing a self-attention mechanism. Furthermore, it exhibits remarkable robustness against channel estimation errors and exceptional adaptability to varying channel conditions and antenna configurations without requiring retraining. These attributes make it a highly promising solution for emerging semantic communication systems over MIMO channels.
- **Chapter 5** proposes JSCCformer-f, a novel feedback-aided JSCC architecture for wireless image transmission. JSCCformer-f exploits the semantics of the source image and the available feedback signal to generate coded symbols and refine the decoder's belief about the source image across multiple blocks. The proposed scheme is resource-efficient, using a single encoder for multiple feedback interactions without additional parameters or training, and supports flexible transmission strategies, including variable rate transmission, to optimize resource usage. Numerical experiments demonstrate that JSCCformer-f achieves the state-of-the-art performance across various SNR and bandwidth scenarios with robustness against feedback noise. Furthermore, it can be adapted for broadcasting channels, effectively managing feedback from multiple sources.
- **Chapter 6** investigates DeepJSCC for collaborative edge inference problem and proposes

a novel contrastive learning-based semantic communication paradigm. Specifically, we study how channel conditions impact the transmission of semantic information from multiple edge devices and optimize inference performance by leveraging correlations among transmitted signals, multiple access strategies, and integrating JSCC with over-the-air technologies. Numerical experiments show that the proposed scheme outperforms existing approaches, particularly in low SNR and limited bandwidth scenarios.

- **Chapter 7** summarizes the key conclusions and findings of this thesis.

1.3 Publications

The thesis is supported by the following publications:

- Haotian Wu, Yulin Shao, Krystian Mikolajczyk, Deniz Gündüz, “Channel-Adaptive Wireless Image Transmission with OFDM”, *IEEE Wireless Communications Letters*, 2022.
- Haotian Wu, Yulin Shao, Chenghong Bian, Krystian Mikolajczyk, Deniz Gündüz, “Vision Transformer for Adaptive Image Transmission over MIMO Channels”, *IEEE International Conference on Communications (ICC)*, 2023, **Best paper award**.
- Haotian Wu, Yulin Shao, Chenghong Bian, Krystian Mikolajczyk, Deniz Gündüz, “Deep joint source-channel coding for adaptive image transmission over MIMO channels”, *IEEE Transactions on Wireless Communications*, 2024.
- Haotian Wu, Yulin Shao, Emre Ozfatura, Krystian Mikolajczyk, Deniz Gündüz, “Transformer-aided wireless image transmission with channel feedback”, *IEEE Transactions on Wireless Communications*, 2024.
- Haotian Wu, Chenghong Bian, Yulin Shao, Deniz Gündüz, “Deep Joint Source and Channel Coding”, *Foundation of Semantic Communication Networks*, Wiley, 2025.

- Haotian Wu, Yulin Shao, Nitish Mital, Krystian Mikolajczyk, Deniz Gündüz, “Features-over-the-Air: Contrastive Learning Enabled Cooperative Edge Inference”, *under review*, 2025.
- Haotian Wu, Gongpu Chen, Pier Luigi Dragotti, Deniz Gündüz, “LotteryCodec: Searching the Implicit Representation in a Random Network for Low-Complexity Image Compression”, *2025 International Conference on Machine Learning (ICML)*, 2025, **Spotlight (Top 2.6%)**.
- Haotian Wu, Gongpu Chen, Deniz Gündüz, “Actions Speak Louder Than Words: Rate-Reward Trade-off in Markov Decision Processes”, *2025 International Conference on Learning Representations (ICLR)*, 2025, **(Top 3.27%)**.
- Haotian Wu, Maojun Zhang, Yulin Shao, Krystian Mikolajczyk, Deniz Gündüz, “MIMO Channel as a Neural Function: Implicit Neural Representations for Extreme CSI Compression”, *2025 International Conference on Acoustics, Speech, and Signal Processing (ICASP)*, 2025.

Other papers produced during the PhD study:

- Chenghong Bian, Yulin Shao, Haotian Wu, Deniz Gündüz, “Space-time design for deep joint source channel coding of images Over MIMO channels”, *IEEE International Workshop on Signal Processing Advances in Wireless Communications*, 2023.
- Wing Fei Lo, Nitish Mital, Haotian Wu, Deniz Gündüz, “Collaborative semantic communication for edge inference”, *IEEE Wireless Communications Letters*, 2023.
- Chenghong Bian, Yulin Shao, Haotian Wu, Deniz Gündüz, “Deep joint source-channel coding over cooperative relay networks”, *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 2024.
- Haotian Wu, Maojun Zhang, Yulin Shao, Krystian Mikolajczyk, Deniz Gündüz, “MIMO Channel as a Neural Function: Implicit Neural Representations for Extreme CSI Compression in Massive MIMO Systems”, *under review*, 2025.

- Ruiqing Mao, Haotian Wu, Yukuan Jia, Zhaojun Nan, Yuxuan Sun, Sheng Zhou, Deniz Gündüz, and Zhisheng Niu, “DiffCP: Ultra-Low Bit Collaborative Perception via Diffusion Model”, *International Conference on Robotics and Automation (ICRA)*, 2025.
- Chenghong Bian, Yulin Shao, Haotian Wu, Deniz Gündüz, “Process-and-Forward: Deep Joint Source-Channel Coding Over Cooperative Relay Networks”, *IEEE Journal on Selected Areas in Communications (JSAC)*, 2025.
- Ruiqi Shen, Haotian Wu, Wenjing Zhang, Jiangjing Hu, Deniz Gündüz, “Compression Beyond Pixels: Semantic Compression with Multimodal Foundation Models”, *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2025.
- Maojun Zhang, Haotian Wu, Guangxu Zhu, Richeng Jin, Xiaoming Chen, and Deniz Gündüz, “Semantics-Guided Diffusion for Deep Joint Source-Channel Coding in Wireless Image Transmission”, *under review*, 2025.
- Meng Hua, Chenghong Bian, Haotian Wu, Deniz Gündüz, “Realizing Fully-Connected Layers Over the Air via Reconfigurable Intelligent Surfaces”, *under review*, 2025.
- Jiangjing Hu, Haotian Wu, Wenjing Zhang, Fengyu Wang, Wenjun Xu, Hui Gao, Deniz Gündüz, “Zero-Shot Semantic Communication with Multimodal Foundation Models”, *under review*, 2025.

Chapter 2

Background and related works

This chapter will establish the foundational concepts and provide literature reviews on deep joint source and channel coding and semantic communication systems. These two emerging fields offer substantial opportunities and present unique challenges, as they necessitate the integration of cutting-edge machine learning technologies with the proper design of wireless communication systems.

2.1 From Shannon’s separation theorem to Deep JSCC

Shannon’s separation theorem Shannon’s separation theorem is the foundational principle in the design of conventional wireless communication systems. According to this theorem, reliable transmission of a source is achievable in a point-to-point communication system if the minimum source coding rate is below the channel capacity [62]. Conversely, reliable transmission becomes impossible if the minimum source coding rate exceeds the channel capacity. The theorem further asserts that these results can even hold when the source and channel codes are designed independently. These two findings lead to the conclusion that the separation design of source and channel coding is optimal under the assumptions of infinite source and channel bandwidths, ergodic source and channel distributions, and in the context of an additive distortion measure. However, these assumptions are often impractical in real-world

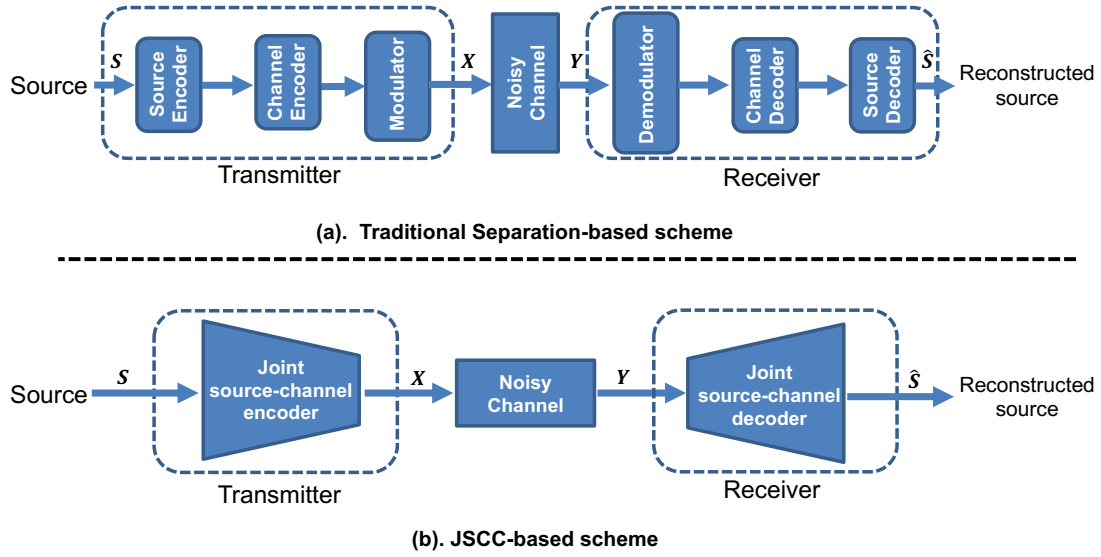


Figure 2.1: A comparison between conventional separation-based scheme and JSCC scheme, where source S is transmitted as a channel input X , and is reconstructed as $\hat{S}$ based on the channel output Y .

scenarios. For instance, wireless communication systems typically operate under limited block length and bandwidth constraints, especially in applications demanding low latency. Despite these practical limitations, Shannon's source-channel separation theorem has still been widely employed as a foundation for designing modern communication systems, commonly referred to as separation-based communication schemes.

Separation-based communication scheme: According to Shannon's separation theorem, a conventional separation-based scheme is structured with distinct source and channel coding modules. As depicted in Fig. 2.1 (a), the transmitter sequentially employs source coding, channel coding, and modulation techniques. Conversely, the receiver performs corresponding demodulation, channel decoding, and source decoding operations in a reverse order to reconstruct the message. Particularly, the source coding process aims to eliminate redundancy within the source data, thereby achieving efficient communication. This allows the source to be transmitted within fewer bits, which can be achieved either in a lossy or lossless manner. For semantic sources such as images and videos, a range of competitive compression codecs is available, including BPG [39], JPEG [63], and advanced neural compression algorithms [5, 64, 65, 66, 67]. After the source coding, a channel coding module is employed to detect and correct bit errors

arising from the noisy transmission. In this stage, a source data sequence of length k -bits is encoded into a longer sequence of channel input with length n bits ($n > k$) by incorporating structured redundant bits to enhance error correction capabilities. Various channel coding schemes, such as LDPC codes [40], Turbo codes [68], and Polar codes [69], have demonstrated exceptional performance with notably low block-error rates (BLER). These techniques are crucial for ensuring the robustness of modern digital communication systems, as the presence of even a single bit error in the bit stream can prevent the successful decoding of the source signal.

Despite its theoretical elegance, the separation-based scheme has three primary drawbacks:

- Firstly, the optimality of separation-based design is often unachievable in practical scenarios constrained by finite bandwidth. In most real-world applications, there is a growing demand for the efficient transmission of large data volumes while maintaining high quality and low latency, challenging separation-based methods significantly.
- Secondly, the separation-based scheme significantly relies on accurate channel estimation. Exceeding the channel capacity during transmission can directly yield bit errors and consequent rate outages, potentially resulting in unsuccessful transmission or decoding.
- Thirdly, the separation-based scheme lacks adaptability to fluctuating channel conditions, necessitating the frequent switching of communication schemes to match varying channel conditions. Otherwise, this can result in a significant degradation in performance, known as the cliff effect, particularly when the signal-to-noise ratio drops below a critical threshold for each specific transmission strategy.

Classical JSCC communication scheme: Given the practical limitations of separation-based schemes, researchers are increasingly turning their attention to joint source and channel coding methodologies. Different from separation-based schemes, which independently perform source coding and channel coding, a classical JSCC scheme [13] directly maps the source signal into the channel symbols at the transmitter, as shown in Fig. 2.1 (b). Consequently, the receiver reconstructs the source signal based on the received channel outputs. One of the key

advantages of JSCC-based systems is that they can avoid the cliff effect, commonly associated with separation-based approaches. Instead, JSCC systems demonstrate graceful degradation, wherein the reconstruction quality gradually diminishes as channel conditions deteriorate, which is particularly beneficial in scenarios with uncertainties in channel estimation. In recent decades, various communication systems have demonstrated substantial benefits through the application of JSCC technologies. These applications encompass video transmission [19, 70], image communication [17, 71], and speech transmission [72]. Additionally, JSCC has proven effective in diverse wireless channel models, such as MIMO [18, 73], OFDM [74, 75], broadcasting channels [76, 77], feedback channels [78], and relay channels [79].

Typical JSCC designs in wireless image transmission schemes are based on the rate-distortion (R-D) trade-off paradigm [80]. This paradigm characterizes the trade-off between the transmission quality and the minimum transmission rate to achieve that specified quality level. Instead of independently selecting source and channel codes based on the target transmission performance, classic JSCC approaches involve a joint optimization of both source and channel coding. This joint optimization process has demonstrated potential for enhancing communication quality, particularly in scenarios involving channel mismatch [13, 17, 18, 19, 70, 71, 73]. The primary challenge here is to attain an optimal balance between the quantization error introduced by source coding and the transmission error arising from the inherent imperfections in the channel coding scheme. Achieving this objective necessitates the development of an optimization procedure that incorporates the characteristics of both the source coder, such as its RD curve, and the channel coder, encompassing its error-correcting capabilities and corresponding code rates. Authors in [81] first introduced a methodology to assess the joint R-D performance and estimated the information-theoretic bounds of JSCC schemes under a fixed bandwidth constraint. Another advanced JSCC scheme designed for time-varying channels is presented in [82], utilizing multi-resolution modulation constellations to enhance adaptability across different scenarios. Building upon these foundations, a progressive image transmission scheme has been proposed in [83], which optimizes intermediate transmission rates to maximize transmission performance under channel noise, thereby enhancing the robustness against potential outages.

Despite their advantages over traditional separation-based schemes in practical finite block-length regimes, classical JSCC schemes still encounter substantial challenges. The primary issue is the high computational complexity involved in optimizing source and channel code configurations. Furthermore, classical JSCC approaches often lack adaptability across varying source and channel distributions, which may limit their practical applicability. Additionally, designing effective classical JSCC schemes is highly challenging, and no such codes have been known to perform on par with state-of-the-art separation-based codes.

Deep JSCC: To address the limitations of classical JSCC methods, researchers have started to exploit DNNs to design novel and competitive JSCC schemes to transmit high information content signals over wireless channels [15, 32, 33, 34, 35, 36, 37, 38, 48, 84]. These DNNs-based JSCC schemes, denoted as DeepJSCC, directly map the input source signal to channel symbols, and vice versa at the decoder. These methods remove 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 to enhance 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 source or channel distribution.

Specifically, DeepJSCC is a data-driven technique that uses DNNs to learn an optimal representation of the source data that is both compact and robust to channel impairments, effectively merging the compression and error-correction functionalities. The process typically involves:

- An encoder network: A DNN that encodes the source data into a compact representation tailored for transmission over a specific communication channel. The encoded representation is designed to be resilient to the impairments of the channel, such as noise, fading, or interference.
- A decoder network: A DNN takes the received (and possibly corrupted) encoded representation and attempts to reconstruct the original source data. The decoder is designed to correct errors and recover the original data as accurately as possible.

- End-to-end optimization: The encoder and decoder are jointly optimized to minimize the end-to-end reconstruction error, taking into account the specific characteristics of the channel. This can lead to more efficient use of bandwidth and better performance under certain conditions compared to traditional methods.

DeepJSCC has been pioneered in [15], where an autoencoder-based JSCC architecture is proposed for wireless image transmission, which outperformed standard compression and channel coding schemes over additive white Gaussian noise (AWGN) and Rayleigh fading channels. Subsequent to this work, channel output feedback was incorporated in [35, 47] by designing multiple coding processes for each feedback iteration, where the model can refine the transmission from the feedback signals. Expanding upon prior DeepJSCC frameworks, [32, 41] extended the methodology to OFDM channels by training the specific AE pair within an OFDM scenario. To achieve the flexible channel bandwidth of the DeepJSCC, [84] proposed a layered transmission scheme where images are progressively transmitted in distinct layers, while [85] employed a separate policy network to control the transmission rate. Above broad applicability is attained through the joint training of the model over relevant channel systems, followed by model-driven optimization techniques customized for specific communication scenarios. Furthermore, [33] employed the attention mechanism to adapt the DeepJSCC model to AWGN channels with varying Signal-to-Noise Ratio (SNR) values, where the intermediate features are scaled by the attention mechanism based on the channel SNR values. An alternative generative architecture is considered in [36, 37, 38] to improve the performance of distortion and perceptual quality. The success of DeepJSCC schemes can be extended to other downstream tasks beyond signal reconstructions, such as image retrieval [57, 86], visual question answering [87], and edge inference [10], where the transmitter only transmits the relevant semantic features instead of all the image pixels, saving much bandwidth. Beyond its efficacy in various tasks, the practicality of DeepJSCC in real-world systems has been evaluated through prototype development [88, 89], implementation with constrained channel inputs [90], and operation under peak-to-average power ratio constraints [42].

In summary, DeepJSCC schemes can be adapted to various types of data (e.g., texts [14, 91],

images [15], videos [48], speech [49], point clouds [92, 93]) and different communication channels (e.g., OFDM [32, 41, 42], MIMO [43, 44], relay [45], multiple-access [46], broadcasting and feedback channels [47]) by training the encoder and decoder on relevant data and channel models, making it an appealing technology for next-generation communication systems [4, 50, 51, 57, 86].

2.2 From technical to semantic communication

To date, digital communication systems have been developed and optimized with the primary objective of reliably transmitting bits over noisy channels. This approach facilitates efficient engineering designs that are independent of the message content or the communication objectives. However, next-generation systems have the potential to enhance their performance by incorporating message semantics and communication goals into their design framework, which paves the way for the development of novel and innovative design strategies in the future.

Shannon’s coding theory [11] establishes the fundamental limits on the rate at which reliable bit sequence can be transmitted. This theory assumes a communication system that functions independently of the messages’ semantics and the communication’s goal, focusing solely on the efficient transmission of information. Although this approach has demonstrated substantial efficacy in traditional systems, the rise of emerging AI applications has significantly boosted the interest and potential in semantic communications. In scenarios such as autonomous driving and the IoT, there is a notable tendency towards integrating computation and communication with physical processes, which allows processing units to collaborate and communicate efficiently, thereby improving task performance. Unlike conventional communication networks, which prioritize the accurate and independent reconstruction of the original message, the objective in these scenarios shifts towards enabling the receiver to make correct decisions aligned with the intended communication objective. Consequently, communication systems can be further optimized by incorporating the message’s semantics and the communication’s specific goals, underscoring the necessity for developing semantic communication systems.

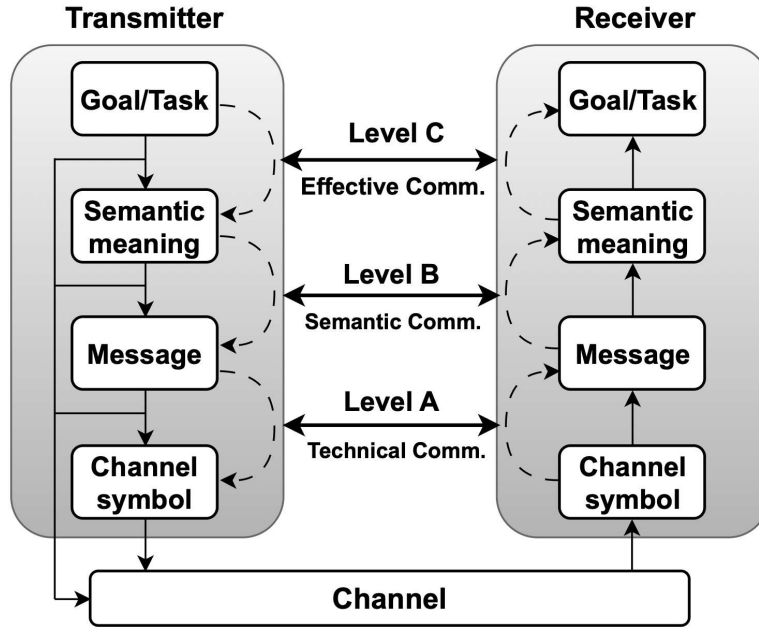


Figure 2.2: Illustrations of Weaver's three levels communication problems.

To provide a comprehensive understanding of semantic communication, we introduce its background through three key aspects: the definition of semantic communication, the motivation behind its development, and the relevant surveys on this topic.

What is semantic communication: Following the Shannon's information theory [11], Weaver in [52] categorized a board communication problem into three distinct levels:

- **Level A (The technical problem):** "How accurately can channel symbols be transmitted?"
- **Level B (The semantic problem):** "How precisely does the communicated message convey the intended semantic meaning?"
- **Level C (The effective problem):** "How effectively does the received meaning influence the communication goal in the desired manner?"

In the context of the three-level communication problem, as depicted in Fig. 2.2, effective communication at level C emphasizes generating the intended meaning to achieve the communication goal, taking into account the beliefs of both the transmitter and the receiver, as well

as ongoing task progress. In contrast, semantic communication focuses on constructing messages to accurately convey semantic meaning based on a shared language or protocol. Lastly, technical communication addresses the design of channel symbols to ensure that messages are transmitted with maximum precision. According to Shannon’s coding theory, the amount of information in a message is quantified through the concept of ‘uncertainty,’ emphasizing the probabilities associated with the message rather than its semantic meaning. Conversely, semantic communication focuses on the conveyed meaning of a message, considering the communication’s objective while minimizing the associated communication cost. Note that mapping between semantic meanings and messages is often a many-to-many correspondence in a communication task. This implies that distinct semantic meanings can produce identical messages; conversely, the same semantic meaning can be conveyed through multiple messages. As a result, these many-to-many relationships can lead to two typical scenarios: successful semantic communication with failed or imperfect technical communication and, conversely, failed semantic communication with successful technical communication. Of these two scenarios, the former is an acceptable outcome that we can optimize for, while the latter is undesirable and must be avoided.

Although Weaver explicitly excluded semantics from Shannon’s theory, he still argued that Shannon’s theory has implications for Level B and Level C problems, as discussed in [94]. Meanwhile, many researchers at the time were excited about applying Shannon’s statistical framework to interpreting and measuring semantics. Wiener, in [95], wrote that “The amount of meaning can be measured. It turns out that the less probable a message is, the more meaning it carries, which is entirely reasonable from the standpoint of common sense.” However, many scholars also argue that the classical Shannon theory cannot address the semantic aspects of communication systems, where Shannon’s channel coding theorem emphasizes reliable bit transmission, and his source coding theory focuses on rate-distortion without considering the semantic content [96]. These different views have, in turn, also spurred advancements in the exploration of semantic communication.

Motivation of semantic communication More recently, there has been a growing interest in semantic communication systems, primarily driven by the rise of AI applications that are anticipated to dominate the future communication traffic. Existing communication systems are meticulously designed and optimized to ensure the reliable and efficient delivery of data packets, which are agnostic to the packet's content or its impact on the receiver. Recent advances have sparked increasing interest in incorporating content semantics and its relevance to communication objectives into system design, with the aim of enhancing overall communication or task performance. The underlying reasons for this trend can be summarized as follows:

- The increasing traffic is going to result in network capacity saturation.

With the rapid advancement of AI, applications such as autonomous driving and multi-modal large language models are increasingly dominating the communication traffic. These applications typically depend on extensive data transfers for training large-scale models or high-volume data exchanges for remote inference, potentially leading to network capacity saturation. In response, the communication system should not only function as a passive pipe for bits transmission but also as a semantic-aware communication framework with its own intelligence. This shift is crucial, as it ensures that only the critical traffic pertinent to the task is transmitted at the optimal time and speed.

- The demand for low-latency and semantically accurate transmission is growing rapidly across various applications.

For instance, in video transmission, the rise of applications like live streaming, autonomous driving, and real-time factory monitoring is driving the need for low latency, particularly in bandwidth-constrained environments. However, the traditional approach of maximizing video fidelity with the least number of bits may not always be necessary for every user or scenario. This is where the incorporation of intelligence into the communication system becomes crucial. It facilitates the efficient transmission of task-relevant or crucial semantic information with minimal delay and high reliability, thereby paving the way for advancements in low-latency communication.

- DL-based end-to-end communication paradigms exhibit significant robustness to imperfect technical transmissions

Recent advancements in deep learning have significantly enhanced the exploration and development of deep learning-based end-to-end communication systems [50], including areas such as deep joint source-channel coding [35] and goal-oriented communication frameworks [10]. A key characteristic of these systems is their capability for end-to-end optimization tailored to specific communication objectives. Advanced deep learning techniques, such as auto-encoders [85], transformers [47], and diffusion models [97], have shown exceptional proficiency in generating channel symbols by learning semantic representations and mitigating the impact of noise in communication channels. Additionally, even in the presence of imperfect technical transmissions, where reconstructed messages are significantly different from the original ones, these end-to-end approaches can sustain acceptable levels of performance for specific metrics relevant to downstream tasks, such as perceptual quality [47] and inference accuracy [51], by efficiently transmitting critical features under limited resource constraints.

Relevant surveys With the rising interest in semantic communications, there has been a noticeable increase in the number of surveys and tutorials available on this subject.

Authors in [50] provide basic information theoretic concepts for semantic communication from the viewpoint of JSCC, where several practical frameworks for goal-oriented semantic communication over noisy channels are provided for text, speech, and image sources. In [98], semantic communication is formulated into two fundamental problems: language exploitation and design. The first focuses on minimizing misinterpretations between the transmitter and receiver when both have agreed on languages. The second explores how to design these languages through negotiation to convey semantic meaning efficiently. In [99], semantic communications are categorized into three types: human-to-human, human-to-machine, and machine-to-machine. The key performance indicators and system design considerations for semantic learning in future wireless communication networks are elaborated in [100]. The work presented in [101] focuses on goal-oriented signal processing, integrating graph-based semantic language and the represen-

tation of semantic information into the semantic signal processing. Different architectures for semantic communication in AI-empowered wireless networks are examined in [102] and [103]. The technical aspects and potential applications for intelligent and efficient semantic communication network design are explored in [55], while [104] delves into the principles and challenges of deep learning-enabled semantic communications.

Chapter 3

DeepJSCC for adaptive image transmission over OFDM channels

3.1 Introduction

Recently, researchers have started to exploit deep neural networks to design novel and competitive JSCC schemes to transmit high information content signals, such as images or videos, over wireless channels [15, 32, 33, 35, 36, 37, 38, 48, 84]. However, adaptability to various channel conditions is still a challenge for deep-learning-based JSCC. Previous methods are either trained for a specific SNR, or for a range of channel SNRs. The former requires significant storage memory to store different network parameters for different channel conditions, while the latter sacrifices performance and does not exploit the CSI.

In conventional digital communication systems, CSI at the transmitter can allow power allocation to boost the communication rate. In [32, 33, 48], CSI is used in a similar manner in the context of learning-aided design, mainly to adjust the feature weights according to the CSI; however, in the case of OFDM, CSI will be instrumental not only for power control, but also to decide the mapping of the features to different subcarriers according to their relative qualities. For example, more critical features of the input image can be mapped to more reliable subcarriers.

In this chapter, we introduce a channel-adaptive JSCC (CA-JSCC) scheme over the OFDM channels, which employs a dual-attention mechanism to adjust its features in the multi-scale intermediate layers according to the estimated CSI at the encoder and decoder. Specifically, our dual-attention mechanism employs both *channel-wise* attention and *spatial* attention, and jointly learns to map the features to the subcarriers and to allocate power judiciously. Our method achieves state-of-the-art performance and can adapt to different channel conditions. Our main contributions in this chapter can be summarized as:

- To the best of our knowledge, channel adaptability for JSCC with OFDM has not been studied in prior research. Existing methods require the training and testing SNRs to match without fully exploiting the CSI. Instead, we present a channel-adaptive JSCC scheme, which incorporates a dual-attention mechanism to dynamically adjust its features in the multi-scale intermediate layers based on the estimated CSI at both the encoder and decoder for the OFDM transmission. Our dual-attention mechanism employs both *channel-wise* attention and *spatial* attention, which collaboratively learns to map the features to the subcarriers and to allocate power judiciously.
- This chapter conducts a comprehensive validation of the proposed CA-JSCC scheme through a series of experiments. Specifically, the investigations encompass evaluations of general performance, channel-gain adaptability, SNR adaptability, and the impact of noisy CSI. The proposed CA-JSCC scheme achieves the state-of-the-art performance in various SNR and bandwidth scenarios. The proposed dual-attention mechanism can enable the CA-JSCC to simultaneously exploit the estimated CSI to aid the allocation of features and power resources to adapt to time-varying channel conditions.

3.2 System Model

We consider a general image transmission problem over a point-to-point wireless communication channel, as shown in Fig. 3.1. The goal is to transmit an input image $\mathbf{S} \in \mathbb{R}^{h \times w \times 3}$ over a wireless

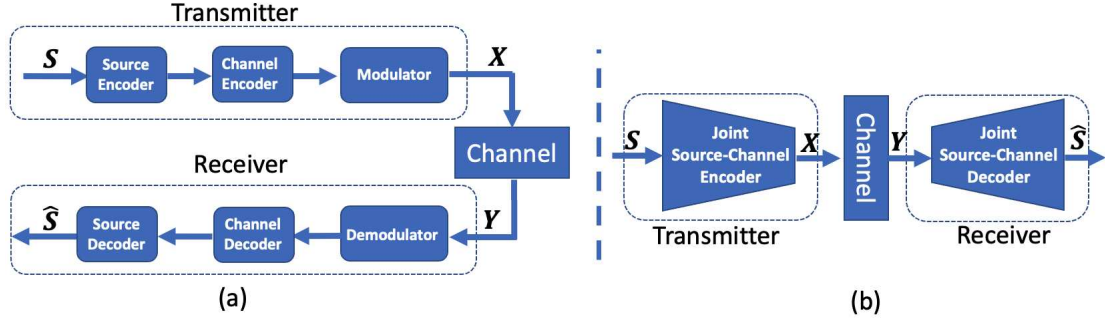


Figure 3.1: Block diagram of the wireless image transmission system: (a) the conventional separation-based scheme and (b) the deep learning based JSCC scheme.

communication channel, where h, w and 3 denote the height, width and color channels of the image. The transmitter maps the source image $\mathbf{S}$ into a channel input matrix $\mathbf{X}$. Due to the practical considerations, such as limited energy resources, $\mathbf{X}$ is subject to a power constraint P_s as: $\|\mathbf{X}\|_F^2 \leq P_s$, where $\|\cdot\|_F$ is the Frobenius norm operation. Subsequently, $\mathbf{X}$ is conveyed through noisy channels, yielding the channel output $\mathbf{Y}$. This output is subsequently input into the receiver to reconstruct the source image as $\hat{\mathbf{S}} \in \mathbb{R}^{h \times w \times 3}$. For the traditional separate transmission scheme, the transmitter sequentially performs image compression, channel coding, and modulation to generate the channel input matrix $\mathbf{X}$. The receiver performs demodulation, channel decoding, and source decoding sequentially to reconstruct the source image as $\hat{\mathbf{S}}$.

Specifically, this chapter considers an OFDM-based JSCC scheme for image transmission over a multipath fading channel with L_t paths. We transmit each input image using N_s OFDM symbols accompanied with N_p pilots for channel estimation of L_f OFDM subcarriers. As shown in Fig 3.2, an encoding function E_θ first maps the input image $\mathbf{S}$ into a complex matrix $\mathbf{X} \in \mathbb{C}^{N_s \times L_f}$. The generated channel input can be denoted by $\mathbf{X} = E_\theta(\mathbf{S}, \hat{\mathbf{h}})$, where $\hat{\mathbf{h}}$ is the estimated CSI vector available at the transmitter. Channel input $\mathbf{X}$ is subject to an average power constraint P_s : $\frac{1}{N_s L_f} \|\mathbf{X}\|_F^2 \leq P_s$. Without loss of generality, we set $P_s = 1$.

Each OFDM symbol is passed through the inverse discrete Fourier transform (IFFT) module, appended with the cyclic prefix (CP), and transmitted to the receiver over the multipath channel. The transfer function of the multipath fading channel with L_t paths is defined as: $\mathbf{y} = h_c(\mathbf{x}) = \mathbf{h}_t * \mathbf{x} + \mathbf{w}$, where $\mathbf{x}$ and $\mathbf{y}$ denote the input and output vectors, respectively; $*$ is the linear convolution operation, $\mathbf{h}_t \in \mathbb{C}^L$ is the channel impulse response, and $\mathbf{w}$ is the

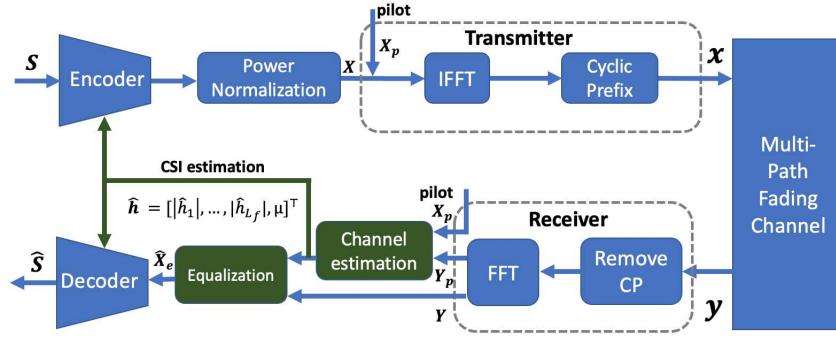


Figure 3.2: Illustration of the proposed CA-JSCC scheme, where the encoder maps the input image $\mathbf{S}$ and estimated CSI $\hat{\mathbf{h}}$ to the channel input matrix $\mathbf{X}$, and the decoder reconstructs the source as $\hat{\mathbf{S}}$.

AWGN term. The receiver first demodulates $\mathbf{y}$ by removing the CP and applying a fast Fourier transform (FFT). The equivalent frequency-domain transfer function of $h_c(\cdot)$ can be written as:

$$Y[i, k] = H[k]X[i, k] + W[i, k], \quad (3.1)$$

where the frequency-domain channel matrix $\mathbf{H} \in \mathbb{C}^{L_f \times L_f}$ is a diagonal matrix with the k -th diagonal element being $H[k]$ (frequency-domain channel response of the k -th subcarrier). Let $\mathbf{X} \in \mathbb{C}^{N_s \times L_f}$ denote the output of the encoder (i.e., the inputs to the IFFT module), where $X[i, k]$ denotes the symbol at the k -th subcarrier of the i -th OFDM symbol. $\mathbf{W} \in \mathbb{C}^{N_s \times L_f}$ is the frequency domain noise matrix, where $W[i, k] \sim \mathcal{CN}(0, \sigma^2)$ and independent of each other.

Given the FFT output of the pilots at the receiver, we use minimum mean square error (MMSE) or least square (LS) channel estimator to estimate the CSI ($H[k]$) in frequency-domain (Eqn. (3.1)). The estimated CSI vector $\hat{\mathbf{h}}$ is then used to equalize the data (MMSE equalizer). We have $\hat{\mathbf{h}} \triangleq \left[|\hat{h}_1|, |\hat{h}_2|, \dots, |\hat{h}_{L_f}|, \mu \right]^T$, where $\hat{h}_i$, $i = 1, \dots, L_f$, is the estimated channel gain of the i -th subcarrier. μ is the average SNR defined as:

$$\mu = 10 \log_{10} \frac{P_s}{\sigma^2} \text{ dB}, \quad (3.2)$$

where P_s is the transmit power and σ^2 is the noise power.

Based on the equalized data $\hat{\mathbf{X}}_e$ and the estimated CSI vector $\hat{\mathbf{h}}$, the decoder D_ϕ reconstructs the transmitted image as $\hat{\mathbf{S}}$, i.e., $\hat{\mathbf{S}} = D_\phi(\hat{\mathbf{X}}_e, \hat{\mathbf{h}})$. The performance indicator is the peak signal-

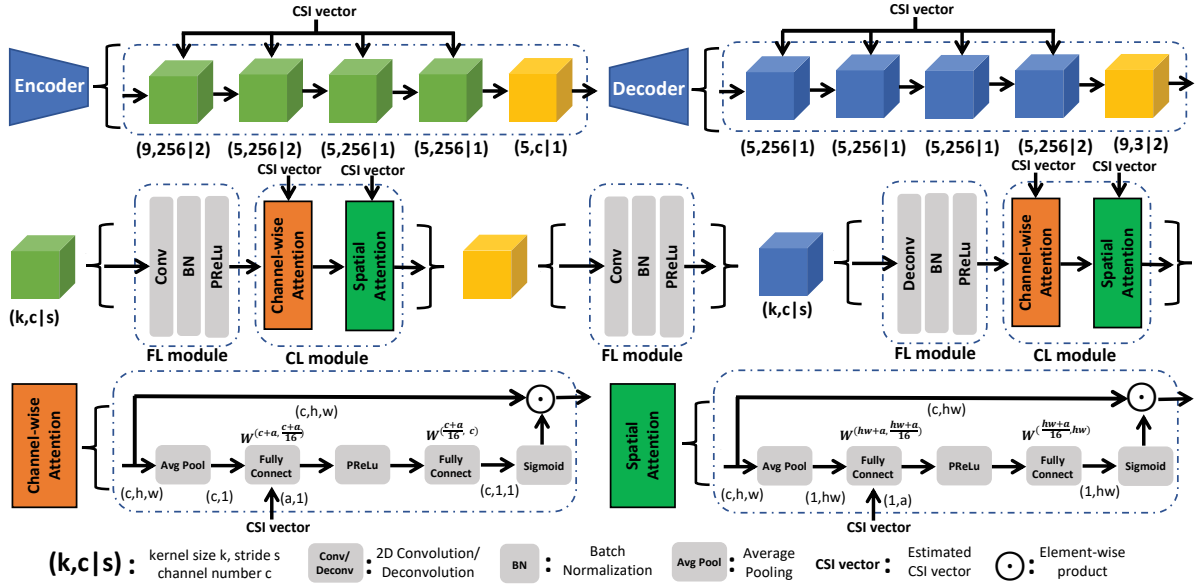


Figure 3.3: Basic blocks of our dual-attention encoder and decoder architectures, where both the encoder and decoder are composed of spatial attention blocks and channel attention blocks arranged in a comb structure.

to-noise ratio (PSNR), defined as:

$$\text{PSNR} = 10 \log_{10} \frac{(\max \mathbf{S})^2}{\text{MSE}(\mathbf{S}, \hat{\mathbf{S}})} \quad (\text{dB}), \quad (3.3)$$

where $\max \mathbf{S}$ denotes the maximum possible value of the input signal $\mathbf{S}$, $\text{MSE}(\mathbf{S}, \hat{\mathbf{S}}) \triangleq E[\|\mathbf{S} - \hat{\mathbf{S}}\|_2^2]$ and the expectation is taken over all pixels.

We then jointly train the encoder and decoder to minimize the loss function: $\mathcal{L}(\theta, \phi) = \mathbb{E}[\text{PSNR}(\mathbf{S}, \hat{\mathbf{S}})]$, where the expectation is taken over the randomness both in the source and channel distributions.

3.3 Proposed method

As shown above, the distinct subcarriers within the OFDM system experience different channel gains. A judicious transmission scheme should be able to intelligently allocate power and features across appropriate subcarriers, thereby adapting to channel variations. To address this, we propose a novel dual-attention based CA-JSCC scheme to fully exploit the estimated

Algorithm 1 Dual-attention mechanism**Input:** $\mathbf{F}_{in} \in \mathbb{R}^{c \times h \times w}$, $\hat{\mathbf{h}} \in \mathbb{R}^{L_f+1}$ **Output:** $\mathbf{F}_{out} \in \mathbb{R}^{c \times h \times w}$ **Stage 1: Channel-wise attention**

- 1: $\mathbf{F}_{ca} = \text{Ave}_c(\mathbf{F}_{in}) \in \mathbb{R}^{(c)}$
- 2: $\mathbf{i}_c = \text{concatenate}(\mathbf{F}_{ca}, \hat{\mathbf{h}}) \in \mathbb{R}^{(c+L_f+1)}$
- 3: $\mathbf{S}_c = f_c(\mathbf{i}_c) \in \mathbb{R}^{(c,1,1)}$
- 4: **for** $i = 0:1:c$ **do**
- 5: $\mathbf{F}_{cout}[i, h, w] = \mathbf{S}_c[i] \odot \mathbf{F}_{in}[i, h, w] \in \mathbb{R}^{(c,h,w)}$
- 6: **end for**

Stage 2: Spatial attention

- 1: $\mathbf{F}_{sa} = \text{Ave}_s(\mathbf{F}_{cout}) \in \mathbb{R}^{(1,h,w)} \Rightarrow \mathbb{R}^{(hw)}$
- 2: $\mathbf{i}_s = \text{concatenate}(\mathbf{F}_{sa}, \hat{\mathbf{h}}) \in \mathbb{R}^{(hw+L_f+1)}$
- 3: $\mathbf{S}_s = f_s(\mathbf{i}_s) \in \mathbb{R}^{(hw)} \Rightarrow \mathbb{R}^{(h,w)}$
- 4: **for** $j = 0:1:h$ **do**
- 5: **for** $k = 0:1:w$ **do**
- 6: $\mathbf{F}_{out}[c, j, k] = \mathbf{S}_s[j, k] \odot \mathbf{F}_{cout}[c, j, k] \in \mathbb{R}^{(c,h,w)}$
- 7: **end for**
- 8: **end for**

CSI. The architecture of our method is shown in Fig. 3.3, both the encoder and the decoder have five feature learning (FL) and four channel learning (CL)¹ modules in a comb structure, allowing modulating each feature with a different scale. The FL module consisting of 2D convolution/deconvolution, batch normalization, and parametric rectified linear unit (PReLU) layers, is designed to learn an intermediate representation of the input. The dual-attention based CL module is designed to learn an attention mask to map the intermediate features to appropriate subcarriers based on the estimated CSI and input features. The CL module consists of a *channel-wise* attention module and a *spatial* attention module. Its operation is presented in Algorithm 1 in detail.

3.3.1 Channel-wise attention module

Our *channel-wise* attention module is inspired by [33], which adapts to a single SNR value in an AWGN channel model. Instead, CA-JSCC learns an attention mask to allocate and map features based on the estimated CSI of all N_s subcarriers.

¹In this chapter, CL represents the channel learning modules, different from CL in Chapter 6 that denotes contrastive learning.

We first apply average pooling $Ave_c(\mathbf{F}_{in})$ on input features $\mathbf{F}_{in}$ along the spatial direction to get vector $\mathbf{F}_{ca}$, where $Ave_c(\mathbf{F}_{in}) \triangleq \frac{1}{hw} \sum_{j=1}^h \sum_{k=1}^w \mathbf{F}_{in}[c, h, w]$. $\mathbf{F}_{ca}$ is then concatenated with $\hat{\mathbf{h}}$ to get the intermediate vector $\mathbf{i}_c$ to compute the *channel-wise* attention mask $\mathbf{S}_c$ by several fully connected (FC) layers: $\mathbf{S}_c = f_c(\mathbf{i}_c)$, where f_c represents the FC layers followed by PReLU functions. Finally, we get the output of our *channel-wise* attention module as $\mathbf{F}_{cout} = \mathbf{S}_c \odot \mathbf{F}_{in}$.

Our *channel-wise* attention module learns to map features from the input to the subcarriers based on the estimated CSI, allowing JSCC to dynamically adjust to different channel SNRs. But the spatial information is ignored when computing the channel attention mask by the average pooling operation. So we design a *spatial* attention module to compensate for spatial information.

3.3.2 Spatial attention module

Our *spatial* attention module learns to match the more critical spatial features along the h and w dimensions with better channel conditions depending on the estimated CSI.

This time we firstly apply average pooling $Ave_s(\mathbf{F}_{cout})$ on the $\mathbf{F}_{cout}$ along the channel direction to get $\mathbf{F}_{sa}$, where $Ave_s(\mathbf{F}_{cout}) \triangleq \frac{1}{c} \sum_{j=1}^c \mathbf{F}_{cout}[c, h, w]$. Then, $\hat{\mathbf{h}}$ and $\mathbf{F}_{sa}$ are concatenated to get the intermediate feature $\mathbf{i}_s$, which is used to compute the *spatial* mask $\mathbf{S}_s$ by several FC layers. We compute the final output feature vector as $\mathbf{F}_{out} = \mathbf{S}_s \odot \mathbf{F}_{cout}$.

The *spatial* attention module can further improve the PSNR performance by exploiting the spatial information and helping the JSCC encoder to do more adaptive power allocation, which matches critical features with better channels.

3.3.3 Design interpretation

While it is difficult to pinpoint exactly the functionality of CL and FL modules in a complex deep neural network architecture, these arguments follow the original intuitions that have motivated us to introduce these models, and the results of the ablation studies that verify their intended

functionalities. Let us explain our design inspirations, and add some ablation experiments later to verify that the *channel-wise* attention and *spatial* attention models do work as expected.

To better illustrate the CL module design, we here draw the visualization of the FL and CL modules in Fig. 3.4 and Fig. 3.5 to show how our FL and CL modules work.

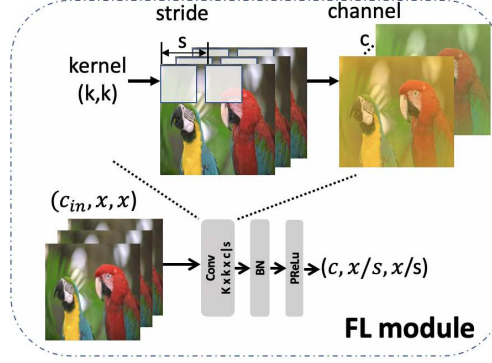


Figure 3.4: Visualization of FL process.

In the FL module, a convolutional or transposed convolutional layer is used to extract features. To visualize the FL process in Fig. 3.4, c kernels each of size (k, k) are used for the convolutional computation with a stride step s over a (c_{in}, x, x) feature map ($c_{in} = 3$), where output will be padded to the size $(c, x/s, x/s)$.

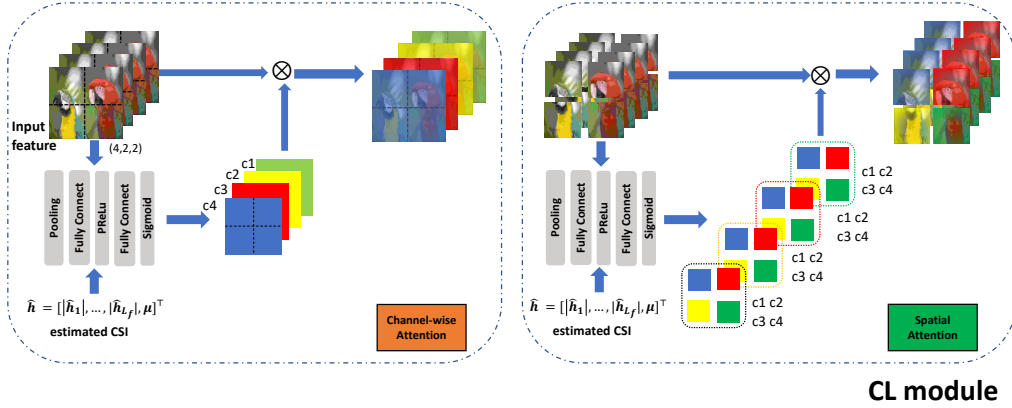


Figure 3.5: Visualization of CL process.

In the CL module, each attention mask is generated by the average pooling operation and two fully-connected layers, followed by batch normalization and the PReLU layer, where details can be seen from Algorithm 1 in the paper. To visualize the attention process, we take the feature map with size $(4, 2, 2)$ as an example input and use different colors to represent different

scale operations in the attention map over different feature parts, as shown in Fig. 3.5. Our *channel-wise* attention generates an attention mask with four scale numbers $c1, c2, c3, c4$ (with different colors) to scale the feature map along different feature channels separately. Similarly, our *spatial* attention mechanism generates a $(2, 2)$ attention mask $\begin{bmatrix} c1 & c2 \\ c3 & c4 \end{bmatrix}$ for features with different pixel positions.

3.4 Training and evaluation of the CA-JSCC

This section presents numerical experiment results to evaluate the performance of our CA-JSCC scheme. The Exp-JSCC scheme introduced in [32] is the most related work to the current study in the literature. We use Exp-JSCC trained on different channel conditions as a benchmark to compare with the proposed CA-JSCC scheme. If not specified otherwise, all experiments were performed on the CIFAR-10 dataset [105] with PyTorch. Models were trained until the performance on a validation set (selected separately from the training dataset) stops improving. The Adam optimizer is used to perform backpropagation.

For a fair comparison with baselines [32, 106], we set the number of subcarriers to $L_f = 64$. The Zadoff-Chu (ZC) sequence [107], denoted by $\mathbf{X}_p \in \mathbb{C}^{(2,64)}$, is used as the pilot. While the reduced number of pilot symbols may degrade channel estimation accuracy, DeepJSCC methods can still achieve competitive performance, as also observed in [106]. Furthermore, under a fixed communication budget, allocating fewer resources to pilot transmission allows for more channel symbols to be used for data, which can enhance overall communication performance, similar to [32]. This trade-off motivates our choice of using $\mathbf{X}_p \in \mathbb{C}^{2 \times 64}$ in our setup. We consider a slow-fading scenario in which the values of channel gains $\{H[k] : k = 1, 2, \dots, L_f\}$ are sampled from a complex Gaussian distribution $\mathcal{CN}(0, 1)$, remain constant during the transmission of each image, and take independent realizations across different transmission blocks. Unless specified otherwise, the frequency-domain channel responses in the experiments are estimated by an MMSE estimator. We also sort the channels based on their estimated CSI to make the training process easier; that is, we have $|H[1]|^2 \geq \dots \geq |H[L]|^2$. Following [35], we define the

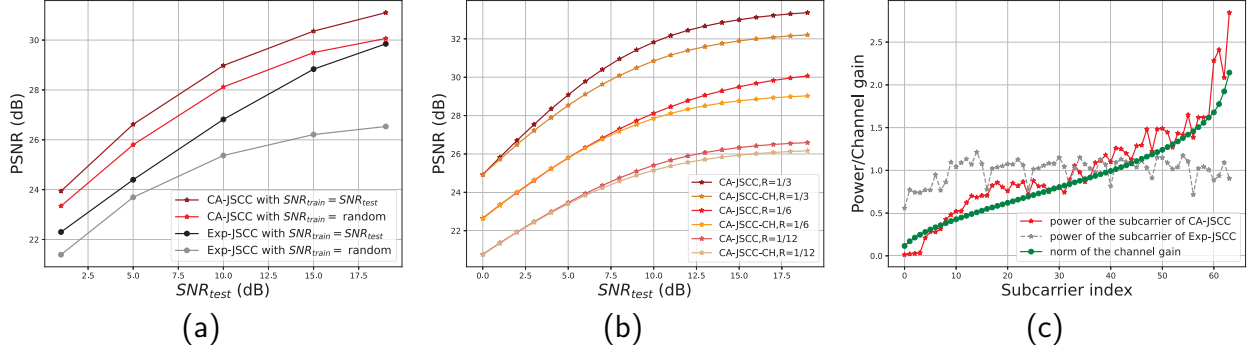


Figure 3.6: (a) Comparison of CA-JSCC scheme with Exp-JSCC when the values of the train and test SNRs are the same. (b) Ablation experiments for the attention strategy. (c) Visualization of the power allocation for the CA-JSCC and the Exp-JSCC.

bandwidth ratio (i.e., bandwidth usage to source symbol ratio) as $R \triangleq \frac{N_s L_f}{c \times h \times w}$, where $N_s L_f$ is the number of symbols transmitted per image.

3.4.1 Channel-gain adaptability

We first verify the adaptability of the CA-JSCC scheme to channel-gain variations. Specifically, under a fixed bandwidth ratio and a given SNR, we want to see if our dual-attention mechanism can instruct the transmitter to exploit better channels and allocate power to different subcarriers judiciously.

The experimental results are shown in Fig 3.6a, where we set the number of OFDM symbols to $N_s = 8$ and the bandwidth ratio to $R = 1/6$. The Exp-JSCC and CA-JSCC models are both trained at a fixed SNR, and tested at the same SNR. As can be seen, by feeding the estimated CSI to the transmitter, CA-JSCC can exploit the channel-gain information and adaptively allocate power to different subcarriers. We can see a significant gain compared to Exp-JSCC at all SNRs. This can be attributed to the advantage of our method in better exploiting the channel gains and allocating power to different subcarriers.

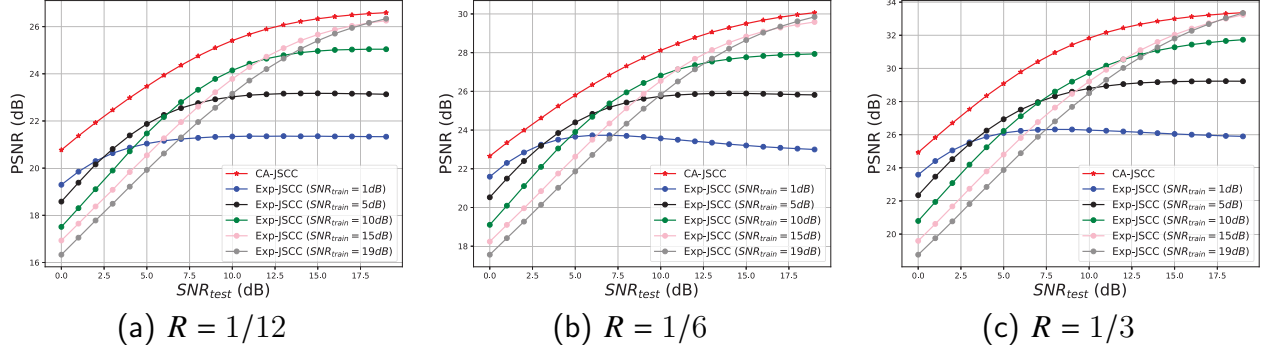


Figure 3.7: Performance of CA-JSCC model compared with the Exp-JSCC model of different bandwidth ratios.

3.4.2 SNR adaptability

Next, we evaluate the SNR adaptability of our scheme. If not specified otherwise, we train the CA-JSCC model at random SNR values of each training episode chosen uniformly from $[0, 20]$ dB and test the well-trained model at different SNRs.

Compared with the CA-JSCC scheme trained at specific SNRs in Fig 3.6a, we observe that there is a slight performance degradation when it is trained at random SNR values. We conclude that, while CA-JSCC can learn to adapt to different channel SNRs, this flexibility comes at the expense of some loss in the PSNR (up to 1dB). However, this CA-JSCC model trained at random SNR values still outperforms the Exp-JSCC models trained at specific SNR values.

We also compare the performance of dual-attention based architecture CA-JSCC with an alternative using only channel-wise attention, called (CA-JSCC-CH), as an ablation study. As shown in Fig 3.6b, for three different bandwidth ratios, CA-JSCC architecture outperforms CA-JSCC-CH at all SNR values, which shows that the *spatial* attention mechanism is essential to achieve the improved performance provided by CA-JSCC. We also observe larger gains by our dual-attention method at higher bandwidth ratios and SNR_{test} values, where more spatial information and better CSI adaptability benefit both feature mapping and power allocation. To visualize the power allocation executed by CA-JSCC and Exp-JSCC, we plot the average channel gain and the average power allocated to each subcarrier in Fig. 3.6c, where we set $\text{SNR} = 1\text{dB}$ and $R = 1/6$. The channel gains are ordered in an increasing manner in the plot. Compared with Exp-JSCC, Fig. 3.6c shows that CA-JSCC generally allocates more power for

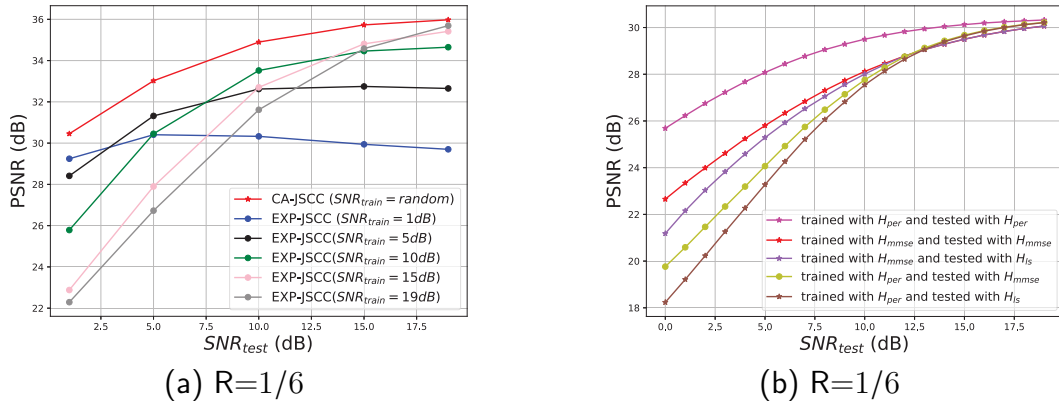


Figure 3.8: (a) Additional experiments with training on the ImageNet dataset tested on the Kodak dataset. (b) Comparison of CA-JSCC with different channel estimation methods.

the subcarrier with better channel conditions, as one would desire.

It is worth noting that the Exp-JSCC scheme is not SNR-adaptive, which means the training and test SNRs of Exp-JSCC must match to achieve a sound performance, as shown in Fig. 3.6a. Fig 3.7 presents the PSNR versus test SNR results for bandwidth ratios of $R = 1/12, 1/6, 1/3$ (we set $L_f = 64$ and vary N_s to attain different R values). As stated above, our CA-JSCC scheme can be trained with random SNRs, and yields a single model for each bandwidth ratio to be verified on a range of test SNRs. The Exp-JSCC scheme, on the other hand, is trained at five different SNRs, yielding five different models under each bandwidth ratio. The Exp-JSCC scheme performs the best when the training and test SNRs match. However, our CA-JSCC scheme is SNR-adaptive and consistently outperforms Exp-JSCC at all SNRs and bandwidth ratios with a considerable margin.

Additional experiments by training over the ImageNet dataset are shown in Fig. 3.8a. We train the models with randomly cropped 64×64 patches from ImageNet, and evaluate the models on the Kodak dataset. Results show that training on a sufficiently large dataset (ImageNet) can allow our CA-JSCC model to perform well on a never-seen dataset (Kodak). CA-JSCC can still achieve state-of-the-art performance with the additional capability of channel adaptability.

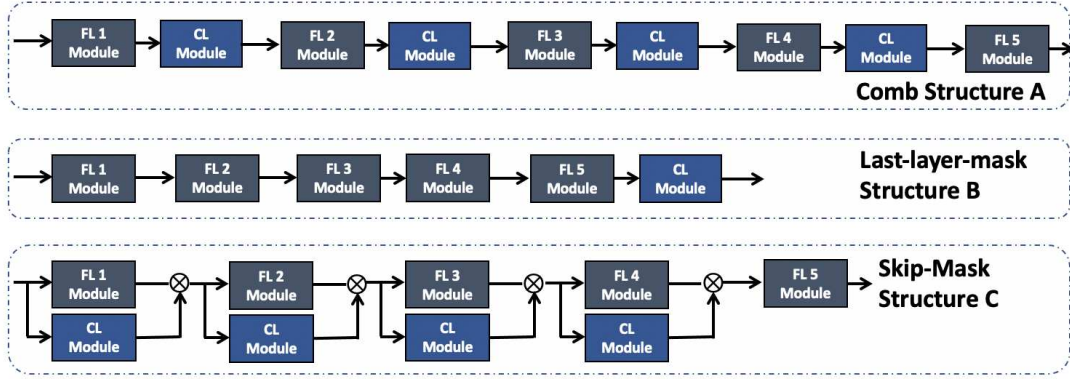


Figure 3.9: Different structures of CA-JSCC.

3.4.3 Impact of CSI estimation errors

In the above experiments, we have assumed MMSE estimated CSI. In this subsection, we look into the effect of channel estimation errors on the performance of CA-JSCC. We repeat the experiment in Fig 3.7b with three types of CSI: i) perfect CSI, H_{per} ; ii) MMSE estimated CSI, H_{mmse} ; and iii) LS estimated CSI, H_{ls} . We remark that H_{mmse} provides a more accurate estimate than H_{ls} .

The experimental results are shown in Fig 3.8b, where we train CA-JSCC models with H_{mmse} and H_{per} , respectively, and evaluate these two models with H_{per} , H_{mmse} and H_{ls} , respectively. As expected, the model trained and tested with H_{per} achieves the best performance. On the other hand, the model trained with perfect CSI is not robust to CSI errors during test time. Its performance gets worse as the quality of channel estimation degrades. Instead, we see that models trained with H_{mmse} perform better, since during training they learn to compensate for CSI estimation errors. We conclude from these results that a more accurate CSI during testing is generally beneficial, and the performance improves if training is done with the same type of CSI.

3.4.4 Ablation study

Ablation study over structures: There are three main reasons for the particular comb structure we used for the CL and FL modules.

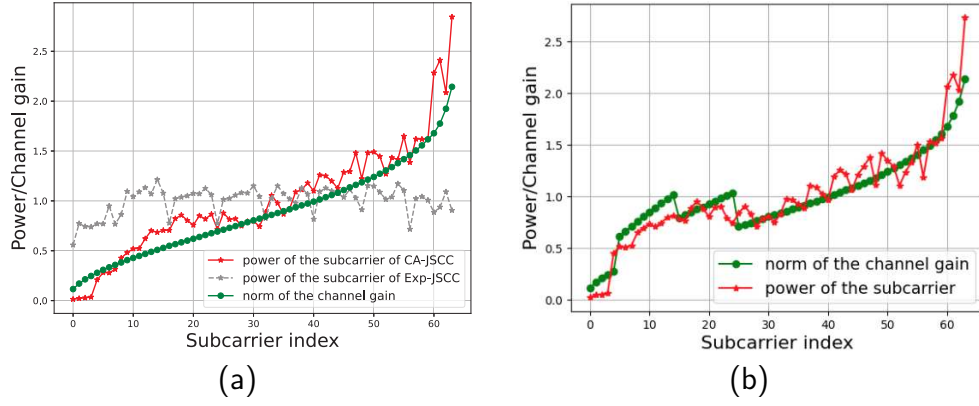


Figure 3.10: Power allocation for CA-JSCC with different channel gain distribution.

The comb structure is widely used and effective for the attention mechanism design in the literature. Previous papers with the attention mechanism achieved remarkable success in various vision tasks by arranging feature learning and attention modules in a comb fashion [33, 108, 109].

Intuitively, we expect the attention based on the estimated CSI to be different in every feature learning step, resulting in different scales, similarly to [110, 111, 112]. That is, the comb structure allows modulating each feature with a different scale, resulting in a more powerful and richer architecture.

The comb structure A (CA-JSCC) achieves the best performance among the three alternatives we tried, illustrated in Fig. 3.9, where each FL and CL module have the same parameters as the Fig. 3.3. Specifically, the other two structures are denoted as Structure B and Structure C, where Structure B is to place the CL module in the last layer, and Structure C is to use a skip style to do feature learning and attention learning at the same time. The performances of the three structures are shown in Fig. 3.11, wherein the comb structure achieves the best performance by a large margin. In addition to the experimental comparison, which demonstrates that Structure A achieves the best performance, we also provide an intuitive analysis to support this result. Specifically, from a design perspective, the attention mechanism scales and masks the features to enable adaptive power allocation. The comb architecture of Structure A allows the model to learn this masking operation from current feature maps and channel state information. Furthermore, Structure A incorporates a multi-scale attention mechanism, which enhances representation capacity and leads to improved transmission performance.

Detailed ablation study of the CL modules: In intuition, we introduce CL modules to exploit the estimated CSI simultaneously to help with the source-to-channel mapping and power allocation. We expect the *channel-wise* attention mechanism to learn a better JSCC mapping with SNR adaptability and expect the *spatial* attention mechanism to learn a flexible power allocation scheme based on the critical features and channel conditions.

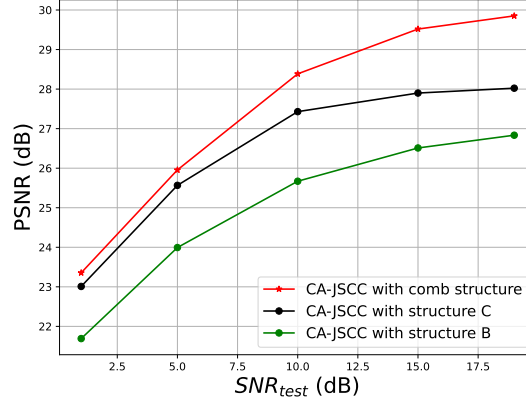


Figure 3.11: Different structures trained with the random SNRs.

Previous JSCC systems could not exploit the CSI for channel adaption or feature mapping. Without the proposed attention mechanism, our method would also achieve a similar performance to the previous state-of-the-art method Exp-JSCC [32] over OFDM channels, which means that the channel adaptability and the overall performance improvement can be attributed to the attention mechanism, instead of a different network structure.

To show the power allocation optimization, we plot the average channel gain and the average power for each subcarrier from different models in Fig. 3.10 and Fig. 3.12 with $R = 1/6$. The average channel gain and average allocated power for each subcarrier is the expectation taken over the randomness both in the source and channel distributions (average SNR= 1dB). Fig. 3.10 (a) compares the power location performance between Exp-JSCC and CA-JSCC, which shows that CA-JSCC can generally allocate more power for the subcarrier with better channel conditions. If some sudden changes exist in the channel gain, where we deliberately change the channel gain in Fig. 3.10 (b), CA-JSCC can still simultaneously allocate power for this condition.

However, for the CA-JSCC without a *spatial* attention mechanism, denoted as CA-JSCC-CH,

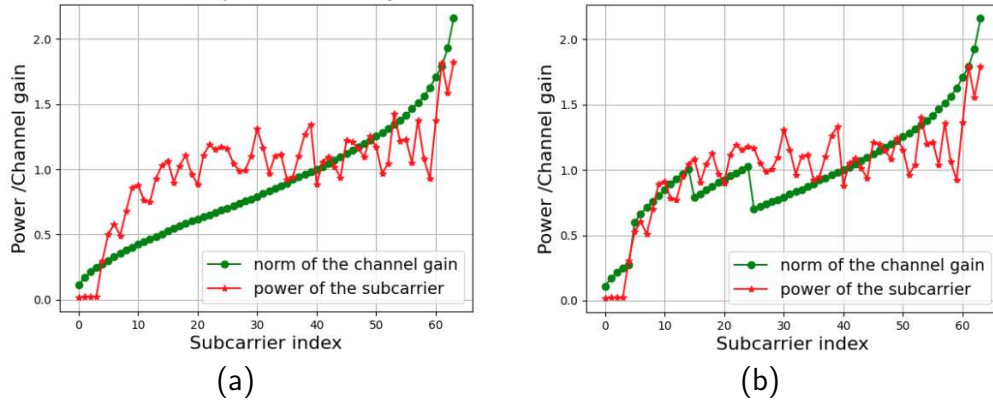


Figure 3.12: Power allocation for CA-JSCC-CH with different channel gain distribution.

Fig. 3.12(a) shows that the power allocation resembles an average power allocation strategy for different subcarriers. From Fig. 3.12(b), we can see that the CA-JSCC-CH cannot flexibly adapt to the sudden changes in the channel conditions, which also shows the power allocation is mainly based on the source signal instead of the CSI.

In summary, we can argue that the *spatial* attention mechanism does indeed carry out power allocation across subcarriers exploiting the estimated CSI.

3.5 Conclusion

This chapter presents our work on adaptive JSCC image transmission over OFDM channels. We proposed a CA-JSCC scheme for OFDM channels, where a dual-attention mechanism is employed to exploit the available CSI to simultaneously aid the allocation of features and power resources adaptively. The dual-attention mechanism comprises a *channel-wise* attention module, which learns to map features to subcarriers according to the CSI, and a *spatial* attention module, which learns high-level spatial features, allowing the JSCC encoder to match the most important features with the most reliable subcarriers. Numerical experiments show that our method achieves state-of-the-art performance in various SNR and bandwidth scenarios.

Chapter 4

DeepJSCC for adaptive image transmission over MIMO channels

4.1 Introduction

While the DeepJSCC schemes for image transmission have demonstrated significant advantages over AWGN [15] and OFDM channels [32, 41], as shown in previous chapter, existing DeepJSCC schemes solely consider single-antenna transmitters and receivers. While there is a growing literature successfully employing DNNs for various MIMO-related tasks in wireless communications, such as detection, channel estimation, or beamforming [30, 113, 114], no previous work has so far applied DeepJSCC to the more challenging MIMO scenario, with the exception of our initial results in [43]. JSCC over MIMO channels was studied in [18] 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.

The 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 [115]. In the subsequent work [116], the authors systematically establish the symbol error rate benchmarks for MIMO channels by evaluating several DAE-based models with CSI. A singular-value decomposition (SVD) based DAE is proposed in [117] 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 DL technologies. However, we remark that the aforementioned works focus on the transmission of bits/one-hot vectors for specific SNR values, and do not consider the contextual features of the input signals, also referred to as ‘semantics’ in the literature [50, 86], and the model’s adaptability to varying channel conditions, as outlined in Table 4.1. Two recent studies that are more relevant to this work are [118] and VST [119], where [118] theoretically analyzes the excess distortion exponent for semantic-aware MIMO systems, while the authors in [119] 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.

This chapter 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 [8, 33, 41, 47, 120], 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

Table 4.1: Overview of alternative end-to-end MIMO communication pipelines.

Potential Schemes	End-to-end AE [115]	DAE methods [116]	SVD-DAE [117]	VST [119]	Our work
Input	One-hot input	One-hot input	Bits	Images	Images
CSIR	✓	✓	✗	✓	✓
CSIT	✓	✓	✓	✗	✓
Antenna flexibility	✗	✗	✗	✗	✓
Channel adaptability	✗	✗	✓	✓	✓
Channel robustness	✓	✗	✗	✗	✓
Metrics	Bit error rate	Block error rate	Bit error rate	PSNR	PSNR & LPIPS
Baseline	SVD+QPSK	STBC*/SVD+GS	Plain-DAE	BPG+LDPC	DL-compression+Capacity
Results	Outperform the baseline for CSIR; Outperform the baseline for CSIT. [†]			Outperform the baseline	Outperform the baseline
Insights	Pioneered the DAE-based MIMO	Benchmark end-to-end MIMO systems	Evaluate the model-driven DAE	Semantic coding over CQI [‡]	First unified DeepJSCC over MIMO

* STBC represents space-time block codes; GS represents geometrically-shaped signal constellations with power loading; 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 [116].

[‡] 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.

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 chapter 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.

4.2 System Model

We consider an $M \times M$ MIMO system, where an M -antenna transmitter aims to deliver an image $\mathbf{S} \in \mathbb{R}^{h \times w \times 3}$ to an M -antenna receiver. Note that we consider a standard scenario $N_t = N_r = M$ for analytical simplicity. However, the proposed JSCC scheme is not restricted to this setting and can be readily extended to configurations with unequal antenna numbers. The transmitter encodes the image into a vector of channel symbols $\mathbf{X} \in \mathbb{C}^{M \times k}$, 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 $\mathbf{X}$ is subject to a power constraint P_s as: $\frac{1}{Mk} \|\mathbf{X}\|_F^2 \leq P_s$, where we set $P_s = 1$ without loss of generality.

The channel model can be written as:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{W}, \quad (4.1)$$

where $\mathbf{X} \in \mathbb{C}^{M \times k}$ and $\mathbf{Y} \in \mathbb{C}^{M \times k}$ denote the channel input and output matrices, respectively, while $\mathbf{W} \in \mathbb{C}^{M \times k}$ is the 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 $\mathbf{H} \in \mathbb{C}^{M \times M}$ also follow a complex Gaussian distribution with zero mean and variance σ_h^2 , i.e., $\mathbf{H}[i, j] \sim \mathcal{CN}(0, \sigma_h^2)$. We consider a block-fading channel model, in which the channel matrix $\mathbf{H}$ 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 $\mathbf{Y}$, the receiver reconstructs the source image as $\hat{\mathbf{S}} \in \mathbb{R}^{h \times w \times 3}$. The reconstruction quality is quantified through the PSNR and the learned perceptual image patch similarity (LPIPS) metrics [121]. The PSNR metric serves as an indicator of image distortion at the per-pixel level between $\mathbf{S}$ and $\hat{\mathbf{S}}$. 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:

$$\text{LPIPS}(\mathbf{S}, \hat{\mathbf{S}}) \triangleq \sum_L \frac{1}{H_l W_l} \sum_{h,w} \|\mathbf{w}^l \odot (\mathbf{y}^l - \hat{\mathbf{y}}^l)\|_2^2, \quad (4.2)$$

where $\mathbf{y}^l, \hat{\mathbf{y}}^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ 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, $\mathbf{w}^l \in \mathbb{R}^{C_l}$ is the weight vector, and $\odot$ is the channel-wise feature multiplication operation [121].

As shown in Fig. 4.1, there can be two approaches to solve this problem, i.e., the traditional separate source and channel coding scheme and the JSCC scheme. This chapter considers both the CSIR (open-loop MIMO) and CSIT (closed-loop MIMO) scenarios for each transmission scheme.

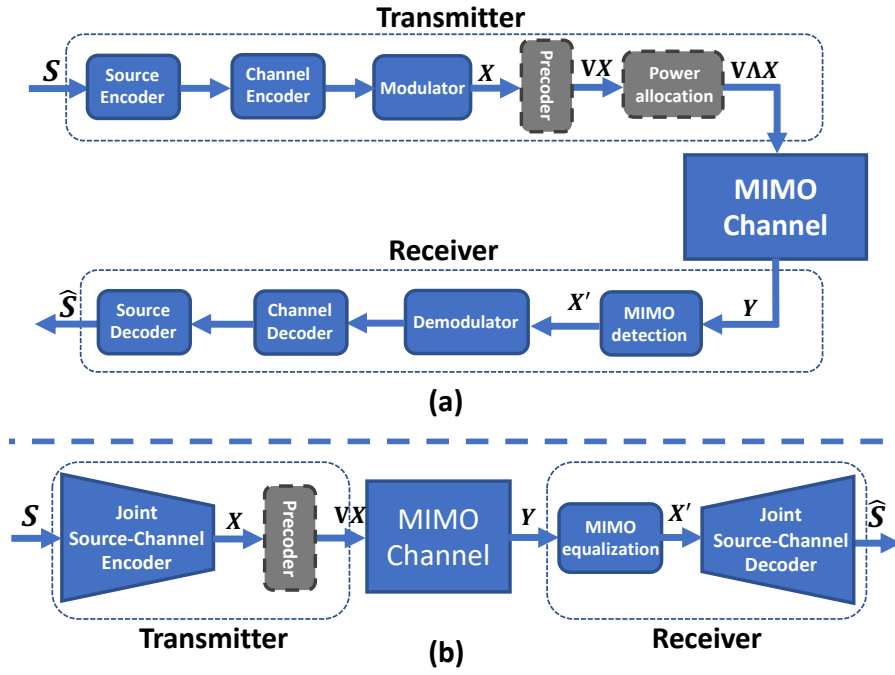


Figure 4.1: Block diagram of the MIMO image transmission system: (a) conventional separate source-channel coding scheme and (b) DeepJSCC scheme, where the gray blocks with dashed lines are the additional operations for the closed-loop MIMO system.

4.2.1 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.

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

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_{θ} , to directly map the source signal $\mathbf{S}$ into the channel input matrix $\mathbf{X}$. The encoder f_{θ} 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: $\mathbf{X} = f_{\theta}(\mathbf{S})$.

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

An alternative approach is to learn the decoding process directly 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 [32, 114].

4.2.2 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 precoding and power allocation at the transmitter, as well as MIMO equalization at the receiver, thereby improving the image transmission quality.

Separate source and channel coding scheme The transmitter sequentially performs source coding, channel coding, and modulation to generate the channel input matrix $\mathbf{X}$, 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 $\mathbf{H}$ by SVD, yielding $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, where $\mathbf{U} \in \mathbb{C}^{M \times M}$ and $\mathbf{V} \in \mathbb{C}^{M \times M}$ are unitary matrices, $\mathbf{V}^H$ is the

conjugate transpose of $\mathbf{V}$, and $\mathbf{\Sigma}$ is a diagonal matrix whose singular values are in descending order. We denote $\mathbf{\Sigma}$ by $\text{diag}(s_1, s_2, \dots, s_M)$, where $s_1 \geq s_2 \geq \dots \geq s_M$.

We denote the power allocation matrix by $\mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is diagonal, with its diagonal elements being the power allocation weights for the signal streams of separate antennas. $\mathbf{\Lambda}$ can be derived using the standard water-filling algorithm [122]. With power allocation and SVD precoding (precoding $\mathbf{X}$ into $\mathbf{V}\mathbf{X}$), Eqn. (4.1) can be rewritten as:

$$\mathbf{Y} = \mathbf{H}\mathbf{V}\mathbf{\Lambda}\mathbf{X} + \mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Lambda}\mathbf{X} + \mathbf{W}. \quad (4.3)$$

Multiplying both sides of Eqn. (4.3) by $\mathbf{U}^H$ (MIMO detection) gives us:

$$\mathbf{X}' = \mathbf{\Sigma}\mathbf{\Lambda}\mathbf{X} + \mathbf{U}^H\mathbf{W}. \quad (4.4)$$

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 $\mathbf{\Lambda}$.

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

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

Unlike the separate source-channel coding scheme, the transmitter does away with power allocation. Instead, we leverage the DeepJSCC encoder to perform feature extraction, 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. Instead, We will follow the model-driven approach, exploiting the SVD and first converting the MIMO channel into subchannels. The received signal can then be written as

$$\mathbf{Y} = \mathbf{H}\mathbf{V}\mathbf{X} + \mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{X} + \mathbf{W}. \quad (4.5)$$

To simplify the training, we apply MIMO equalization at the receiver by left multiplying both sides of Eqn. (4.5) by $\mathbf{\Sigma}^\dagger \mathbf{U}^H$ to obtain

$$\mathbf{X}' = \mathbf{\Sigma}^\dagger \mathbf{U}^H \mathbf{Y} = \mathbf{X} + \mathbf{W}', \quad (4.6)$$

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

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

$$\hat{\mathbf{S}} = f_\phi(\mathbf{X}', \mathbf{H}, \sigma_w^2). \quad (4.7)$$

In summary, our DeepJSCC framework integrates 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 the DL-based JSCC approach by exploiting domain knowledge from conventional MIMO design.

4.3 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. 4.2 and outlined in Algorithm 2. In this scheme, DeepJSCC-MIMO employs a pair of ViT-based encoder f_θ and decoder f_ϕ to encode and decode channel

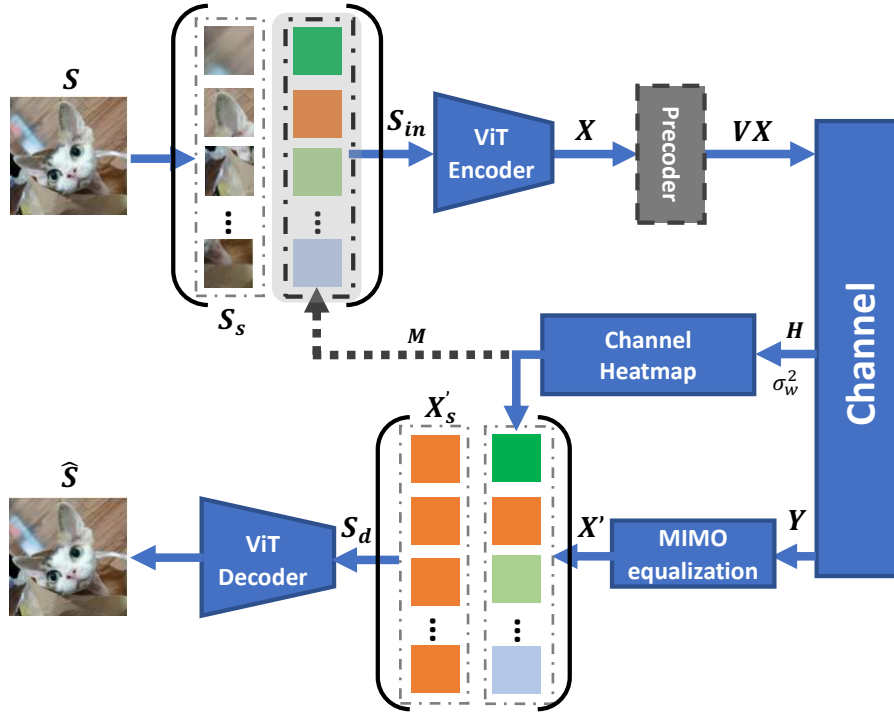


Figure 4.2: The pipeline of the DeepJSCC-MIMO scheme, where the source image $\mathbf{S}$ is encoded by a ViT-encoder and reconstructed by a ViT-decoder as $\hat{\mathbf{S}}$. Precoding operation in dashed line will be performed if the CSI is available at the transmitter, and the CSI ($\mathbf{H}, \sigma_w^2$) is fed to the encoder, if available, and decoder in the form of a “heatmap” $\mathbf{M}$ to facilitate the JSCC encoding/decoding process.

symbols, utilizing the inherent self-attention mechanism over both source features and channel conditions. Further elaborations on the inner structures of these ViTs are provided in Fig. 4.3. 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.

4.3.1 Image-to-sequence transformation

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

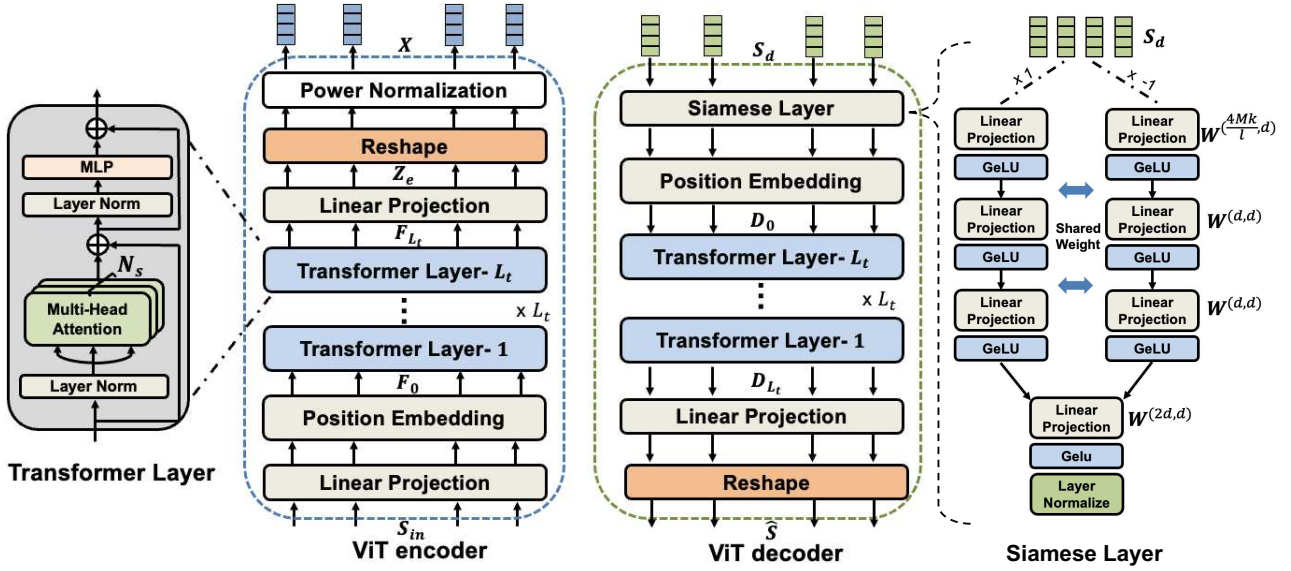


Figure 4.3: The architecture of the ViT-based encoder and decoder, where both encoder and decoder comprise linear projection layers, a positional embedding layer, and multiple transformer layers.

4.3.2 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 $\mathbf{P}_n \in \mathbb{R}^{M \times k}$ as the average power of the equivalent additive noise term $\mathbf{W}'$. Specifically, for different scenarios, $\mathbf{P}_n$ is defined as:

$$\mathbf{P}_n \triangleq \begin{cases} \sigma_w^2 (\mathbf{H}_w \odot \bar{\mathbf{H}}_w) \mathbf{J}_{M \times k}, & \text{MIMO with CSIR} \\ \sigma_w^2 \mathbf{\Sigma}^\dagger \mathbf{U}^H \mathbf{J}_{M \times k}, & \text{MIMO with CSIT,} \end{cases} \quad (4.8)$$

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

Then the heatmap $\mathbf{M} \in \mathbb{R}^{l \times \frac{2Mk}{l}}$ is constructed as follows: $\mathbf{M} \triangleq \text{reshape}(\text{concat}(\frac{1}{2}\mathbf{P}_n, \frac{1}{2}\mathbf{P}_n))$, where $\text{concat}(\cdot)$ and $\text{reshape}(\cdot)$ denote the concatenation and reshape operations, respectively. We concatenate two matrices $\frac{1}{2}\mathbf{P}_n$ to get the shape of $\mathbb{R}^{M \times 2k}$, and reshape it into $\mathbf{M} \in \mathbb{R}^{l \times \frac{2Mk}{l}}$, 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.

4.3.3 ViT encoding

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

$$\mathbf{S}_{in} = \begin{cases} \mathbf{S}_s, & \text{MIMO with CSIR} \\ \text{concat}(\mathbf{S}_s, \mathbf{M}), & \text{MIMO with CSIT,} \end{cases} \quad (4.9)$$

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. 4.3, which mainly consists of linear projection layers, a positional embedding layer, and several transformer layers.

Linear projection and positional embedding: $\mathbf{S}_{in}$ is first linearly projected by $\mathbf{W}_0 \in \mathbb{R}^{c_{in} \times d}$ followed by a positional embedding operation $P_e(\cdot)$ to get the initial input $\mathbf{F}_0 \in \mathbb{R}^{l \times d}$ for the following transformer layers:

$$\mathbf{F}_0 = \mathbf{S}_{in} \mathbf{W}_0 + P_e(\mathbf{p}), \quad (4.10)$$

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) [123] and conditional position embedding (CPE) [124]. In specific, our $P_e(\cdot)$ is implemented with a DPE that employs a dense layer to embed the index vector $\mathbf{p}$ of each patch into a d dimensional vector.

Transformer Layer: As shown in Fig. 4.3, the intermediate feature map $\mathbf{F}_i$ is generated by the i -th transformer layer by a multi-head self-attention (MSA) block and a multi-layer

perceptron (MLP) layer as:

$$\mathbf{F}_i = \text{MSA}(\mathbf{F}_{i-1}) + \text{MLP}(\text{MSA}(\mathbf{F}_{i-1})), \quad (4.11)$$

where $\mathbf{F}_i \in \mathbb{R}^{l \times d}$ is the output sequence of the i -th transformer layer, Gaussian error linear unit (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: $\text{MSA}(\mathbf{F}_i) = \mathbf{F}_i + [\text{SA}_1(\mathbf{F}_i), \dots, \text{SA}_{N_s}(\mathbf{F}_i)]\mathbf{W}_i$, where the output of all SA modules $\text{SA}(\mathbf{F}_i) \in \mathbb{R}^{l \times d_s}$ are concatenated for a linear projection $\mathbf{W}_i \in \mathbb{R}^{d_s N_s \times d}$, $d_s = d/N_s$ is the output dimension of each SA operation. For each SA module, the operations are formulated as: $\text{SA}(\mathbf{F}_{l-1}) = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d}}\right)\mathbf{v}$, where the $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{l \times d_s}$ are the query, key, and value vectors generated through three linear projection layers $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d_s}$ as: $\mathbf{q} = \mathbf{F}_{l-1}\mathbf{W}_q$, $\mathbf{k} = \mathbf{F}_{l-1}\mathbf{W}_k$, $\mathbf{v} = \mathbf{F}_{l-1}\mathbf{W}_v$.

Linear projection and power normalization: After L_t transformer layers, we apply a linear projection $\mathbf{W}_c \in \mathbb{R}^{d \times \frac{2Mk}{l}}$ to map the output of the transformer layers $\mathbf{F}_{L_t}$ to the channel symbols as: $\mathbf{Z}_e = \mathbf{F}_{L_t}\mathbf{W}_c$, where $\mathbf{Z}_e \in \mathbb{R}^{l \times \frac{2Mk}{l}}$ is then reshaped and normalized to satisfy the power constraints to form the complex channel input symbols $\mathbf{X} \in \mathbb{C}^{M \times k}$.

4.3.4 ViT decoding

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

$$\mathbf{X}' = \begin{cases} \mathbf{H}_w \mathbf{Y} + f_{res}(\mathbf{Y}, \mathbf{H}), & \text{MIMO with CSIR} \\ \mathbf{\Sigma}^\dagger \mathbf{U}^H \mathbf{Y}, & \text{MIMO with CSIT.} \end{cases} \quad (4.12)$$

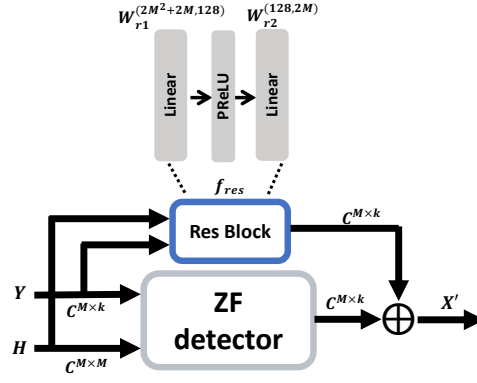


Figure 4.4: Structure of DL-aided channel equalization method, where the input at the i -th channel use is the concatenation of all elements within $\mathbf{H}$ and the channel output

To perform MIMO equalization in an open-loop MIMO system with CSIR, we designed a DL-aided equalization method to retrieve the transmitted signal $\mathbf{X}'$ from $\mathbf{Y}$, as detailed in Fig. 4.4.

Specifically, we add a residual block, denoted as $f_{res}(\cdot)$, to learn the compensation after the ZF channel equalization operation based on the received signals and CSI, given as:

$$\mathbf{X}' = \mathbf{H}_w \mathbf{Y} + f_{res}(\mathbf{Y}, \mathbf{H}) \quad (4.13)$$

$$= \mathbf{X} + \mathbf{H}_w \mathbf{W} + f_{res}(\mathbf{Y}, \mathbf{H}), \quad (4.14)$$

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

$$f_{res}(\mathbf{H}, \mathbf{Y}) \triangleq \text{PReLU}(f_{res_{in}} \mathbf{W}_{r1}) \mathbf{W}_{r2}, \quad (4.15)$$

where the elements of $\mathbf{H}$ and $\mathbf{Y}$ in each time slot are concatenated to form an input $f_{res_{in}} \triangleq \text{concat}(\mathbf{H}, \mathbf{Y}) \in \mathbb{R}^{2M^2+2M}$ 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. (4.5). At the receiver, we perform equalization as in Eqn. (4.6) to get:

$$\mathbf{X}' = \mathbf{\Sigma}^\dagger \mathbf{U}^H \mathbf{Y} = \mathbf{X} + \mathbf{\Sigma}^\dagger \mathbf{U}^H \mathbf{W}, \quad (4.16)$$

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

In summary, the proposed DeepJSCC-MIMO scheme obtains disentangled channel output $\mathbf{X}'$ via channel equalization procedures, which is shown to improve decoding performance.

Given the equalized signal $\mathbf{X}'$, which is reshaped into $\mathbf{X}'_s \in \mathbb{R}^{l \times \frac{2Mk}{T}}$, and the noise heatmap $\mathbf{M}$, a ViT-based decoder f_ϕ is designed to recover the source image as $\hat{\mathbf{S}} = f_\phi(\mathbf{X}'_s, \mathbf{M})$. Our ViT-based decoder consists of a Siamese layer, positional embedding layer, transformer layer, and linear projection layer, elaborated upon below:

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

$$\mathbf{S}_d = \text{concat}(\mathbf{X}'_s, \mathbf{M}) \in \mathbb{R}^{l \times \frac{4Mk}{T}}. \quad (4.17)$$

$\mathbf{S}_d$ multiplied by 1 and -1 are fed into several linear projection layers and GeLU functions, as illustrated in Fig. 4.3. 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 [8]. To introduce the positional information in the decoding process, we use the same positional embedding layer as in the Eqn. (4.10) to get the output $\mathbf{D}_0 \in \mathbb{R}^{l \times d}$: $\mathbf{D}_0 = \text{Siam}(\mathbf{S}_d) + P_e(\mathbf{p})$.

Algorithm 2 Algorithm of the DeepJSCC-MIMO scheme**For the transmitter:****Input:** $\mathbf{S} \in \mathbb{R}^{h \times w \times 3}$, $\mathbf{M} \in \mathbb{R}^{l \times \frac{2Mk}{l}}$ (MIMO with CSIT)**Output:** $\mathbf{X} \in \mathbb{C}^{M \times k}$

- 1: $\mathbf{S}_s = \text{Seq}(\mathbf{S}) \in \mathbb{R}^{l \times c}$ ▷ Image Sequentialization
- 2: **if** MIMO with CSIR **then**
- 3: $\mathbf{S}_{in} = \mathbf{S}_s$
- 4: $\mathbf{X} = f_\theta(\mathbf{S}_{in}) \in \mathbb{C}^{M \times k}$ ▷ ViT encoding
- 5: **else if** MIMO with CSIT **then**
- 6: $\mathbf{S}_{in} = \text{concat}(\mathbf{S}_s, \mathbf{M})$
- 7: $\mathbf{X} = f_\theta(\mathbf{S}_{in}) \in \mathbb{C}^{M \times k}$ ▷ ViT encoding
- 8: $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ ▷ SVD decomposition
- 9: $\mathbf{X} = \mathbf{V}\mathbf{X}$ ▷ Precoding
- 10: **end if**

Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{W}$ **For the receiver:****Input:** $\mathbf{Y} \in \mathbb{C}^{M \times k}$, $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, $\mathbf{M}$ **Output:** $\hat{\mathbf{S}} \in \mathbb{R}^{h \times w \times 3}$

- 1: **if** MIMO with CSIR **then**
- 2: $\mathbf{X}' = \mathbf{H}_w \mathbf{Y} + f_{res}(\mathbf{Y}, \mathbf{H})$ ▷ DL-aided equalization
- 3: **else if** MIMO with CSIT **then**
- 4: $\mathbf{X}' = \mathbf{\Sigma}^\dagger \mathbf{U}^H \mathbf{Y}$ ▷ SVD-based equalization
- 5: **end if**
- 6: $\mathbf{X}' \in \mathbb{C}^{M \times k} \rightarrow \mathbf{X}'_s \in \mathbb{R}^{l \times \frac{2MK}{l}}$ ▷ Reshape
- 7: $\hat{\mathbf{S}} = f_\phi(\text{concat}(\mathbf{X}'_s, \mathbf{M}))$ ▷ ViT decoding

Transformer layer: After the Siamese layer and positional embedding, $\mathbf{D}_0$ is passed through L_t transformer layers, where

$$\mathbf{D}_l = \text{MSA}(\mathbf{D}_{l-1}) + \text{MLP}(\text{MSA}(\mathbf{D}_{l-1})), \quad (4.18)$$

where $\mathbf{D}_i \in \mathbb{R}^{l \times d}$ 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. (4.11).

Linear Projection: Given the output of the L_t -th transformer layer $\mathbf{D}_{L_t}$, we apply a linear projection $\mathbf{W}_{out} \in \mathbb{R}^{d \times c}$, and then reshape the output into a matrix of size $\mathbb{R}^{h \times w \times 3}$ to reconstruct the input image as: $\hat{\mathbf{S}} = \text{reshape}(\mathbf{D}_{L_t} \mathbf{W}_{out})$.

4.3.5 Loss function

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

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \text{MSE}(\mathbf{S}, \hat{\mathbf{S}}) + \lambda \cdot \text{LPIPS}(\mathbf{S}, \hat{\mathbf{S}}), \quad (4.19)$$

where we introduced an additional LPIPS term [125] 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: $(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = \arg \min_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})]$, where the expectation is taken over the image and channel datasets.

4.4 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.

4.4.1 Experimental setup

For the practical separation-based coding scheme, we use the BPG codec [39] as well as the state-of-the-art neural compression algorithms [5, 64, 65, 66, 67] 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,

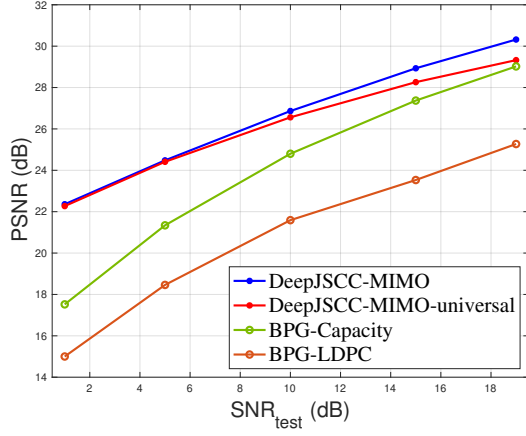
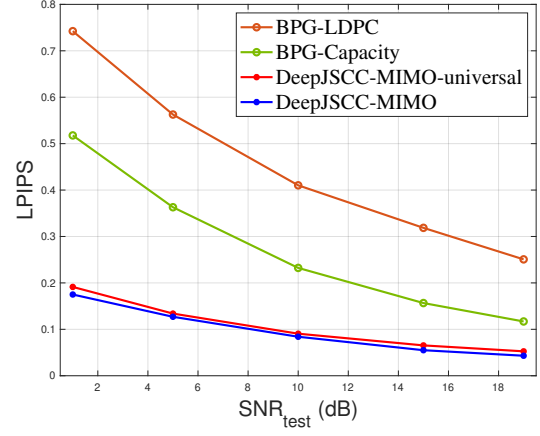
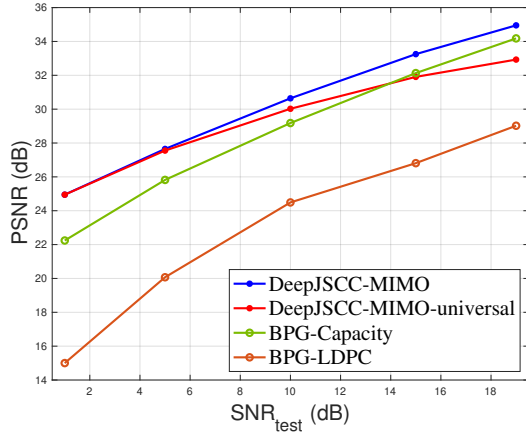
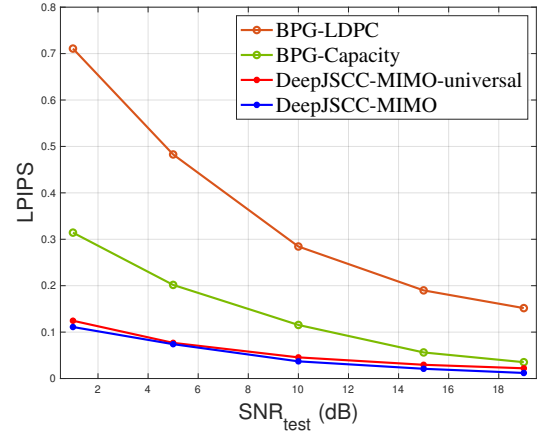
(a) $R = 1/24 - PSNR$ (b) $R = 1/24 - LPIPS$ (c) $R = 1/12 - PSNR$ (d) $R = 1/12 - LPIPS$

Figure 4.5: Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the MIMO systems with CSIR.

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

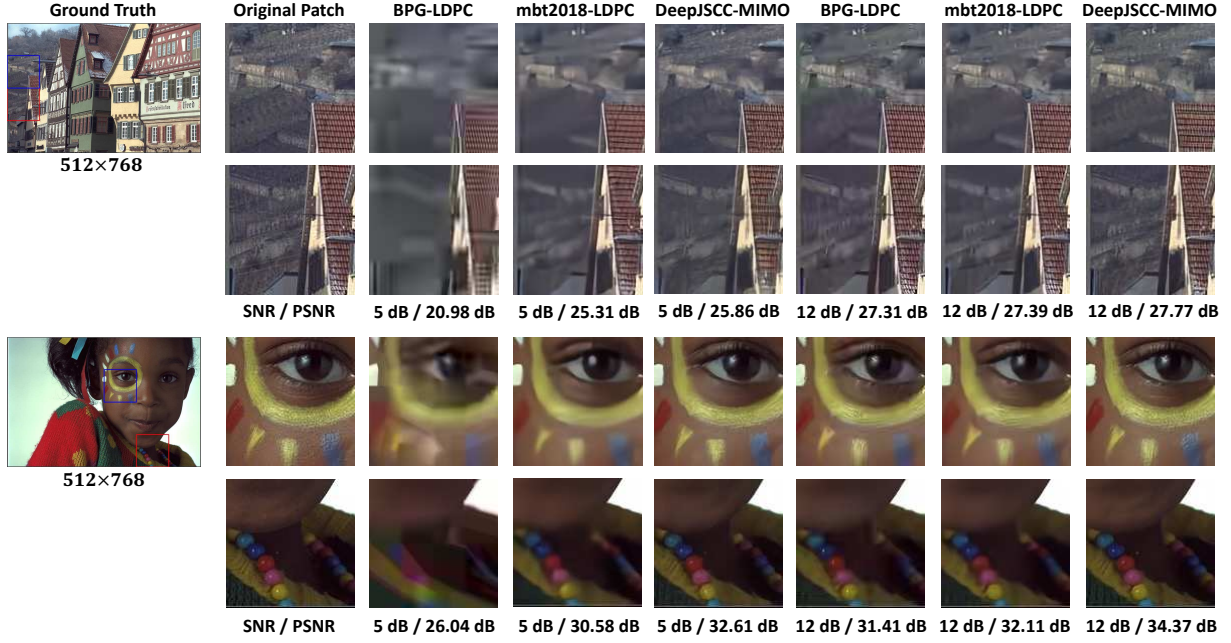


Figure 4.6: Visual comparisons of images from the Kodak dataset transmitted over the open-loop MIMO channel at SNR 5 dB and 12 dB with $R = 1/24$. The first and second columns are the original image and the original patches of the bounding boxes.

open-loop 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 $\mathbf{H}$ is sampled from a complex Gaussian process as $H[i, j] \sim \mathcal{CN}(0, 1)$, where we set $\sigma_h^2 = 1$. To measure the channel quality, we define the channel SNR, denoted by μ , as:

$$\mu \triangleq 10 \log_{10} \frac{\mathbb{E}_{\mathbf{H}, \mathbf{X}}[\|\mathbf{H}\mathbf{X}\|_F^2]}{\mathbb{E}_{\mathbf{W}}[\|\mathbf{W}\|_F^2]} \text{ (dB)} = 10 \log_{10} \frac{M}{\sigma_w^2}. \quad (4.20)$$

4.4.2 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 [126].

General performance: Fig. 4.5a and Fig. 4.5c 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. 4.5b and 4.5d 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.

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

Table 4.2: 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[39]	24.05	25.50	27.02	28.21	29.70
MBT2018-LDPC[64]	24.48	26.03	27.64	28.91	30.39
VTM-LDPC[65]	24.81	26.31	27.91	29.15	30.68
ELIC-LDPC[66]	24.99	26.54	28.16	29.35	30.85
DeepJSCC-MIMO	29.43	30.45	31.17	31.56	31.92

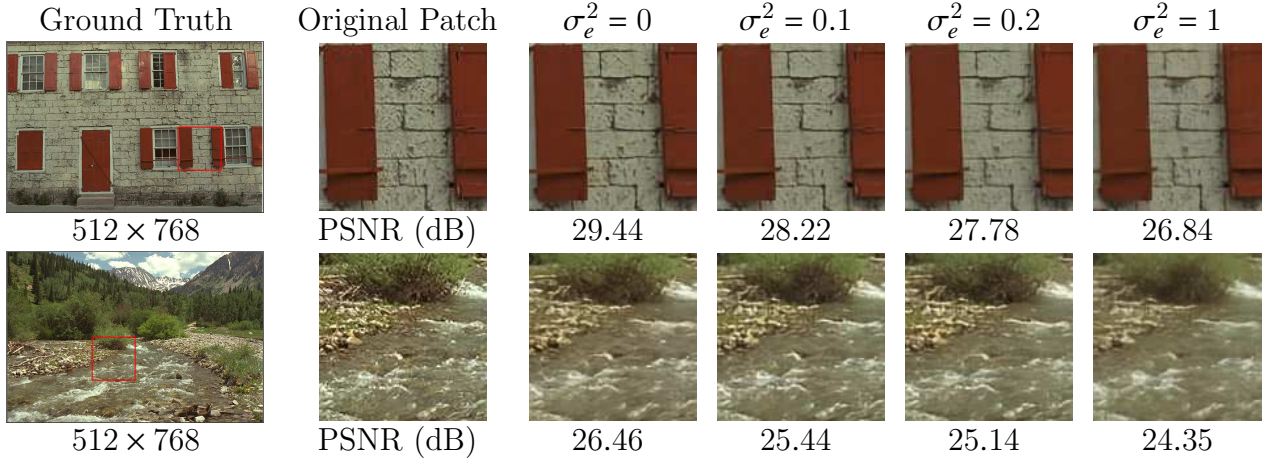


Figure 4.7: Visual comparisons of images reconstructed by DeepJSCC-MIMO-universal in the presence of channel estimation errors, where the model is trained on the ImageNet dataset and validated on the Kodak dataset at $SNR=12$ dB and $R = 1/24$. The first and second columns are the source image and the 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.

test the well-trained model at different SNRs. The comparisons in Fig. 4.5 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.

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 general-

izability, 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.

In Fig. 4.6, 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[5, 64, 65, 66, 67] when SNR=5 dB over the Kodak dataset, as shown in Table. 4.2. 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.

Robustness to channel estimation errors: We take the same approach as in [127] to evaluate the robustness of MIMO systems to channel estimation errors, where $\mathbf{H}$ is imperfectly estimated as $\hat{\mathbf{H}}$ at the receiver. Similar to the work in[127, 128, 129], let $\mathbf{E}_n \triangleq \mathbf{H} - \hat{\mathbf{H}}$, where the entries of $\mathbf{E}_n$ 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 $\sigma_e^2 \in$

Table 4.3: 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

$\{0, 0.2, 1\}$ 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 $\hat{\mathbf{H}}$. The experimental results are shown in Table. 4.3. 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.

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. 4.7. 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.

Model efficiency and other architectures: The proposed DeepJSCC-MIMO scheme is versatile and can be seamlessly integrated with various network architectures. Table 4.4 presents the model size, floating point operations (FLOPs), PSNR performance, and computational complexity for alternative DeepJSCC architectures with different block numbers when transmitting

Table 4.4: 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	$O(4ld^2 + 2l^2d)$		
	DeepJSCC-CNN	$O(k^2F^2C_{in}C_{out})$		
	DeepJSCC-Swin	$O(4ld^2 + 2M_w^2ld)$		
Encoding time / decoding time	BPG-LDPC (CPU)	37.25 / 71.28		
	Proposed method (GPU)	4.82 / 5.16		

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[32, 41] 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[34, 119] 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 tra-

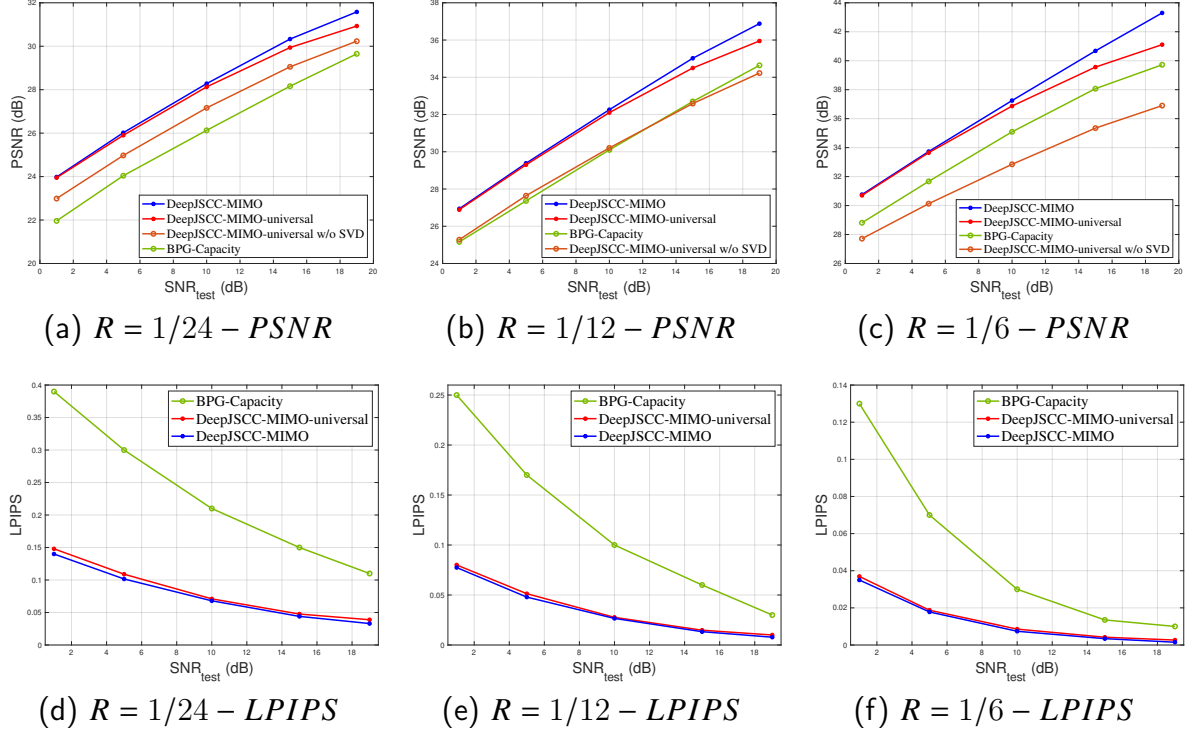


Figure 4.8: Performance of the proposed DeepJSCC-MIMO model compared with the BPG-Capacity benchmark at different channel SNR and bandwidth ratio scenarios for the closed-loop MIMO.

ditional 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.

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

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. 4.8.

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. 4.8d, 4.8e and 4.8f that DeepJSCC-MIMO achieves the best perceptual performance across a wide range of channel conditions and bandwidth ratios, particularly in the low SNR regimes.

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. 4.8 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.

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 the high-SNR regime.

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,

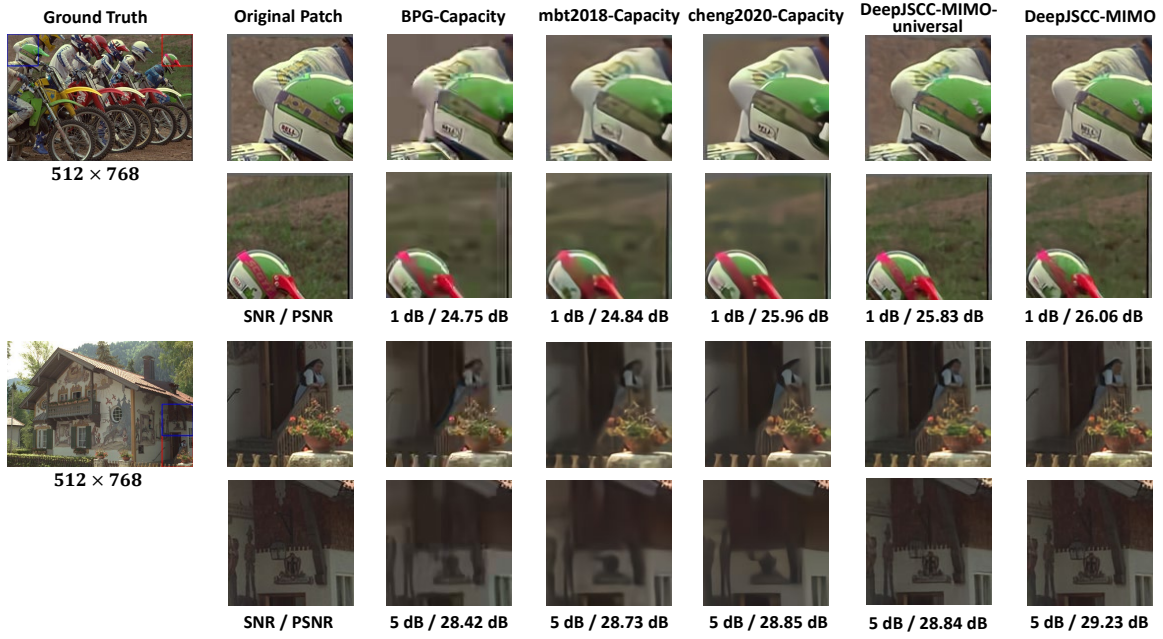


Figure 4.9: Visual comparison of different schemes on two samples from the Kodak dataset at a bandwidth ratio $R = 1/24$ and different SNRs. The first and second columns are the original image and the original patch in the red and blue bounding boxes. The third, fourth, and fifth columns are the patch reconstructed by the BPG-Capacity, MBT2018-Capacity, and Cheng2020-Capacity schemes, respectively. The last two columns demonstrate the patch recovered by the DeepJSCC-MIMO-universal and DeepJSCC-MIMO schemes.

we directly feed the channel heatmap $\mathbf{M}$ into both transceivers and the performance is shown in Fig. 4.8. 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.

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. 4.9 and Fig. 4.10, respectively.

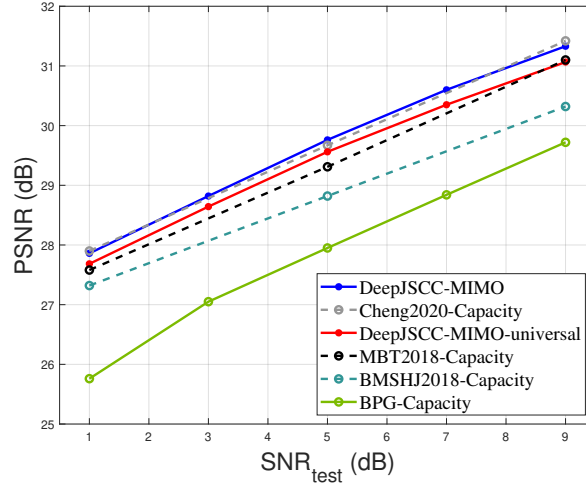


Figure 4.10: Performance of DeepJSCC-MIMO and conventional separation-based scheme in the CelebA dataset when $R = 1/48$.

Similarly to the CIFAR dataset, from Fig. 4.10, 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 [67], MBT2018-Capacity [64], and BMSHJ2018-Capacity schemes [5]. 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.

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. 4.9. 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).

Table 4.5: Ablation study of DeepJSCC-MIMO-universal with/without channel heatmap over the CIFAR10 dataset with CSIR.

Open-loop MIMO system		Channel SNRs				
Ratio	Models	1 dB	5 dB	10 dB	15 dB	19 dB
1/24	DeepJSCC-MIMO-universal	22.27	24.41	26.56	28.26	29.33
	DeepJSCC-MIMO-universal w/o heatmap	21.83	24.00	26.07	27.64	28.53
1/12	DeepJSCC-MIMO-universal	24.95	27.56	30.02	31.91	32.93
	DeepJSCC-MIMO-universal w/o heatmap	24.22	26.69	29.10	30.89	31.86

Table 4.6: Ablation study of DeepJSCC-MIMO-universal with/without channel heatmap over the CIFAR10 dataset with CSIT.

Closed-loop MIMO system		Channel SNRs				
Ratio	Models	1 dB	5 dB	10 dB	15 dB	19 dB
1/24	DeepJSCC-MIMO-universal	23.95	25.90	28.13	29.94	30.93
	DeepJSCC-MIMO-universal w/o heatmap	23.48	25.63	27.75	29.34	30.24
1/12	DeepJSCC-MIMO-universal	26.89	29.30	32.10	34.50	35.95
	DeepJSCC-MIMO-universal w/o heatmap	26.21	28.71	31.36	33.41	34.67
1/6	DeepJSCC-MIMO-universal	30.7	33.65	36.87	39.56	41.11
	DeepJSCC-MIMO-universal w/o heatmap	29.19	32.03	35.02	37.31	38.63

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. 4.5 and 4.6 for open-loop and closed-loop MIMO systems, respectively. 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. 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. 4.7.

Table 4.7: 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

In summary, the channel heatmap serves two objectives: a) it provides CSI for the transmitter/receiver, enabling the model to adapt to diverse channel conditions; b) it simplifies the learning process for the transformer model by presenting the channel condition that each channel symbol faces after specific equalization operations for the attention computation. As a direct result, it yields substantial performance improvements for the DeepJSCC-MIMO scheme, as seen from the ablation studies above. The integration of the above model-driven designs forms our DeepJSCC-MIMO approach, each of them contributing to the competitive transmission performance while maintaining an acceptable computational complexity. It is important to note that each constituent of this integrated approach exerts a distinct influence. Notably, the equalization operation emerges as a pivotal element in the MIMO channel, and its absence would render the proposed DeepJSCC scheme ineffective, similar to its role in traditional systems. While the SVD, heatmap design, and various other architectural choices substantially improve the performance, we can still surpass the performance of traditional separation-based benchmarks even in their absence.

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. 4.11. Within our channel heatmap, we visualize the equivalent noise power experienced by each channel symbol, with the color red indicating higher noise power. 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-

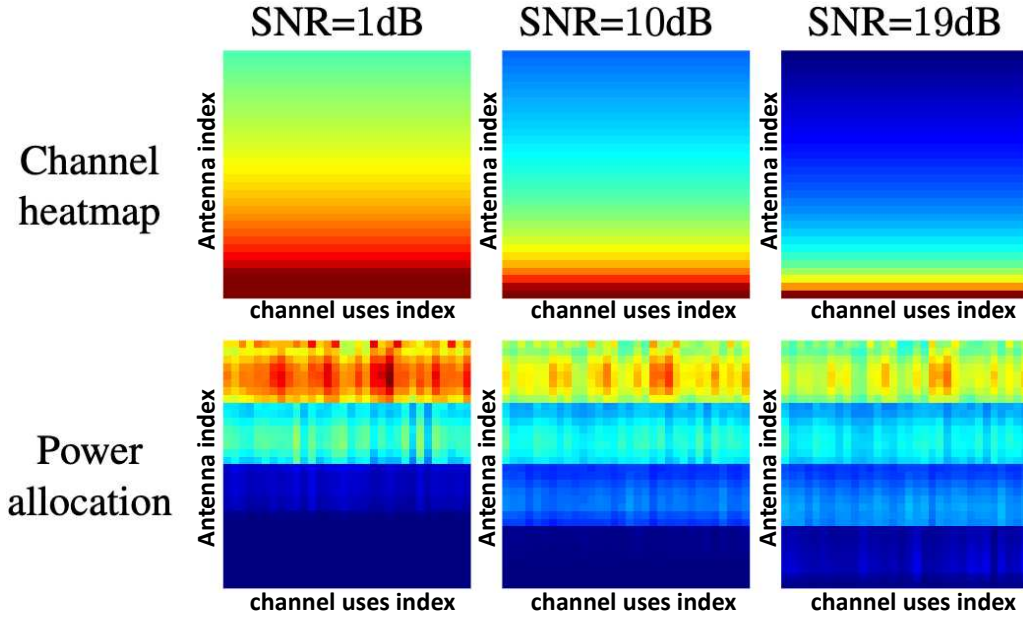
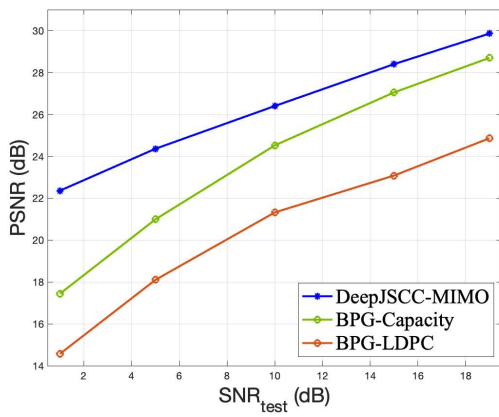
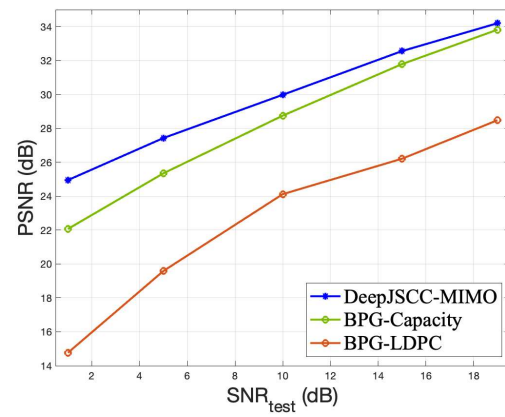


Figure 4.11: Visualization of the channel heatmap and power allocation across the channel symbols, where antenna number is set as $M = 32$ and $R = 1/6$

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.



(a) $R = 1/24$



(b) $R = 1/12$

Figure 4.12: Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the open-loop Kronecker MIMO channels with CSIR.

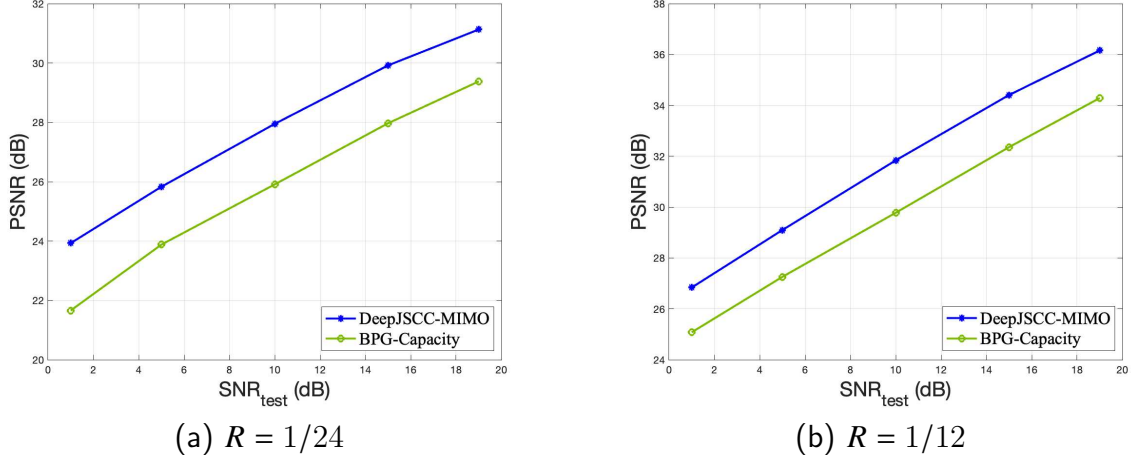


Figure 4.13: Performance comparisons between the proposed DeepJSCC-MIMO and BPG-Capacity over different SNR values and bandwidth ratios for the closed-loop Kronecker MIMO channels with CSIT.

Experiments over realistic correlated MIMO channel models: To investigate a more realistic correlated MIMO channel, we have added supplementary experiments over the MIMO channels for both open-loop and closed-loop MIMO systems. Specifically, following [119] and [118], we employed the transmitter SNR and the Kronecker channel model[130], given as:

$$\mathbf{H} = (\mathbf{R}_H^R)^{1/2} \mathbf{G} (\mathbf{R}_H^T)^{1/2}, \quad (4.21)$$

where the channel covariance is approximated by the Kronecker product of the covariance matrices from the transmit and receive sides. $\mathbf{R}_H^T = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$, $\mathbf{R}_H^R = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, and each element from $\mathbf{G}$ is sampled from a standard Gaussian complex distribution.

The experimental results of DeepJSCC-MIMO in the context of spatially correlated channel models for both open-loop MIMO and closed-loop MIMO systems are presented in Figs. 4.12 and 4.13. It is evident from the figures that the superiority of DeepJSCC-MIMO continues to hold in this channel scenario as well.

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 par-

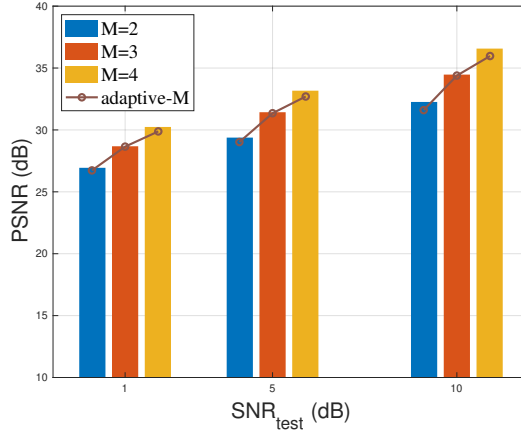


Figure 4.14: Performance of DeepJSCC-MIMO with different antennas number.

ticular, we repeat the experiment of DeepJSCC-MIMO in Fig. 4.8b 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. 4.14. 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 $\{2, \dots, M_{max}\}$ and test the well-trained model with different M values. Specifically, our encoder generates the channel symbols $\mathbf{X} \in \mathbb{C}^{M_{max} \times k}$, 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. 4.14 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.

4.5 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.

4.5.1 Future work

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

- 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 [44]. 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.

Chapter 5

DeeJSCC for feedback-aided semantic communication

5.1 Introduction

Building upon the demonstrated advantages of DeepJSCC in various standard communication channels, such as AWGN, OFDM, and MIMO channels, this chapter investigates its application over feedback channels. In the asymptotic regime, the optimality of separation continues to hold in the presence of feedback [131]. Moreover, feedback does not increase the capacity of a memoryless channel [132, 133]. However, feedback can simplify the coding mechanism, as demonstrated in [134, 135]. In addition, in the practical finite blocklength regime, the optimal transmission scheme is wide open also in the presence of feedback. Interestingly, in [135], it was shown that when transmitting a single Gaussian source sample over finitely many uses of a Gaussian channel, a JSCC scheme with perfect channel output feedback is shown to achieve the optimal performance, although the optimal performance without feedback remains unknown. This shows that feedback in practice can be highly beneficial in both improving the performance and simplifying the coding scheme [78].

It is noteworthy that only a limited number of studies have addressed the JSCC problem with channel feedback in the literature. In most of the earlier existing works, a scalable compression

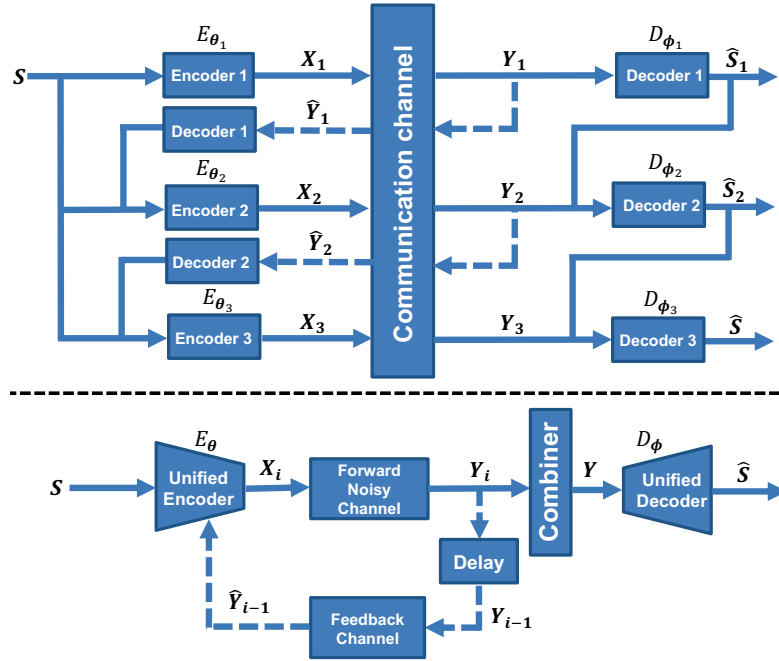


Figure 5.1: Illustration of alternative JSCC schemes for channels with feedback, where the solid lines represent forward channels, while the dotted lines represent feedback channels. Top: Illustration of the DeepJSCC-f scheme (with $m = 3$ blocks as an example), where multiple independent encoders and decoders are trained as the channel codes in each block. Bottom: Illustration of our proposed JSCCformer-f scheme, where a single unified encoder-decoder pair is employed at each block, significantly reducing the training and memory requirements as well as the coding complexity.

scheme is considered for the source (e.g., image or video), and an unequal error protection scheme is employed for the channel code. These schemes typically employ separate source and channel codes, with their rates adjusted based on channel feedback. For instance, [136] and [137] exploit one-bit acknowledgment/negative acknowledgment (ACK/NACK) feedback to adjust the channel code rate. In [138], CSI is considered as feedback to enable the transmission of the source signal in a multicarrier scenario. In [139], video signals are considered, and an automatic repeat request (ARQ) scheme is proposed to jointly allocate source and channel rates to video blocks. Similar rate allocation problems have also been studied in [140, 141, 142] for streaming media.

The first truly joint coding scheme for transmitting images over a wireless channel with feedback was proposed in [35]. The authors presented a stacked autoencoder structure, called DeepJSCC-f, to transmit the image in multiple blocks. As illustrated in Fig. 5.1, DeepJSCC-f utilizes a set of encoders and decoders in the transmitter and receiver, respectively, and a different pair of

encoder and decoder is used at each iteration. It is shown in [35] that feedback can improve the performance of CNN-based DeepJSCC, and the proposed DeepJSCC-f scheme achieves state-of-the-art performance. Even more significant gains are exhibited in [35] when variable-length coding is considered.

The design of DeepJSCC-f, however, exhibits a few limitations that we wish to address in this chapter.

- **High complexity.** DeepJSCC-f trains and employs independent encoders and decoders for each interaction between the transmitter and receiver. This results in a substantial surge in both training complexity and memory requirements when the number of feedback blocks increases. A practical JSCC scheme should be able to effectively exploit channel feedback with low complexity.
- **Inadaptability.** Adaptability to varying channel conditions is another critical property for learning-based JSCC schemes. Specifically, training encoder and decoder network parameters for specific channel conditions, as demonstrated in prior works [15, 35, 84], would necessitate the storage of multiple sets of network parameters. Alternatively, a JSCC scheme can achieve adaptability to channel conditions by leveraging the attention mechanism and training at random SNRs [33, 41, 43]. This approach enables dynamic scaling of the features according to the channel conditions. Furthermore, it should be noted that all of these methods require the acquisition and feedback of CSI, which may not always be feasible. To the best of our knowledge, the direct learning of channel adaptability from noisy channel feedback has not yet been explored.
- **Suboptimality.** The training of DeepJSCC-f is conducted in a block-by-block fashion, where pairs of encoder and decoder are trained progressively. This leads not only to training instability, but also to suboptimal performance. In addition, the advent of novel DL architectures [123, 124, 143] with respect to image processing and natural language processing, has rendered the CNNs-enabled encoder and decoder as suboptimal alternatives.

- **Non-generalizability.** DeepJSCC-f was designed exclusively for point-to-point channels with feedback. On the other hand, we anticipate a more versatile encoding and decoding JSCC scheme that can be generalized to other communication scenarios, such as multiple access channels and broadcast channels, with feedback mechanisms.

This chapter presents a novel JSCC paradigm with feedback, called JSCCformer-f, that effectively addresses the concerns outlined above. Unlike DeepJSCC-f, JSCCformer-f utilizes a unified encoder-decoder pair at the transmitter and receiver, respectively. Specifically, the transmission of source signals is divided into multiple blocks, and the channel feedback from previous blocks is used to facilitate the encoding process. The unified encoder leverages the semantic information from the source image, the CSI, and the feedback (i.e., the decoder’s current belief about the source image) to generate coded symbols at each interaction. These coded symbols are then transmitted to the receiver to refine its belief and mitigate the impact of channel noise in subsequent interactions. Finally, the unified decoder accumulates the received successive coded blocks and reconstructs the source image after the last round of interaction. Overall, an image is transmitted to the receiver with multiple blocks in a coarse-to-fine fashion, and the well-trained model exhibits channel adaptability, thanks to the encoder’s ability to attend to feedback and CSI.

Furthermore, JSCCformer-f is built upon the ViT architecture [123, 143] as the unified encoder and decoder. Unlike CNNs that learn the semantic features of images from a local to global context with a limited receptive field, ViTs employ a global self-attention (SA) mechanism for a more discriminative representation of semantic features. This transformer-based SA strategy naturally fits our encoding and decoding process with feedback, where we rethink the image-to-channel mapping with interactions from a sequences-to-sequence perspective, inspired by the feedback channel code design in [120].

Our main contributions can be summarized as follows:

- We present a new feedback-aided JSCC architecture, JSCCformer-f, for wireless image transmission with channel feedback. JSCCformer-f exploits the semantics of the source

image and the available feedback signal in each interaction to generate coded symbols and to refine the decoder's belief about the source image. JSCCformer-f introduces a new way of utilizing the semantics and the feedback through an SA mechanism of the transformer architecture. Although it is used here for image transmission, this framework is general and can be used to convey any desired semantic information [50, 144].

- Our scheme is a resource-friendly pipeline, employing only a single unified encoder to accommodate a wide range of interaction numbers without introducing additional parameters or training effort. Flexible strategies, such as variable rate transmission, are also enabled to allow a considerable economy of resources for attaining a specific quality objective of the transmission. The proposed JSCCformer-f scheme can also directly learn to adapt the various channel conditions from the noiseless/noisy channel feedback and achieves considerable gains compared to all the previous methods.
- Numerical experiments verify that our JSCCformer-f scheme achieves state-of-the-art performance in all SNRs and bandwidth scenarios. We also show its robustness to feedback channel noise.
- To demonstrate the generalizability of the proposed scheme, we extend the JSCCformer-f framework to a broadcast channel with two receivers, where the encoder at the transmitter has to attend to the feedback from both receivers and generate coded symbols accordingly.

5.2 System Model

We consider the problem of wireless transmission of images in the presence of noisy channel output feedback. The goal is to transmit an input image $\mathbf{S} \in \mathbb{R}^{h \times w \times 3}$ over a noisy communication channel with the help of the channel output feedback, where h, w and 3 denote the height, width, and color channels of an RGB image. The transmission of each image is done in m blocks, where $m = 1$ corresponds to the no-feedback scenario. We assume that after each block, the channel output is fed back to the transmitter via a noisy or noiseless feedback link. At each block, the transmitter can exploit the feedback signals it has received up until that block.

Accordingly, for a fixed total bandwidth allocation, a JSCC code that transmits an image of n pixel intensities in m blocks, each consisting of k channel symbols, is denoted by the triplet (n, m, k) . Here $n = 3hw$ denotes the source bandwidth, while mk denotes the total channel bandwidth cost. We note that, for a fixed bandwidth ratio R , an increase in the number of blocks leads to a reduction in the number of channel uses per block. Following [35], we define the *bandwidth ratio* as $R = mk/n$, representing the average number of available channel symbols per source dimension.

As shown in Fig 5.1, in the i -th transmission block, the encoder function E_{θ_i} , parameterized by θ_i , maps $\mathbf{S}$ and all the available feedback signals into a complex channel vector $\mathbf{X}_i = E_{\theta_i}(\mathbf{S}, \hat{\mathbf{Y}}_{1:i-1}) \in \mathbb{C}^k$, where $\hat{\mathbf{Y}}_{1:i-1} \triangleq [\hat{\mathbf{Y}}_1, \dots, \hat{\mathbf{Y}}_{i-1}]$ denotes all the previous feedback signals available at the i -th transmission block, and $\hat{\mathbf{Y}}_i \in \mathbb{C}^k$ is the channel output feedback received after the i -th block. Note that the encoder, in general, can consist of m encoding functions, one for each block. The transmitted signal $\mathbf{X}_i$ is subject to a power constraint P_s as:

$$\frac{1}{mk} \sum_{i=1}^m \mathbb{E} [\|\mathbf{X}_i\|_2^2] \leq P_s, \quad (5.1)$$

where we set $P_s = 1$ without loss of generality.

The forward channel in the i -th transmission block is modeled as an additive white Gaussian noise (AWGN) channel with fading as:

$$\mathbf{Y}_i = \mathcal{H}(\mathbf{X}_i) = \begin{cases} h\mathbf{X}_i + \mathbf{W}_i, & \text{slow fading channel} \\ \mathbf{X}_i + \mathbf{W}_i, & \text{AWGN channel,} \end{cases} \quad (5.2)$$

where $\mathbf{Y}_i \in \mathbb{C}^k$ is the channel output in i -th forward transmission block, $h \in \mathbb{C}$ is the channel gain, and $\mathbf{W}_i \in \mathbb{C}^k$ is the AWGN term. Each element of $\mathbf{W}_i$ is sampled from independent and identically distributed (i.i.d.) complex Gaussian noise as $W[i] \sim \mathcal{CN}(0, \sigma_w^2)$. We consider a slow fading scenario and assume that h remains constant for the whole duration (mk channel symbols) of the transmission of one image, but takes an independent realization for each image. We assume that h is sampled from an i.i.d. complex Gaussian distribution, i.e., $h \sim \mathcal{CN}(0, \sigma_h^2)$.

For the static AWGN scenario, we set $h = 1$. In order to measure the channel quality, we define the SNR as $10 \log_{10} \frac{\sigma_h^2}{\sigma_w^2} (\text{dB})$.

After each block, the channel output vector $\mathbf{Y}_i$ is then fed back to the transmitter through a feedback link, which can be either perfect or noisy, modeled as:

$$\hat{\mathbf{Y}}_i = \mathcal{H}_f(\mathbf{Y}_i) = \begin{cases} \mathbf{Y}_i, & \text{perfect feedback link} \\ \mathbf{Y}_i + \mathbf{W}_f, & \text{AWGN feedback link,} \end{cases} \quad (5.3)$$

where $\mathbf{W}_f \in \mathbb{C}^k$ is the AWGN term sampled from an i.i.d. additive complex Gaussian distribution, i.e., $W_f[i] \sim \mathcal{CN}(0, \sigma_f^2)$.

The decoding function D_ϕ , parameterized by ϕ , reconstructs the transmitted image as $\hat{\mathbf{S}} = D_\phi(\mathbf{Y}) \in \mathbb{R}^{h \times w \times 3}$, where $\mathbf{Y} \triangleq [\mathbf{Y}_1, \dots, \mathbf{Y}_m]$.

The reconstruction quality is quantified through the PSNR and the LPIPS metrics. The PSNR metric serves as an indicator of image distortion at the per-pixel level. A higher PSNR value is indicative of a better image reconstruction quality. LPIPS computes the dissimilarity within the feature space between the input image and the reconstruction, employing a deep neural network, such as VGG. LPIPS exhibits a superior alignment with human perception compared to conventional pixel-wise metrics. A lower LPIPS score implies a better perceptual quality, indicative of a greater degree of perceptual similarity observed among image patches.

Given the code parameters (n, m, k) , we aim to optimize the encoder and decoder parameters $\theta_1, \dots, \theta_m$ and ϕ , respectively, with the objective of improving the average reconstruction performance at the receiver.

5.3 JSCCformer-f Architecture

This section introduces the proposed feedback-aided JSCC paradigm for an image transmission task, denoted by JSCCformer-f, depicted in Fig 5.2. Transmission is partitioned into m blocks, each corresponding to one interaction between the transmitter and receiver. JSCCformer-f

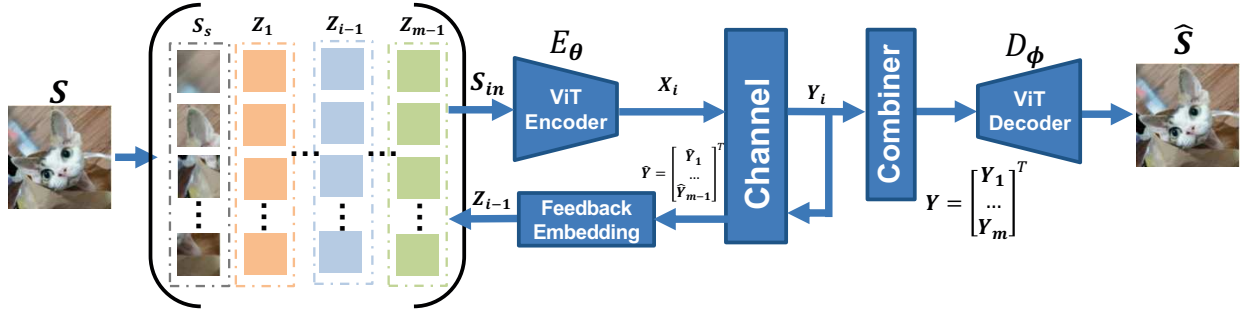


Figure 5.2: The pipeline of our JSCCformer-f scheme. In the i -th transmission block, the ViT-based encoder E_θ encodes the channel symbols $\mathbf{X}_i$ based on the input image and the channel feedback signals received until that time, where $\hat{\mathbf{Y}}_{i-1}$ is the noisy channel feedback for the $i-1$ -th transmission and $\mathbf{Z}_{i-1}$ is its embedding. Note that $(\hat{\mathbf{Y}}_i, \dots, \hat{\mathbf{Y}}_m)$ and $(\mathbf{Z}_i, \dots, \mathbf{Z}_{m-1})$ are padded with zeros as they correspond to future channel feedback signals.

utilizes a pair of ViTs, which possess symmetric inner structures, as the encoder and decoder, respectively. The architecture of JSCCformer-f is illustrated in Figure 5.3, and consists of three key components: ViT encoder, ViT decoder, and the loss function.

5.3.1 JSCCformer-f encoder

The encoding process entails sequentializing the image, embedding the feedback signal, and utilizing a ViT-based encoder E_θ to map the updated input sequence to channel symbols. The encoding process is elaborated in four steps: (1) image-to-sequence transformation, (2) feedback embedding, (3) input sequence update, and (4) ViT-based encoding. The encoding process is presented in Algorithm 3 as a pseudocode.

Image-to-sequence transformation To construct the input of JSCCformer-f at each block, we first convert the three-dimensional input image $\mathbf{S}$ into a sequence of vectors, denoted by $\mathbf{S}_s$. Specifically, given a source image $\mathbf{S} \in \mathbb{R}^{h \times w \times 3}$, we divide $\mathbf{S}$ into a grid of $p \times p$ patches, and flatten the pixel intensities of each patch to form a sequence of vectors of dimension $\mathbb{R}^{\frac{3hw}{p^2}}$. In this way, $\mathbf{S}$ is converted to $\mathbf{S}_s \in \mathbb{R}^{l \times c}$, where $l = p^2$ is the sequence length and $c \triangleq \frac{3hw}{p^2}$ is the dimension of each vector. Here p serves as a hyperparameter, where a higher value of p indicates a more detailed partition of the image into smaller patches, facilitating more fine-grained feature learning but at the expense of a corresponding rise in computational complexity.

Algorithm 3 ViT-encoding algorithm**Input:** $S_s \in \mathbb{R}^{l \times c}$ **Output:** $X_i \in \mathbb{R}^{l \times \frac{2k}{T}}, Y_i \in \mathbb{R}^{l \times \frac{2k}{T}}$ **For the 1-st block**1: **for** $i = 1:m-1$ **do**2: $\hat{Y}_i = \mathbf{0}_{l, \frac{2k}{T}} \in \mathbb{R}^{l \times \frac{2k}{T}}$ 3: **end for**4: $\hat{Y} = [\hat{Y}_1, \dots, \hat{Y}_{m-1}] \in \mathbb{R}^{l \times \frac{2k(m-1)}{T}}$ ▷ Initialize $\hat{Y}$ 5: **for** $i = 1:m-1$ **do**6: $Z_i = \mathbf{0}_{l, z} \in \mathbb{R}^{l \times z}$ 7: **end for**▷ Initialize each Z_i 8: $Z_{1:m-1} \triangleq [Z_1, \dots, Z_{m-1}] \in \mathbb{R}^{l \times (m-1)z}$ 9: $S_{in} = \text{concat}(S_s, Z_{1:m-1}) \in \mathbb{R}^{l \times (c+(m-1)z)}$ 10: $X_1 = E_\theta(S_{in}) \in \mathbb{R}^{l \times \frac{2k}{T}} \rightarrow \mathbb{C}^k$ 11: $Y_1 = \mathcal{H}(X_1) \in \mathbb{C}^k \rightarrow \mathbb{R}^{l \times \frac{2k}{T}}$

▷ Forward link

For the i -th block, where $i \in [2, m-1]$ 1: $\hat{Y}_{i-1} = \mathcal{H}_f(Y_{i-1}) \in \mathbb{C}^k \rightarrow \mathbb{R}^{l \times \frac{2k}{T}}$

▷ Feedback link

2: $\hat{Y} = [\hat{Y}_1 \dots \hat{Y}_{m-1}]$ ▷ Update $\hat{Y}$ 3: $Z_{i-1} = f_e(\hat{Y})$ ▷ Update Z_{i-1} 4: $Z_{1:m-1} = [Z_1, \dots, Z_{m-1}]$ ▷ Update $Z_{1:m-1}$ 5: $S_{in} = \text{concat}(S_s, Z_{1:m-1})$ ▷ Update input S_{in} 6: $X_i = E_\theta(S_{in}) \in \mathbb{R}^{l \times \frac{2k}{T}} \rightarrow \mathbb{C}^k$ 7: $Y_i = \mathcal{H}(X_i) \in \mathbb{C}^k \rightarrow \mathbb{R}^{l \times \frac{2k}{T}}$

Feedback embedding Within each transmission block, a feedback embedding module is employed at the transmitter, parameterized as $f_e(\cdot) : \mathbb{R}^{l \times \frac{2mk}{T}} \rightarrow \mathbb{R}^{l \times z}$, to embed all the received feedback signals $\hat{Y}$ into Z_i for the subsequent ViT-encoding step, where z is the output dimension for the feedback embedding block.

We propose two different feedback embedding schemes, JSCCformer-f and JSCCformer-f (lite), respectively. Specifically, the embedding scheme of JSCCformer-f utilizes an additional ViT-based decoder at the transmitter (the weights of which are copied from the decoder at the receiver) to estimate the receiver's current belief from the received feedback, which is then combined with the raw received feedback as the input to the encoder. During the training phase, the gradient over the decoder is stopped at the transmitter to simplify the training process, especially in the application of variable rate and successive refinement transmission. The second scheme, in contrast, directly feeds the received channel feedback to the encoder. That is why JSCCformer-f with the second embedding scheme is called JSCCformer-f (lite).

For JSCCformer-f and JSCCformer-f (lite), we have $z = \frac{2k}{l} + c$ and $z = \frac{2k}{l}$, respectively. Thanks to the integration of the receiver's belief, JSCCformer-f achieves a better performance at the cost of additional complexity.

The detailed formula of $\mathbf{Z}_i = f_e(\hat{\mathbf{Y}})$ is provided below:

$$\mathbf{Z}_i = \begin{cases} \text{concat}(\text{sg}(D_\phi(\hat{\mathbf{Y}})), \hat{\mathbf{Y}}_i), & \text{JSCCformer-f,} \\ \hat{\mathbf{Y}}_i, & \text{JSCCformer-f (lite),} \end{cases} \quad (5.4)$$

where $\hat{\mathbf{Y}}_i \in \mathbb{R}^{l \times \frac{2k}{l}}$ is the channel output feedback of the i -th block, reshaped from $\mathbb{C}^k$, $\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_{1:m} \triangleq [\hat{\mathbf{Y}}_1, \dots, \hat{\mathbf{Y}}_m] \in \mathbb{R}^{l \times \frac{2mk}{l}}$, $\text{sg}(\cdot)$ means stop gradient operation, $\text{concat}(\cdot)$ denotes concatenation, and $\mathbf{Z}_i \in \mathbb{R}^{l \times z}$ is the embedded feedback matrix given $\hat{\mathbf{Y}}_i$ for the $(i+1)$ th block. We pad $\hat{\mathbf{Y}}_t$ with a matrix of zeros when there is no knowledge of $\hat{\mathbf{Y}}_t$. That is, in the i -th block of a total m block transmission, $\hat{\mathbf{Y}}_{1:i-2}$ is given, $\hat{\mathbf{Y}}_{i-1}$ is updated, while $\hat{\mathbf{Y}}_{i:m}$ is set to zeros.

Input sequence update As shown in Algorithm 3, we concatenate $\mathbf{S}_s$ and the embedded feedback matrix $\mathbf{Z}_{1:m-1} \triangleq [\mathbf{Z}_1, \dots, \mathbf{Z}_{m-1}]$ to form the input sequence $\mathbf{S}_{in}$ of the ViT-based encoder in the i -th block as: $\mathbf{S}_{in} = \text{concat}(\mathbf{S}_s, \mathbf{Z}_{1:m-1})$, where $\mathbf{Z}_{1:m-1} \in \mathbb{R}^{l \times (m-1)z}$ and $\mathbf{S}_{in} \in \mathbb{R}^{l \times (c+(m-1)z)}$.

In each transmission block (i -th block), we update $\mathbf{Z}_{1:m-1}$ and $\mathbf{S}_{in}$, once $\mathbf{Z}_{i-1}$ generated from Eqn. (5.4) with a new $\hat{\mathbf{Y}}_{i-1}$ becomes available. Similarly, if there is no knowledge of feedback $\hat{\mathbf{Y}}_t$ in the current i -th transmission block (i.e., $t \geq i$), $\mathbf{Z}_k$ would be padded with zeros. In particular, for the i -th block of a total m transmission blocks, $\mathbf{Z}_{i-1}$ is updated while $\mathbf{Z}_{i:m-1}$ is set to zeros.

ViT-based encoding In each block, we use a ViT-based encoder $E_\theta : \mathbb{R}^{l \times (c+(m-1)z)} \rightarrow \mathbb{C}^k$, parameterized by θ , to generate the channel symbols based on the updated $\mathbf{S}_{in}$. The architecture of our ViT-based encoder is shown in Fig. 5.3, mainly consisting of linear projection layers, a position embedding layer, several transformer layers, and a power normalization layer. A triplet (L_t, N_s, d) can be utilized to characterize our proposed ViT-based encoder, wherein L_t , N_s , and d represent the number of transformer layers, SA heads, and hidden units, respectively.

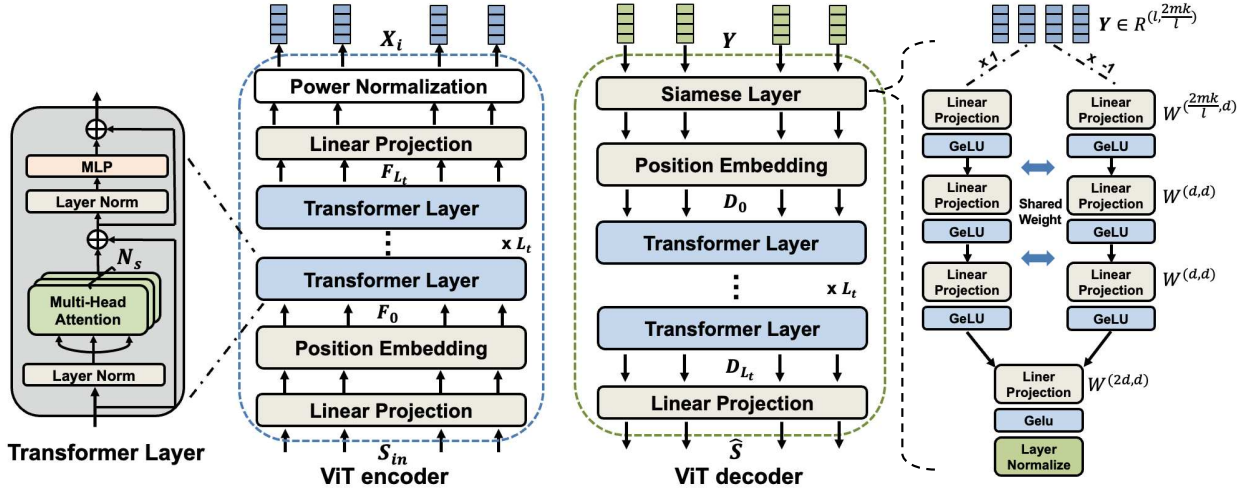


Figure 5.3: The architecture of the encoder and decoder, where a symmetric structure is designed to encode the input sequence and reconstruct the source signal.

Linear projection: The input matrix S_{in} first undergoes a linear projection with parameters $W_0 \in \mathbb{R}^{(c+(m-1)z) \times d}$, followed by a position embedding operation $P_e(\cdot) : \mathbb{R}^{l \times d} \rightarrow \mathbb{R}^{l \times d}$, resulting in the initial input $F_0 \in \mathbb{R}^{l \times d}$:

$$F_0 = S_{in}W_0 + p_e, \quad (5.5)$$

where d is the output dimension of the hidden layer, $F_0 \in \mathbb{R}^{l \times d}$ is the output of the linear projection with position embedding operation for the following transformer layers, and $p_e \in \mathbb{R}^{l \times d}$ is the output of the position embedding (PE) operation $P_e(\cdot)$.

Position embedding: Two methods are adopted for the positional embedding $P_e(\cdot)$ of JSCCformer-f: a DPE method [123] and a CPE method [124].

DPE employs a dense layer to embed the index vector $\mathbf{p}$ of each patch into a d -dimensional vector, denoted as: $\mathbf{p}_e = P_e(\mathbf{p})$. On the other hand, CPE employs a 2-D convolution with a kernel size of $k = 3$ and $\frac{k-1}{2}$ zero padding to generate positional encodings dynamically, conditioned on the local neighborhood of the input. To elaborate further, we first reshape the encoded feature $S_{in}W_0 \in \mathbb{R}^{l \times d}$ into a 2-D image of dimension $\mathbb{R}^{p \times p \times d}$, which is then fed into a 2-dimensional convolution layer. The resultant outputs are then reshaped as $\mathbf{p}_e \in \mathbb{R}^{l \times d}$. The final result of the conditional position embedding can be written as: $\mathbf{p}_e = P_e(S_{in}W_0)$.

DPE is the default positional embedding scheme for JSCCformer-f. It is lightweight and yields

satisfactory performance. In contrast, CPE requires more computation resources. It is worth pointing out that CPE is more suitable for high-resolution images.

Transformer Layer: As shown in Fig. 5.3, the intermediate feature map $\mathbf{F}_i$ is generated by the i -th transformer layer by a MSA block and a MLP layer with a residual skip as:

$$\mathbf{F}_i = \text{MSA}(\mathbf{F}_{i-1}) + \text{MLP}(\text{MSA}(\mathbf{F}_{i-1})), \quad (5.6)$$

where $\mathbf{F}_i \in \mathbb{R}^{l \times d}$ 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 SA modules with a residual skip, which can be formulated as: $\text{MSA}(\mathbf{F}_i) = \mathbf{F}_i + [\text{SA}_1(\mathbf{F}_i), \dots, \text{SA}_{N_s}(\mathbf{F}_i)]\mathbf{W}_i$, where the output of all the SA modules $\text{SA}(\mathbf{F}_i) \in \mathbb{R}^{l \times d_s}$ are concatenated for a linear projection $\mathbf{W}_i \in \mathbb{R}^{d_s N_s \times d}$, and $d_s = d/N_s$ is the output dimension of each SA operation. For each SA module, the operations can be formulated as: $\text{SA}(\mathbf{F}_{l-1}) = \text{softmax}(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d}})\mathbf{v}$, where $\mathbf{q}, \mathbf{k}, \mathbf{v} \in \mathbb{R}^{l \times d_s}$ are the query, key, and value vectors, respectively, generated from three linear projection layers $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d_s}$ as: $\mathbf{q} = \mathbf{F}_{l-1}\mathbf{W}_q$, $\mathbf{k} = \mathbf{F}_{l-1}\mathbf{W}_k$, $\mathbf{v} = \mathbf{F}_{l-1}\mathbf{W}_v$.

Linear projection and power normalization: After L_t transformer layers, we apply a linear projection $\mathbf{W}_c \in \mathbb{R}^{d \times \frac{2k}{T}}$ to map the output of the transformer layers $\mathbf{F}_{L_t}$ to channel symbols as: $\bar{\mathbf{X}}_i = \mathbf{F}_{L_t}\mathbf{W}_c$, where $\bar{\mathbf{X}}_i \in \mathbb{R}^{l \times \frac{2k}{T}}$ is then reshaped and normalized to satisfy the power constraint to form the channel input symbols $\mathbf{X}_i \in \mathbb{R}^{l \times \frac{2k}{T}}$.

5.3.2 JSCCformer-f decoder

At the decoder, we first employ a combiner to form the input sequence for the ViT-decoder by concatenating all the received signals as:

$$\mathbf{Y} \triangleq [\mathbf{Y}_1, \dots, \mathbf{Y}_m] \in \mathbb{R}^{l \times \frac{2mk}{T}}. \quad (5.7)$$

Algorithm 4 ViT-based decoder algorithm**For the transmitter of i -th block transmission:****Input:** $\hat{Y}_{i-1} \in \mathbb{R}^{l \times \frac{2k}{T}}$, **Output:** $\hat{S}_{si} \in \mathbb{R}^{l \times c}$ 1: $\hat{Y} = [\hat{Y}_1, \dots, \hat{Y}_{m-1}]$ ▷ Update $\hat{Y}$ with $\hat{Y}_{i-1}$ 2: $\hat{S}_{si} = D_\phi(\hat{Y}) \in \mathbb{R}^{l \times c}$ ▷ Reconstruct S_s for refining**For the receiver:****Input:** $Y_i \in \mathbb{R}^{l \times \frac{2k}{T}}$, **Output:** $\hat{S} \in \mathbb{R}^{h \times w \times 3}$ 1: **for** $i = 1:m$ **do**2: **if** $i == 1$ **then** $Y = Y_1$ 3: **else** $Y = \text{concat}(Y, Y_i)$ 4: **end if**5: **end for**▷ Combine all received Y_i 6: $\hat{S} = D_\phi(Y) \in \mathbb{R}^{l \times c} \Rightarrow \mathbb{R}^{h \times w \times 3}$ ▷ Reconstruct the S

While we use the concatenation operation as a combiner for simplicity, we could also use a learning-based method, such as CNNs or a fully connected layer.

Given Y , a symmetrical ViT-based decoder $D_\phi : \mathbb{R}^{l \times 2mk} \rightarrow \mathbb{R}^{l \times c}$, parameterized by ϕ and the same triplet (L_t, N_s, d) with the encoder, is designed to reconstruct the source image as $\hat{S} = D_\phi(Y)$, consisting of a Siamese layer, position embedding layer, transformer layer and linear projection layer. Each component of the ViT decoder is explained next, also presented as pseudocode in Algorithm 4.

Siamese layer and position embedding: We employ a weight-shared Siamese layer $\text{Siam}(\cdot)$ consisting of several linear projection layers and GeLU activation functions. As shown in Fig. 5.3, Y multiplied by 1 and -1 are fed into several linear projection layers and GeLU functions. We expect these GeLU functions and linear projection layers can learn to truncate excessive noise realizations to bootstrap the performance, similarly to the designs in [8, 120].

To introduce the position information to the decoding process, we add the same position embedding layer as in (5.5) to get the output $D_0 \in \mathbb{R}^{l \times d}$ as : $D_0 = \text{Siam}(S_d) + p_e$, where $\text{Siam}(\cdot)$ denotes the operations of the Siamese layer, with parameters illustrated in Fig. 5.3, and $p_e \in \mathbb{R}^{l \times d}$ is the output of the position embedding operation.

Transformer layer: After the Siamese layer and positional embedding, D_0 is passed through

L_t transformer layers:

$$\mathbf{D}_l = \text{MSA}(\mathbf{D}_{l-1}) + \text{MLP}(\text{MSA}(\mathbf{D}_{l-1})), \quad (5.8)$$

where $\mathbf{D}_i \in \mathbb{R}^{l \times d}$ 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 (5.6).

Linear Projection: Given the output of the L_t -transformer layer $\mathbf{D}_{L_t}$, we apply a linear projection $\mathbf{W}_{out} \in \mathbb{R}^{d \times c}$, and then reshape the output into a matrix of size $\mathbb{R}^{h \times w \times 3}$ to reconstruct the input image as:

$$\hat{\mathbf{S}} = \text{reshape}(\mathbf{D}_{L_t} \mathbf{W}_{out}). \quad (5.9)$$

5.3.3 Loss function

The encoder and decoder are optimized jointly by minimizing the mean square error between the input image $\mathbf{S}$ and its reconstruction $\hat{\mathbf{S}}$ as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}[\text{MSE}(\mathbf{x}, \hat{\mathbf{x}})] \triangleq \mathbb{E}[\|\mathbf{S} - \hat{\mathbf{S}}\|_2^2], \quad (5.10)$$

where the expectation is taken over the randomness both in the source pixels and the channel state. The aim is to find the parameters $(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$ that minimize the loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$.

5.4 Training and evaluation

This section presents experimental results to evaluate the performance of our JSCCformer-f model across various scenarios. DeepJSCC-f [35] and the bounds on the conventional separation-based digital schemes, assuming capacity-achieving channel codes, are employed as benchmarks. In the conventional separation-based digital schemes, we use Better Portable Graphics (BPG) as the compression method and assume capacity-achieving channel codes, denoted as the BPG-Capacity scheme. It is noteworthy that the BPG-Capacity scheme is practically unattainable and requires perfect CSI at both the transmitter and receiver ends.

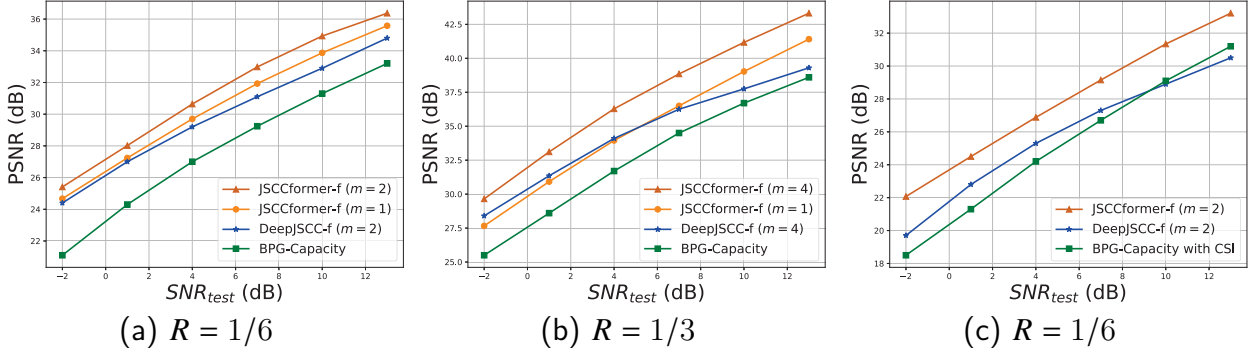


Figure 5.4: Performance of different schemes at various SNR values and bandwidth ratios with noiseless feedback, where models in subfigures (a) and (b) undergo assessment within the AWGN channel, and models in subfigure (c) are evaluated in a Rayleigh Fading channel.

Unless stated otherwise, all experiments in this section were performed on the CIFAR10 dataset[105], consisting of a training dataset of 50000 images and a test dataset of 10000 images, with the shape of $3 \times 32 \times 32$ (color, height, width). To achieve the optimal result, models of JSCCformer-f and DeepJSCC-f are all trained at a fixed average channel SNR value and tested at the same SNR value. All models were implemented in Pytorch with two GTX 3090Ti GPUs and optimized by an Adam optimizer. We use a learning rate of 0.00005 and a batch size of 128. Models were trained until the performance of a validation set stopped improving. Considering the model complexity and experiment performance, we set the $p = 8$, $l = 64$, and $c = 48$ for image vectorization. We set $d = 256$, $L_t = 8$, and $N_s = 8$ for each transformer layer of the ViT.

5.4.1 Transmission performance

General performance Considering AWGN channels, Figs. 5.4a and 5.4b present the PSNR performance versus different forward channel SNR values from -2 dB to 13 dB, where the bandwidth ratio is set to $R = 1/6$ and $R = 1/3$, respectively. We assume the feedback link is noiseless for now, and the noisy feedback case will be considered later in Section 5.4.1.

We first evaluate JSCCformer-f without channel feedback, i.e., $m = 1$. As shown in Figs. 5.4a and 5.4b, JSCCformer-f with $m = 1$ significantly outperforms the BPG-Capacity scheme in all SNRs and bandwidth ratios. We note again that the BPG-Capacity performance is

not achievable in the short block length regime. We next compare JSCCformer-f ($m = 1$) with DeepJSCC-f ($m = 2$). As shown in Fig. 5.4a, JSCCformer-f ($m = 1$) can surprisingly outperform DeepJSCC-f ($m = 2$) (up to 0.97dB) at all SNR values, even though DeepJSCC-f utilizes additional feedback from the receiver.

Then, we consider JSCCformer-f with $m > 1$. For a fair comparison with the simulation results in [35], we use the same number of blocks for JSCCformer-f. Specifically, we set $m = 2$ for $R = 1/6$ in Fig. 5.4a and $m = 4$ for $R = 1/3$ in Fig. 5.4b. JSCCformer-f ($m > 1$) improves the performance of JSCCformer-f ($m = 1$) in all SNR values (up to 1.06 dB for $m = 2$ and 2.11 dB for $m = 4$). Further, it can be observed that a larger bandwidth ratio leads to a larger gain. We also observe that with the same number of blocks, JSCCformer-f significantly outperforms DeepJSCC-f at all test SNRs. The gains are up to 2.03 dB and 3.88 dB for $R = 1/6$ and $R = 1/3$, respectively, demonstrating that JSCCformer-f can exploit feedback better than DeepJSCC-f. Moreover, with the increase in the bandwidth ratio, the improvement from JSCCformer-f becomes even more significant.

Fig. 5.4c shows the performance of JSCCformer-f and DeepJSCC-f in fading channels, where the fading gain is sampled from $h \sim \mathcal{CN}(0, 1)$. We observe that DeepJSCC-f outperforms the BPG-Capacity scheme only in the low SNR regime (-2dB to 10dB), although we should remind that this is a highly favorable bound for the separation-based scheme. However, JSCCformer-f consistently outperforms both DeepJSCC-f and BPG-Capacity schemes in all SNR regimes with a significant gain of up to 2.71dB and 3.57dB, respectively.

Impacts of bandwidth ratio and block number We evaluate the performance of JSCCformer-f over a wide range of bandwidth ratios R in Fig. 5.5a, where the forward SNR is fixed to 10dB, the feedback is noiseless, and $m = 4$. JSCCformer-f outperforms DeepJSCC-f in all bandwidth ratio settings with a gain of up to 3.42 dB. Interestingly, even with no feedback, JSCCformer-f ($m = 1$) outperforms DeepJSCC-f ($m = 4$) in the low bandwidth ratio regime ($R \leq 0.45$).

Next, we assess the impact of the block number on the PSNR of DeepJSCC-f and JSCCformer-f. To this end, we consider different feedforward SNR values (-2dB , 4dB , and 10dB), noiseless

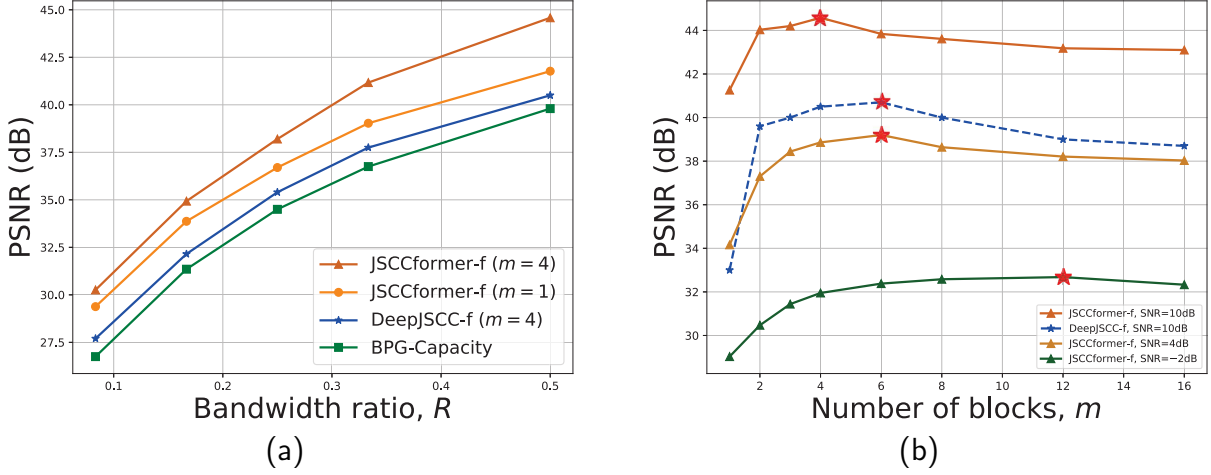


Figure 5.5: (a) Performance comparison of different models versus bandwidth ratio R in AWGN channel when SNR= 10 dB with noiseless feedback. (b) Performance of models over different block numbers and SNRs in AWGN channel with $R = 1/2$, where the optimal block number is highlighted with a red star.

feedback, $R = 1/2$, and vary the number of blocks m , from 1 to 16. We note that, in principle, increasing m means that we benefit from more feedback at the encoder. However, it can be observed in Fig. 5.5b that the performances of JSCCformer-f and DeepJSCC-f do not always improve with m . Specifically, increasing m brings some improvements at first for both DeepJSCC-f and JSCCformer-f; but the PSNR declines beyond a certain threshold. One possible explanation for this is the increasing learning difficulty brought about by increasing m . We can observe that the optimal m values for JSCCformer-f are $m = 4, 6, 12$ corresponding to $SNR = 10\text{dB}, 4\text{dB}, -2\text{dB}$, respectively. Notably, the optimal number of blocks tends to be larger under worse channel conditions. This can be explained by the fact that a greater number of feedback blocks can provide more information about the decoder's current beliefs and the channel conditions, which is more beneficial for the model in poor channel conditions.

We also remark that DeepJSCC-f adopts multiple encoders and decoders as different refinement layers; hence, unlike JSCCformer-f, DeepJSCC-f model complexity increases significantly with m . In contrast, JSCCformer-f employs a unified pair of encoder and decoder, yielding less complexity while achieving better performance. Interestingly, both DeepJSCC-f and JSCCformer-f tend to converge towards a sub-optimal performance as m becomes sufficiently large, thus revealing the existence of the trade-off between the training complexity and model performance.

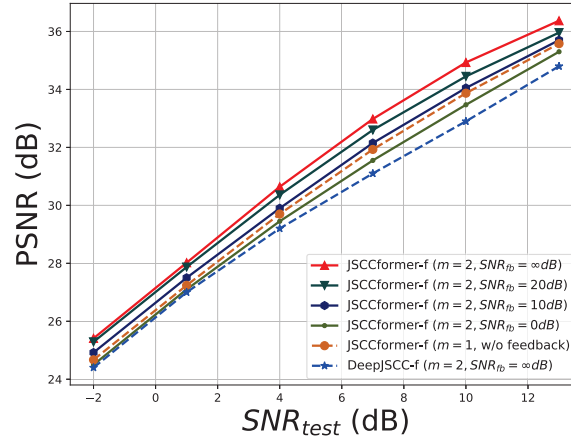


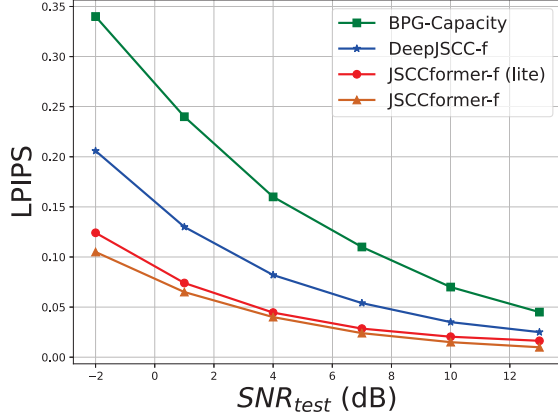
Figure 5.6: Performance of JSCCformer-f over noisy feedback link in AWGN channel, where $R = 1/6$, $m = 2$.

Our experimental results suggest that $m = 4$ provides a reasonable performance and complexity trade-off for JSCCformer-f in this channel condition, which can achieve competitive performance with an acceptable computation complexity.

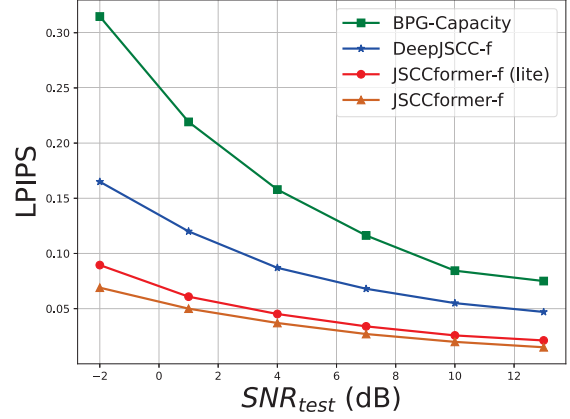
Noisy feedback channel In the above experiments, we have assumed a noiseless feedback channel. In this subsection, we investigate the impact of noisy feedback links for JSCCformer-f. Introducing noise in the feedback channel can lead to a drastic performance decrease [145] or even system breaking down [146], as feedback can be dominated by noise. DeepJSCC-f is the first practical image transmission scheme that considers noisy feedback, which we shall use as a benchmark.

We repeat the experiments in Fig 5.4a by replacing the noiseless feedback channel with an AWGN channel. The feedback channel SNR is set to $\text{SNR}_{\text{fb}} = 0\text{dB}$, 10dB , and 20dB . Fig. 5.6 shows the performance of the JSCCformer-f model with different SNR_{fb} values. When $\text{SNR}_{\text{fb}} = 20\text{dB}$ (high-quality feedback), the performance only degrades slightly (for at most 0.48dB), when compared with the perfect feedback case. When the feedback channel quality gets worse, e.g., $\text{SNR}_{\text{fb}} = 10\text{dB}$, the PSNR continues to decrease, but the gap is less than 0.88dB compared with the noiseless case). This verifies the robustness of JSCCformer-f to feedback channel noise. In particular, it can be observed that the performances of JSCCformer-f ($m = 2$) with $\text{SNR}_{\text{fb}} = 20\text{dB}$ and $\text{SNR}_{\text{fb}} = 10\text{dB}$ are still better than JSCCformer-f ($m = 1$) without

feedback, indicating that JSCCformer-f can still make good use of the noisy feedback. Even with larger feedback noise $\text{SNR}_{\text{fb}} = 0\text{dB}$, our model can still outperform DeepJSCC-f with a perfect feedback link. It again demonstrates the superiority of our scheme in noisy feedback scenarios. Overall, we conclude that JSCCformer-f is robust to feedback noise.



(a) $R = 1/6$ for CIFAR10 dataset.



(b) $R = 1/12$ for CelebA dataset.

Figure 5.7: LPIPS performance of different schemes, where models are trained with $m = 2$.

Perceptual quality evaluation To conduct a more comprehensive assessment of the model's perceptual qualities, we extend our evaluation with LPIPS measurements. To enhance the optimization of the LPIPS metric, akin to prior works [125], we introduce an additional LPIPS term into the loss function, formulated as follows: $\mathcal{L}(\theta, \phi) = \mathbb{E}[\|\mathbf{S} - \hat{\mathbf{S}}\|_2^2] + \lambda \cdot \text{LPIPS}(\mathbf{S}, \hat{\mathbf{S}})$, where λ is set as 0.1 in our simulations.

As shown in Fig. 5.7, JSCCformer-f achieves the best perceptual performance across a spectrum of channel conditions, bandwidth ratios, and datasets. In particular, in poor SNR conditions, JSCCformer-f exhibits a substantial performance advantage over DeepJSCC-f and BPG-Capacity scheme, with a maximum LPIPS discrepancy of 0.235 and 0.245 at SNR -2dB for CIFAR10 and CelebA, separately. A similar trend is observed with the JSCCformer-f (lite), which preserves adaptability to channel variations but suffers a slight performance degradation relative to the original JSCCformer-f model.

Table 5.1: Performance of models over the Kodak dataset.

Different models	-2 dB	1 dB	4 dB	7 dB
JSCCformer-f	29.20 dB	30.84 dB	32.05 dB	33.16 dB
JSCCformer-f (lite)	29.16 dB	30.76 dB	31.90 dB	32.91 dB
DeepJSCC-f	28.02 dB	29.57 dB	31.04 dB	32.20 dB

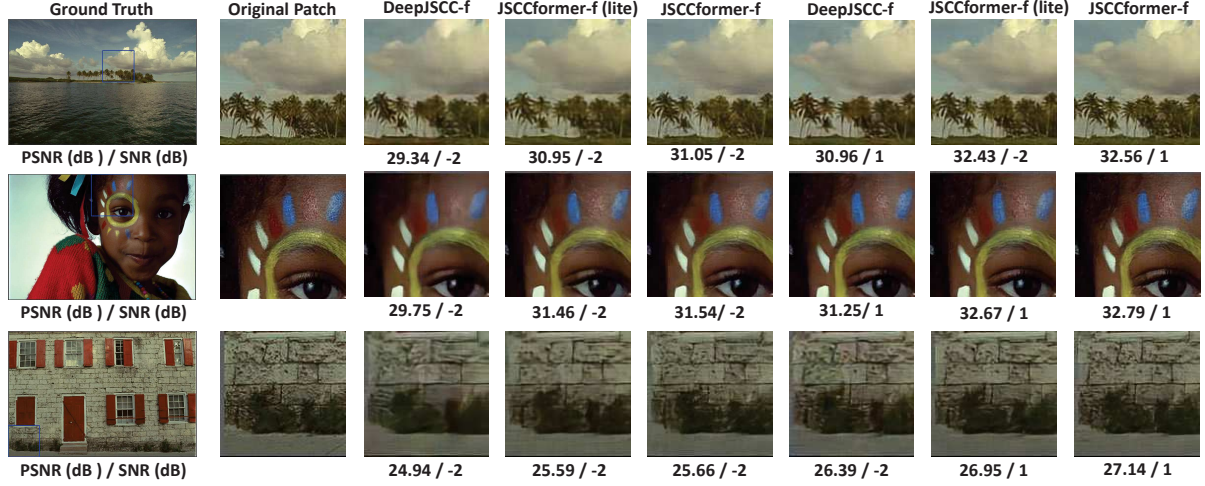


Figure 5.8: Visual comparisons of images transmitted by DeepJSCC-f, JSCCformer-f (lite), and JSCCformer-f over the AWGN channel, where the model is trained on the ImageNet dataset and validated on the Kodak dataset at various SNR values and $R = 1/12$.

High resolution dataset and visualization In order to evaluate the model generalizability, we validate the JSCCformer-f and JSCCformer-f (lite) with $m = 2$ on the Kodak dataset using $SNR_{test} \in [1, 10]$ dB and $R = 1/12$, utilizing CPE method. Specifically, models are trained with randomly cropped 256×256 patches from the ImageNet dataset and subsequently evaluated on the Kodak dataset.

The performances of JSCCformer-f, trained at specific SNRs, and JSCCformer-f (lite), trained with random SNRs, are compared against DeepJSCC-f over the AWGN channel, and the results are presented in Table 5.1, where $R = 1/12$, $m = 2$. In high-resolution datasets, JSCCformer-f based schemes maintain state-of-the-art performance and channel adaptability across various SNRs. Moreover, a sufficiently large dataset (such as ImageNet) allows our model to perform well on previously unseen images across a wide range of channel conditions (SNRs).

We visualize the comparisons of the recovered image patches by DeepJSCC-f, JSCCformer-f (lite) and JSCCformer-f in Fig. 5.8. We can observe that JSCCformer-f performs the best with more detailed high-frequency features (e.g., hair and trees), particularly in the low SNR

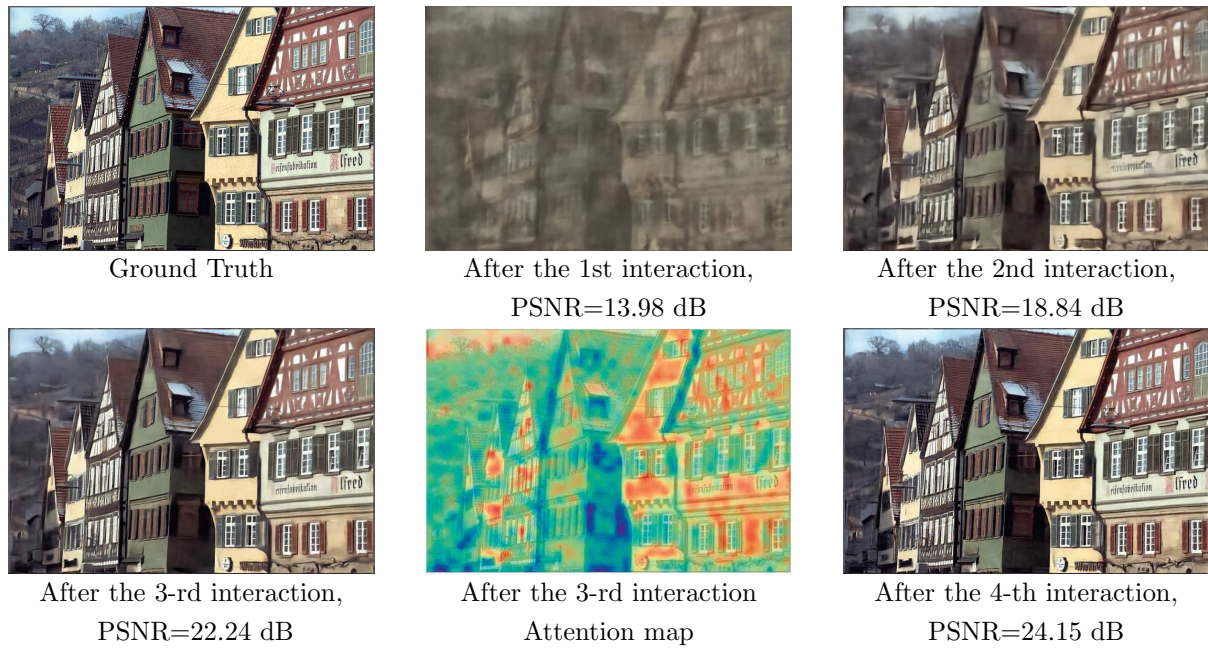


Figure 5.9: Visualization of the reconstructed image with JSCCformer-f after each interaction for an input image from the Kodak dataset, where the model is trained on the ImageNet dataset with $R = 1/12$, and $SNR = -2$ dB.

regime, e.g., at -2 and 1 dB. JSCCformer-f (lite) demonstrates comparable visual performance to JSCCformer-f and exhibits a noteworthy improvement over DeepJSCC-f.

Additionally, the effect of each interaction and the attention of the JSCCformer-f model in each transmission block are presented in Fig. 5.9 and 5.10, where we trained the models from the ImageNet dataset and tested the models over the Kodak and the CelebA datasets. We can observe that the reconstruction performance is significantly improved after each interaction, particularly in some high-frequency details. Interestingly, the model tends to acquire an initial understanding of the overall semantic structure after the initial interaction, subsequently recovering a rough sketch. It then proceeds to enhance and refine its prior reconstructions through successive interactions. Furthermore, the attention map of the model in the final interaction revealed that our model could concentrate on some blurry areas and refine them in the next block, demonstrating how our method refines results in a coarse-to-fine manner based on the attention mechanism.

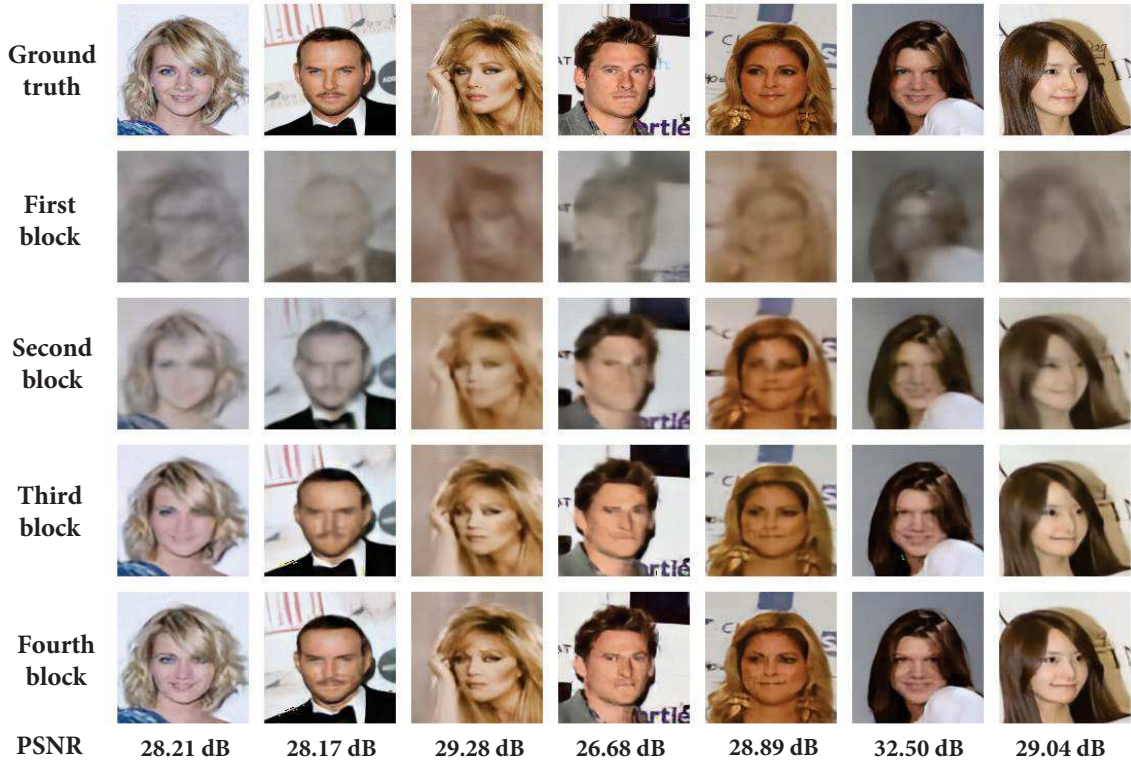


Figure 5.10: Visualization of the reconstructed image with JSCCformer-f after each interaction for images from the CelebA dataset, where the model is trained on the ImageNet dataset with $R = 1/12$, and $SNR = -2$ dB.

5.4.2 Adaptability

This section validates the SNR-adaptability of JSCCformer-f. To verify the SNR adaptability, we adopt a random-SNR training method. That is, the SNRs in the training phase are uniformly sampled from the range $[-2, 15]$ dB. After training, the well-trained model is then evaluated at different test SNRs. We expect the JSCCformer-f and JSCCformer-f (lite) encoder to learn the channel conditions from the feedback signal to help the model map the source signal into channel symbols via the SA mechanism.

Table 5.2: Ablation study over different training strategies (fixed SNR or random SNR), where $m = 4$ and $R = 1/3$.

Different channel SNR	1 dB	7 dB	13 dB
JSCCformer-f (fixed)	33.11 dB	38.85 dB	43.32 dB
JSCCformer-f (random)	32.69 dB	38.31 dB	41.47 dB
JSCCformer-f (lite) (fixed)	32.91 dB	38.64 dB	42.77 dB
JSCCformer-f (lite) (random)	32.89 dB	38.57 dB	41.87 dB

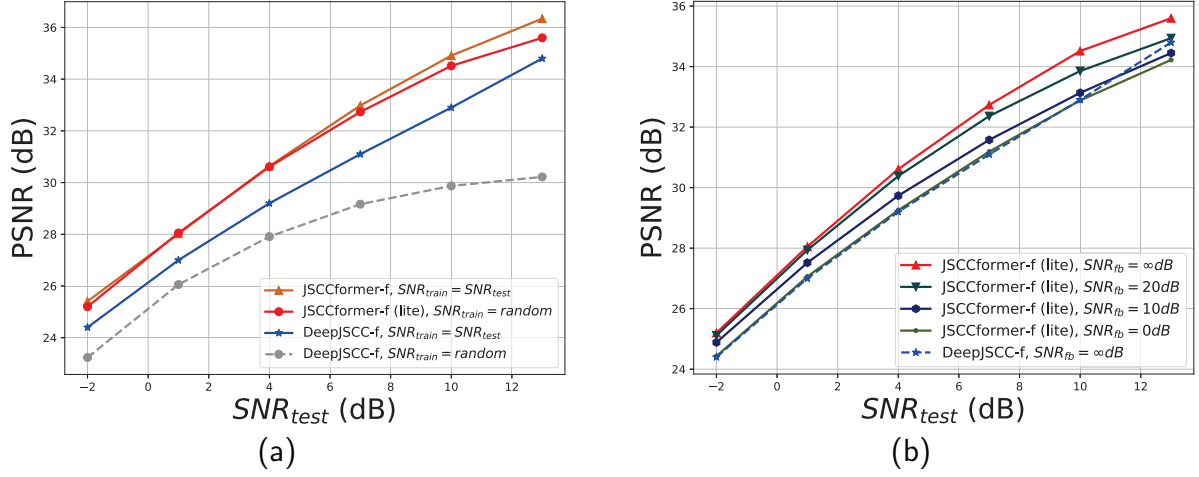


Figure 5.11: (a) Performance of JSCCformer-f (lite) compared with other schemes in AWGN channel, where $R = 1/6$, and $m = 2$. (b) Performance of JSCCformer-f (lite) trained with random SNRs over noisy feedback link in the AWGN channel, where $R = 1/6$ and $m = 2$.

We first conduct an ablation study over different training strategies for JSCCformer-f and JSCCformer-f (lite) in Table 5.2, where $m = 4$ and $R = 1/3$. When trained for a specific SNR, JSCCformer-f achieves superior performance, tailoring model parameters for this channel condition. On the other hand, when trained over random channel SNRs, both models experience performance degradation. However, JSCCformer-f (lite) exhibits a reduced level of performance degradation due to its simplified architecture, particularly the lack of a decoder module at the transmitter, which makes it difficult to adapt to varying channel conditions. These results highlight that JSCCformer-f (lite) is tailored for varying channel conditions, and we set the model trained for varying channel SNRs as the default JSCCformer-f (lite) model.

To compare with the benchmark, we repeat the experiments in Fig. 5.4 and plot the performance of JSCCformer-f, JSCCformer-f (lite) and DeepJSCC-f with different training strategies in Fig. 5.11a. It can be observed that the DeepJSCC-f scheme exhibits a significant degradation in performance when trained with the random SNR strategy, indicating its inability to adapt to diverse channel conditions. Then, we compare the performance of the JSCCformer-f (lite) model trained using random SNRs and the optimal DeepJSCC-f models trained at each individual SNR. As shown, JSCCformer-f (lite) outperforms DeepJSCC-f across all SNR scenarios, with a maximum gain of 1.64 dB. JSCCformer-f (lite) can adapt to different test SNRs. Importantly, this channel adaptability is attributed to the proposed method's inherent attention

mechanism rather than being solely an outcome of the random training strategy.

In a comparative analysis between JSCCformer-f (lite) and JSCCformer-f from Fig. 5.11a, the results indicate that the performance of JSCCformer-f (lite) is only slightly worse than the optimal performance of JSCCformer-f at all SNRs. The performance gap is smaller than 0.74dB. In summary, our JSCCformer-f (lite) is able to achieve close performance to JSCCformer-f in a wide range of SNRs and is significantly better than DeepJSCC-f. It is worth noting that we do not feed the channel SNRs explicitly to the encoder and decoder. Instead, they are implicitly learned by JSCCformer-f from the observed channel outputs.

We also evaluate the robustness of JSCCformer-f (lite) to the noisy feedback case while maintaining channel adaptability. We consider JSCCformer-f (lite) with noisy AWGN feedback links with feedback SNRs of 0 dB, 10 dB, and 20 dB, whose performance is illustrated in Fig. 5.11b. When $\text{SNR}_{\text{fb}} = 20$ dB, the performance is only slightly worse (the gap is smaller than 0.6 dB) compared to that with a noiseless feedback channel. When $\text{SNR}_{\text{fb}} = 10$ dB and $\text{SNR}_{\text{fb}} = 0$ dB, the performance continues to degrade; the gap from the noiseless feedback case is up to 1.1dB and 1.3 dB, respectively. Compared to DeepJSCC-f with ideal feedback, JSCCformer-f (lite) exhibits superior performance even when the feedback channel SNR is 0 dB. This observation confirms the superiority of our JSCCformer-f (lite) model and its robustness to noisy feedback.

To better understand the effect of the Siamese layer, we conduct an additional ablation study over the Siamese layer, where JSCCformer-f and JSCCformer-f (lite) models are trained over the CIFAR10 dataset with $R = 1/6$ and $m = 2$. The results are presented in Table 5.3, which demonstrates that similar to [8, 120], the incorporation of a Siamese layer enhances the transmission performance.

Table 5.3: Ablation study of the Siamese layer on the CIFAR10 dataset, $R = 1/6$, $m = 2$.

Channel SNR	1dB	7dB	13dB
JSCCformer-f	28.06dB	32.98dB	36.37dB
JSCCformer-f w/o Siamese	27.45dB	32.53dB	36.01dB
JSCCformer-f (lite)	28.05dB	32.73dB	35.60dB
JSCCformer-f (lite) w/o Siamese	27.52dB	32.43dB	35.35dB

5.4.3 Model efficiency

This section evaluates the model efficiency of the different schemes. Table 5.4 presents the model size, floating point of operations (FLOPs), and the coding time for DeepJSCC-f, JSCCformer-f, and JSCCformer-f (lite) with different number of blocks on the CIFAR10 dataset (with a batch size of 1 and $R = 1/2$).

Table 5.4: The number of parameters (millions), FLOPs (G), and coding time (ms) for different models; bold figures correspond to the minimal values among different models for each m value.

Block number m		1	2	3	4	6	8	12
Parameters	DeepJSCC-f	10.58	25.4	40.25	55.21	85.10	114.99	174.76
	JSCCformer-f	12.93	12.95	12.96	12.97	12.99	13.02	13.07
	JSCCformer-f (lite)	12.93	12.93	12.93	12.93	12.93	12.93	12.93
FLOPs	DeepJSCC-f	1.06	2.82	4.58	6.34	9.86	13.38	20.42
	JSCCformer-f	0.83	1.67	2.51	3.35	5.03	6.72	10.12
	JSCCformer-f (lite)	0.83	1.24	1.64	2.05	2.86	3.66	5.28
Encoding time (ms)	DeepJSCC-f	1.82	6.12	10.03	14.10	21.99	32.07	45.94
	JSCCformer-f	4.75	13.76	23.20	32.64	50.22	71.54	111.02
	JSCCformer-f (lite)	4.54	9.04	13.72	18.20	27.64	36.83	58.82
Decoding time (ms)	DeepJSCC-f	1.84	4.32	6.13	8.21	12.15	17.40	24.12
	JSCCformer-f	4.68	5.44	4.89	5.36	4.96	3.78	4.46
	JSCCformer-f (lite)	4.97	4.12	5.00	4.73	4.79	3.98	4.15
Total time (ms)	DeepJSCC-f	3.66	10.44	16.16	22.31	34.14	49.47	70.06
	JSCCformer-f	9.43	19.02	28.09	38.00	55.81	75.32	115.48
	JSCCformer-f (lite)	8.51	13.16	18.72	22.93	32.43	40.81	62.97

As shown in the table, the parameter count of DeepJSCC-f rapidly increases as the block number increases, leading to a more complex model. Conversely, JSCCformer-f and JSCCformer-f (lite) maintain a lower parameter count, even as m increases ($m > 6$). Notably, JSCCformer-f (lite) exhibits the lowest parameter count when $m > 1$. Interestingly, when $m = 12$, JSCCformer-f (lite) achieves improved performance while reducing memory consumption by up to 92.6% compared to DeepJSCC-f. In terms of computational complexity, it becomes evident that DeepJSCC-f incurs greater computational costs (FLOPs), particularly as the number of blocks increases. JSCCformer-f (lite) requires up to 3.8x times fewer FLOPs while maintaining competitive performance, indicating superior efficiency. Our JSCCformer-f-based methods demand fewer computations to achieve optimal performance, implying significantly lower latency in

practical use.

While FLOPs measure theoretical complexity, actual inference time also depends on factors like memory access, I/O bottlenecks, hardware parallelism, and software-level optimizations. For instance, models with fewer FLOPs but irregular structures (e.g., sparse or highly modular architectures) may result in less efficient execution due to inefficient cache usage or kernel overheads. To bring a more comprehensive understanding of the model’s computation cost, we presented the encoding and decoding time of different methods with a GPU of RTX A6000 and a CPU of Intel Xeon Gold 5220R in Table 5.4. The coding speed of the DeepJSCC-f is the fastest, coming from the well-optimized CNNs inference library. For the JSCCformer-f, it is feasible to maintain the average coding time at an acceptable range within 51ms when $m < 6$. Compared with JSCCformer-f, JSCCformer-f (lite) significantly reduces the coding time, with potential savings of up to 47%. Regarding the decoding time, it is noteworthy that the decoding operation for JSCCformer-based models is executed just once, with an average decoding time within 5 ms, indicating a rapid decoding speed at the receiver. Conversely, the DeepJSCC-f model must conduct multiple decoding operations at each block, resulting in high receiver-side latency, especially when m increases.

In summary, our proposed method demonstrates both practicality and efficiency in terms of the model size and computation complexity. It is worth highlighting that further optimization exists in our models’ inference speed, encompassing alternative hardware options, different implementations, and acceleration techniques for ViT.

5.4.4 Variable Rate Transmission

The JSCC coding problem with channel feedback can be reformulated by imposing a particular transmission quality objective and minimizing the corresponding channel bandwidth, where more significant gains can be observed when considering variable-length coding [35, 78]. As previously emphasized, the design of JSCCformer-f, which maps the channel symbols considering the decoder’s present belief of the transmission, is inherently suitable for variable rate

transmission. Specifically, the transmitter can determine the stopping time by analyzing the decoder's current knowledge of the transmission with the help of perfect channel output feedback. The transmitter stops the transmission of the remaining blocks when the current performance meets the desired target.

We evaluate the performance of JSCCformer-f with variable transmission rate, where we compute the average bandwidth necessary to achieve a predefined PSNR target. For a fair comparison, we set $m = 8$, same with DeepJSCC-f. In the BPG-Capacity scheme, we compute the number of channel uses required to transmit the minimum number of compressed bits while satisfying the target PSNR value using a capacity-achieving channel code. In order to improve the efficacy of variable rate transmission, we fine-tune the well-trained JSCCformer-f while preserving the end-to-end transmission performance, which is achieved through the utilization of the subsequent loss function: $\mathcal{L}(\theta, \phi) = \sum_{i=1}^8 \|\mathbf{S} - \hat{\mathbf{S}}_i\|_2^2$, where $\hat{\mathbf{S}}_i \triangleq D_\phi(\hat{\mathbf{Y}})$ is the intermediate reconstruction of the source signal at the transmitter after each block. This additional loss term can guarantee the intermediate transmission quality while incurring a trade-off with the final performance, where the weight assigned to each term can regulate this trade-off.

The cumulative distribution function of the required bandwidth to achieve two distinct PSNR values (25 dB and 30 dB) is presented in Fig. 5.12a. We can observe that JSCCformer-f provides significant improvements compared to DeepJSCC-f and BPG-Capacity across the entire range, particularly when the target PSNR is higher. Fig. 5.12b displays the average bandwidth ratio needed to achieve distinct target PSNR values. The observed significant improvements compared to the digital scheme confirm the theoretical results in this practical scenario. Furthermore, the JSCCformer-f model exhibits superior performance and establishes a new state-of-the-art performance, particularly in the high target PSNR value regimes. We can conclude that the JSCCformer-f maintains its superiority in the variable rate transmission scenarios and achieves state-of-the-art performance.

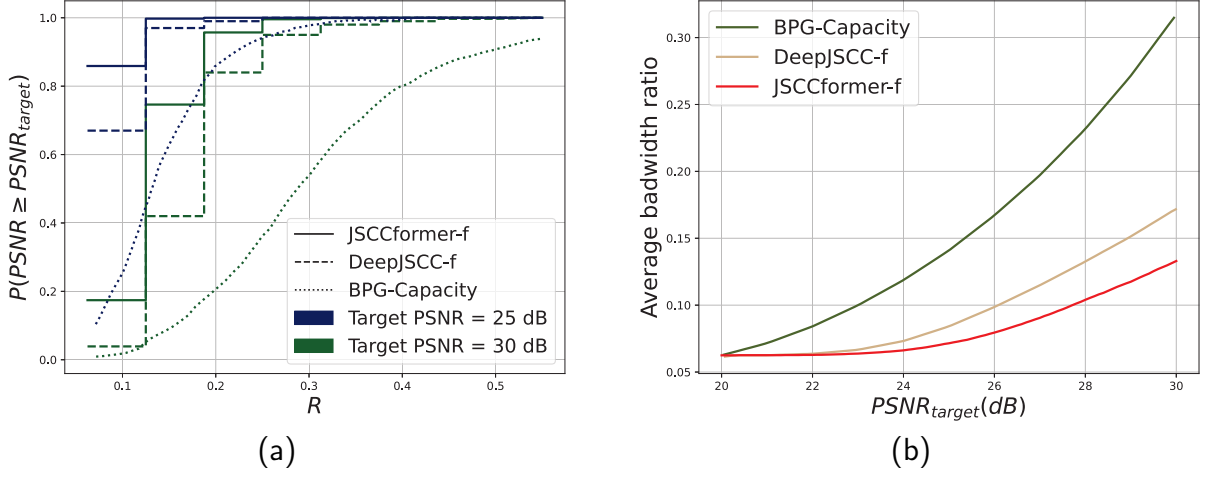


Figure 5.12: Target PSNR analysis of different schemes for variable rate transmission. (a) The cumulative distribution function over various bandwidths given a specific target PSNR. (b) The average bandwidth ratio to achieve a specific target PSNR.

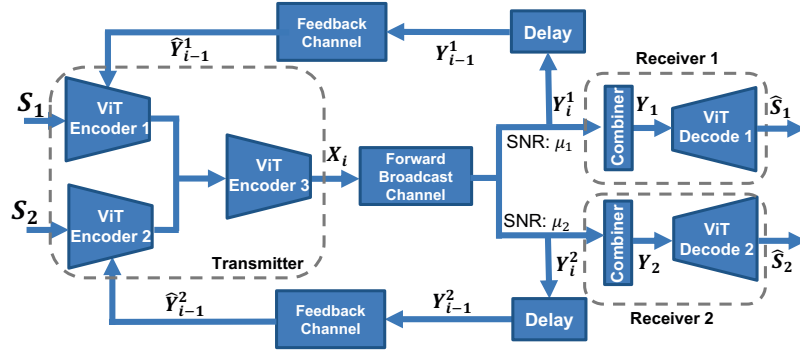


Figure 5.13: Schematic illustration of extending our JSCCformer-f scheme into broadcast channels.

5.4.5 JSCCformer-f for broadcast channels

This section verifies the generalizability of the proposed JSCCformer-f architecture by extending it to the broadcast channel. In the broadcast channel model, as shown in Fig. 5.13, one transmitter aims to deliver two images S_1, S_2 to two receivers, respectively. The transmitter first encodes the two inputs to a block of symbols X_i and then transmits X_i through the broadcast channel. The SNRs associated with the two receivers are μ_1 and μ_2 . We denote the received signal at the two receivers by Y_i^1 and Y_i^2 , respectively. Subsequently, Y_i^1 and Y_i^2 are fed back to the transmitter for the encoding of the next block, X_{i+1} , and so on. The feedback is assumed to be noiseless in this section.

In contrast to the previous point-to-point scheme, the transmitter in this scheme employs a

broadcast code to map the signal source based on the feedback signals from both receivers. To optimize the network efficiency for multi-receiver scenarios, we adopt two compact ViT-encoders ($L_t = 4, N_s = 4, d = 256$) per message, following Algorithm 3, and an additional ViT-encoder ($L_t = 4, N_s = 4, d = 256$) with the same structure to combine the feature maps of the two source signals. At each receiver, a ViT-decoder ($L_t = 4, N_s = 4, d = 256$) is deployed to reconstruct the received images $\hat{\mathbf{S}}_i$ following Algorithm 4. The PSNR of receiver i is denoted by $\text{PSNR}_i, i = 1, 2$. The loss function is given by: $\mathcal{L}_2 = \lambda \text{MSE}(\mathbf{S}_1, \hat{\mathbf{S}}_1) + (1 - \lambda) \text{MSE}(\mathbf{S}_2, \hat{\mathbf{S}}_2)$, where parameter $\lambda \in [0, 1]$ balances the performances of the two receivers.

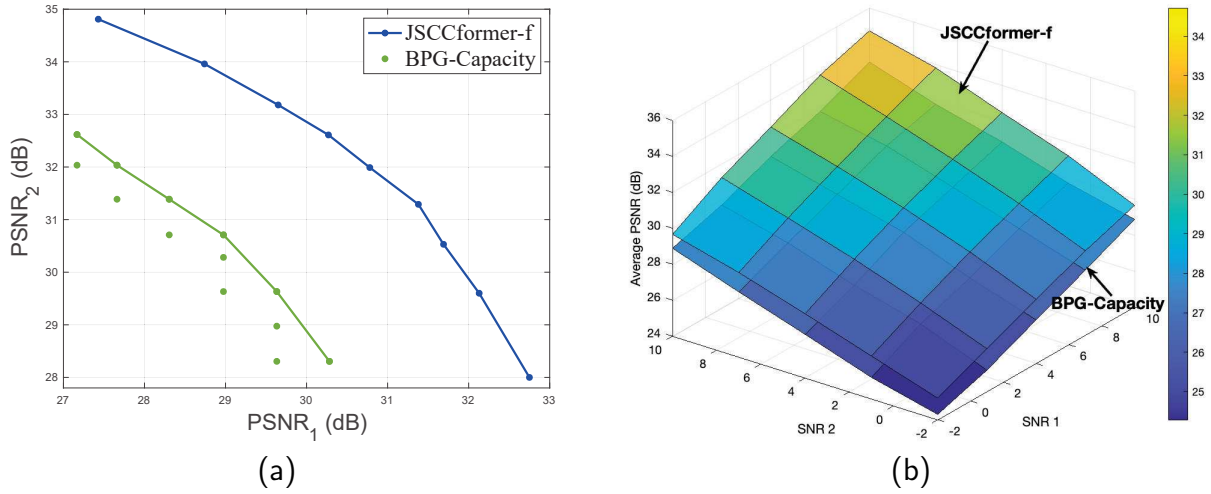


Figure 5.14: (a) PSNR performance region from different schemes, where $R = 1/6$, $\mu_1 = 4$ dB and $\mu_2 = 7$ dB. (b) Average PSNR performance of JSCCformer-f (broadcast) for different channel SNR pairs.

As the benchmark, we again consider the digital scheme ‘BPG-Capacity’, which utilizes BPG as the source coding scheme and assumes capacity-achieving channel codes of the broadcast channel with feedback [147, 148]. In particular, the achievable rate region lies within the intersection of the curves parameterized by $\alpha \in [0, 1]$ as:

$$R_1 = \frac{1}{2} \left(1 + \frac{\alpha P_s}{\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \right), R_2 = \frac{1}{2} \left(1 + \frac{(1-\alpha) P_s}{\alpha P_s + \sigma_2^2} \right), \quad (5.11)$$

and

$$R_1 = \frac{1}{2} \left(1 + \frac{(1-\alpha) P_s}{\alpha P_s + \sigma_1^2} \right), R_2 = \frac{1}{2} \left(1 + \frac{\alpha P_s}{\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \right), \quad (5.12)$$

where R_1 and R_2 are the achievable rates, σ_1^2 , σ_2^2 are the corresponding noise power of each

channel, and P_s is the power constraint. We first compare the achievable PSNR region of the JSCCformer-f and the BGP-Capacity method in Fig. 5.14a, where we consider $\mu_1 = 4, \mu_2 = 7$ (dB) and $R = 1/6$. For BPG-Capacity, we plot each PSNR performance pair $(\text{PSNR}_1, \text{PSNR}_2)$, considering the convex hull of the capacity region. For JSCCformer-f, we plot the performance pair from each model trained with different $\lambda \in [0.05, 0.95]$. Significant improvements can be observed from our JSCCformer-f-broadcast scheme (at least 3dB for each receiver).

We then consider the average PSNR $(\frac{\text{PSNR}_1 + \text{PSNR}_2}{2})$ of JSCCformer-f and BGP-Capacity as a function of various SNR pairs $\mu_1, \mu_2 \in [-2, 10]$ dB in Fig. 5.14b, where we set $\lambda = 0.5$ and $R = 1/6$. We can observe that JSCCformer-f outperforms BPG-Capacity in all the SNR pairs. The average PSNR gains are at least 0.9 dB. In conclusion, the proposed JSCCformer-f framework can be easily generalized to broadcast channels and achieves significantly better results than the conventional separation approach.

5.5 Conclusion

We presented a new feedback-aided wireless image transmission paradigm, dubbed JSCCformer-f. JSCCformer-f addresses the four key problems of existing image transmission methods with channel feedback, i.e., high complexity, inadaptability, suboptimality, and non-generalizability. First, JSCCformer-f generates coded symbols using a unified encoder by exploiting the semantics from the source image and the feedback signal from the receiver (i.e., the decoder's current belief about the source image), and is computationally efficient. Second, JSCCformer-f shows channel adaptability over a wide range of SNRs without the need for a separate channel estimation module. Third, JSCCformer-f sets a new state-of-the-art and significantly improves the transmission quality in all the SNRs and bandwidth ratio values considered in this chapter, compared with DeepJSCC-f and traditional digital schemes. Finally, we extended JSCCformer-f to broadcast channels and demonstrated that the transmitter can learn to adaptively map the inputs and the multiple feedback signals to the channel input, considering different channel conditions to satisfy multiple receivers.

5.5.1 Future work

Moving forward, JSCCformer-f can be extended in multiple directions:

- We have assumed that the receiver passively feeds back the received coded symbols and relies on the encoder to extract semantic information from the feedback. More generally, the receiver can actively code its feedback based on its knowledge about the source and can add protection against noise in the feedback channel [31].
- A major performance gain of JSCCformer-f arises from additional feedback overhead. This cost can be mitigated by designing a joint DeepJSCC scheme for the feedback link, reducing communication load while improving robustness to channel noise. The trade-off between feedback resources and performance gains can be further explored in future work from a more practical perspective.
- The encoder of JSCCformer-f utilizes the SA mechanism to process the received feedback. In principle, more advanced learning techniques, e.g., cross-attention mechanism and contrastive learning methods, can extract useful semantics from the feedback signals and generate coded symbols to refine the receiver's belief.
- Another interesting line is to consider a generative model available at the receiver [125], which may not be known to the transmitter. Feedback, in this case, needs to convey this additional knowledge to the encoder.

Chapter 6

DeepJSCC empowered edge inference for semantic communication

6.1 Introduction

Following the investigation of DeepJSCC for image transmission across various channel models in previous chapters, this chapter examines the application of DeepJSCC technologies in edge inference tasks. Recent advancements in distributed edge computing have intensified the need for efficient inference methods, particularly in the context of cooperative image retrieval tasks. These tasks involve multiple edge devices attempting to identify the same object within a gallery database, captured from various cameras at different angles, as illustrated in Fig. 6.1(a). This process, known as re-identification (ReID), presents a significant challenge due to the diverse perspectives and qualities of the images involved. The complexity of these tasks is compounded when data must be communicated between nodes to increase inference accuracy, leveraging limited communication resources across distributed devices [10, 57, 86].

Traditionally, communication systems aimed at supporting such distributed tasks encode the images at the application layer and convey the encoded features to the edge server using existing source coding, channel coding, and modulation techniques. However, the strict latency requirements inherent in many edge applications necessitate extremely short blocklength codes,

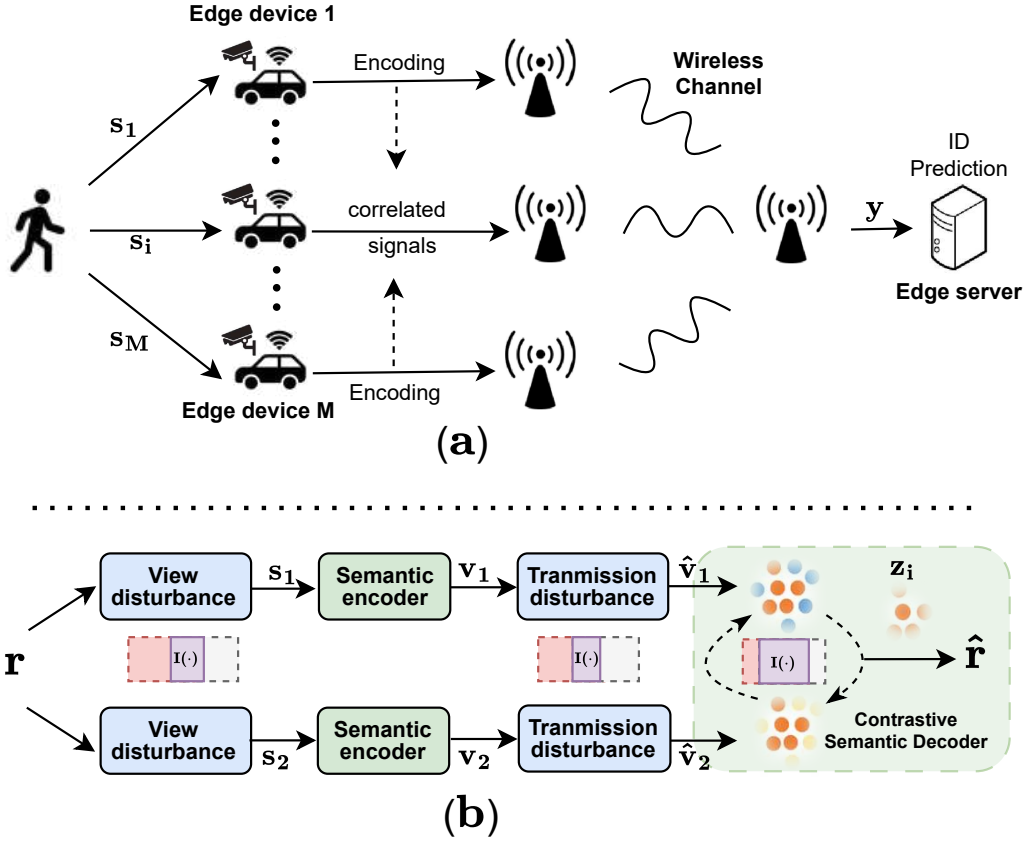


Figure 6.1: (a) Collaborative edge retrieval task. (b) CL for collaborative edge retrieval: the dotted and solid rectangles represent the information contained in each observation and the shared identity-related information between two observations, respectively; the common identity-related features are represented by the orange points, while the inconsistent features are represented by other colors. We expect CL module to help $\hat{v}_1$ and $\hat{v}_2$ to be more discriminative with maximum semantic consistency and identity-related features.

leading to heightened suboptimality in separate source-channel coding schemes as the code length decreases [11]. The suboptimality of separate source and channel coding employed in the digital transmission scheme can be attributed to two primary factors. First, the suboptimality of the conventional separation transmission scheme becomes apparent in the finite blocklength regime, as demonstrated in [57] within a point-to-point scenario. Second, even in scenarios with an infinite blocklength regime, the separation scheme becomes suboptimal when the transmitted sources over the MAC exhibit correlation [149, 150]. Existing literature establishes that harnessing the correlation between sources to generate correlated codewords at the transmitter can strictly enhance end-to-end performance [150, 151]. Facilitating partial cooperation among distributed transmitters necessitates allowing the transmitted signals to exhibit statistical dependence on the source outputs, thereby inducing correlation among the

transmitted signals. In contrast, separation-based schemes operate oppositely, where the dependence among sources is destroyed by separate source and channel coding operations, resulting in independent transmitted signals and sub-optimal performance.

To address these inefficiencies, there is a growing shift towards semantic communication systems in edge computing. These systems leverage DeepJSCC to prioritize the transmission of goal-oriented semantic content over raw data [4, 50, 54, 91, 152]. By focusing on the semantics and relevance of the information, semantic communication optimizes the process to transmit only the most critical semantic features. This strategic focus significantly reduces bandwidth and power consumption and enhances the accuracy of inference tasks by ensuring that essential information is effectively preserved during transmission. As a result, semantic communication offers a robust and efficient solution for managing the complexities of distributed image retrieval in edge environments. One notable example is the implementation of a deep learning-based JSCC wireless retrieval scheme, which has demonstrated superior performance compared to traditional separation-based digital approaches under various channel conditions [57].

Additionally, research on remote retrieval involving multiple users has led to the development of collaborative semantic communication schemes. These schemes improve inference accuracy by leveraging additional information from multiple views and utilizing DeepJSCC technologies [86]. Furthermore, the study explores both orthogonal multiple access (OMA) and non-orthogonal multiple access (NOMA) schemes. In particular, the NOMA scheme, where devices transmit over a shared channel, has been shown to provide substantial benefits by maintaining the correlations among the transmitted symbols, thus enhancing the overall system performance.

In addressing the multi-device edge retrieval problem, the inherently distributed nature introduces significant collaborative challenges. Edge devices must work together to extract pertinent semantic information, discarding irrelevant data to optimize inference accuracy within the limited bandwidth constraints. To this end, contrastive learning (CL) emerges as a powerful tool for developing discriminative representations of common semantics among correlated inputs. CL operates on the principle of enhancing feature discrimination by minimizing the distance between features of the same identity and maximizing the distance between features of differ-

ent identities. Well-known CL frameworks such as MOCO-V2 [153], SimCLR [154], and BYOL [155] have demonstrated their efficacy across various machine learning tasks, including object detection [156], semantic segmentation [157], and classification [158].

The essence of CL aligns closely with the goals of semantic communication, especially in edge computing environments. By focusing on the semantic aspects of data, CL in semantic communication systems [159, 160] facilitates more effective preprocessing and helps disentangle complex data patterns, significantly enhancing the quality of the information transmitted across networks. To leverage these advantages, we propose a CL-based collaborative semantic communication (CL-SC) scheme. This approach not only fine-tunes the semantic feature extraction but also tailors it to the specific requirements of edge inference tasks. The CL-SC scheme systematically optimizes the extracted features in a coarse-to-fine manner, capitalizing on the inherent correlations within the data to boost overall inference performance.

One key innovation of the CL-SC framework lies in our approach to interpreting the received noisy signals. We consider the received noisy signals with correlations from various JSCC encoders and channel realizations as distinct augmentations of the source signal in the presence of diverse disturbances. This design allows us to employ CL technologies to maximize inference accuracy by refining the received coarse semantic features. Specifically, we model the problem as a multi-terminal and noisy channel variant of the remote source compression problem [161]. As shown in Fig. 6.1(b), we model the source images $\mathbf{s}_1$ and $\mathbf{s}_2$ from different cameras as the distinct noisy versions of the desired identity $\mathbf{r}$ through stochastic channels $p(\mathbf{s}_1|\mathbf{r})$ and $p(\mathbf{s}_2|\mathbf{r})$, denoted as ‘view disturbance’, where the varying backgrounds and viewing angles can introduce extraneous information. Following previous pipelines [57, 86] with promising performance, we first encode the source signals into semantic features $\mathbf{v}_1$ and $\mathbf{v}_2$, which are transmitted over the multiple access channel (MAC) and then reconstructed as the coarse semantic features by a JSCC scheme, denoted as $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$. This process is modeled as the ‘transmission disturbance’ $p(\hat{\mathbf{v}}_1|\mathbf{v}_1)$ and $p(\hat{\mathbf{v}}_2|\mathbf{v}_2)$. Note that the transmission disturbance process is optimized to maximize the final reference accuracy instead of minimizing the transmission distortion. The reconstructed features are then input to a contrastive semantic decoder to output the inference identity $\hat{\mathbf{r}}$, where a cross-view CL method is proposed to refine the coarse semantic features by

maximizing the common identity-related information between different disturbed views.

Another key innovation is that, by combining the proposed cross-view CL mechanism with the “over-the-air” (OTA) technologies [162], the proposed CL-SC scheme transmits and aggregates the semantic features as analog values over the shared wireless channel, which can exploit the superposition property of the wireless medium and achieve the state-of-the-art (SOTA) performance. Specifically, we first consider the OMA scheme employing time division multiple access (TDMA), and compare our CL-SC with previous JSCC and digital schemes [57, 86]. Then, we investigate the proposed CL-SC scheme over an alternative NOMA transmission approach employing OTA technologies, which reveals additional noteworthy advantages across various distributed inference tasks [163, 164]. The integration of OTA technologies within the NOMA scheme is expected to optimize the inference performance, specifically those denoted as feature-over-the-air. The CL-SC framework leverages the contrastive learning mechanism to explicitly acquire the discriminative features of signals based on their correlations. This aligns synergistically with the inherent characteristics of OTA technologies, increasing the distance between different clusters. Consequently, this alignment enhances the retrieval performance within the system.

Our main contributions can be summarized as follows:

- We put forth a novel CL paradigm for edge inference and investigate the effect of the channel condition on the properties of semantic information transmitted by different edge devices. To the best of our knowledge, this is the first work to introduce contrastive learning technologies for edge inference in semantic communication systems.
- The proposed CL paradigm reveals how the correlations between the transmitted signals from different wireless devices can be optimized to improve wireless edge inference performance. This paradigm can be used in any scenario where multi-correlated signals exist and can be extended into more than two sources. Additionally, we apply OTA technologies to enhance the inference performance by exploiting the superposition property of transmitting the correlated signals.

- Numerical experiments verify that our method improves upon the SOTA edge inference performance in all the considered channel conditions and bandwidths. The utilization of CL technologies can significantly improve retrieval performance for both OMA and NOMA transmissions, particularly in scenarios characterized by low signal-to-noise ratio (SNR) and limited bandwidth. Furthermore, the integration of the proposed CL-SC scheme with OTA technologies in the NOMA transmission method attains optimal performance.

6.2 Related work

Image retrieval The goal of image retrieval is to recognize the identity of the images [165]. Given a query image, its similarities with images of different viewpoints from the gallery database are assessed. Through performing the nearest neighbors search, the image retrieval model can search gallery images containing the same identity as the query image, facilitating the prediction of the query image's identity. The retrieval performance can be evaluated through the top-1 retrieval accuracy [165].

The image retrieval problem has undergone extensive exploration in recent years [166]. Authors in [167] introduced attention-aided models, while [168] proposed a Horizontal Pyramid Matching approach. To achieve the SOTA performance, a transformer-based model is proposed in [169]. In the multi-view scenarios, [170] utilized a temporal attention mechanism, and [171] presented a parsing-guided cross-part reasoning network.

While these advanced methods can improve the model's performance significantly, they add tremendous computation complexity to the model architecture, which is critical to the remote inference problem. As the complex retrieval algorithm cannot be deployed and performed locally on the edge device, the necessity arises to transmit semantic features via a wireless communication channel for inference processing on the edge server, forming the focal point of our study.

Remote inference at the wireless edge Traditional communication systems aim to deliver the source signals from the transmitter to the receiver with minor distortion under specific bandwidth resources. However, with the rapid growth of IoT networks and the machine-to-machine short package communications, the goal of emerging semantic communication systems is shifting towards edge intelligence, which requires the edge devices to make accurate inferences [10, 50] rather than reconstructing the source signals.

Most previous literature on joint edge-device inference [172, 173] employed digital separation-based schemes. These schemes involve the compression and separate transmission of source signals, coupled with distributed inference for tasks situated at the wireless edge. Inspired by the success of DeepJSCC schemes in point-to-point transmission [35, 41, 43, 45, 47, 48, 160], a DeepJSCC-based retrieval-oriented communication scheme is proposed in [57], where both digital and analog communication schemes are comprehensively studied. The proposed DeepJSCC-based approach achieves SOTA performance in image retrieval tasks under all channel conditions at the wireless edge.

Following this, researchers in the study [10] delved into task-oriented communication in the context of edge inference. This investigation focuses on a specific scenario wherein an edge device transmits the extracted feature vector of a local data sample to the edge server for remote classification. Specifically, an information bottleneck (IB) framework is formalized to characterize the rate-distortion tradeoff between the informativeness of the encoded feature and the inference performance. Extensive experiments demonstrate a better rate-distortion tradeoff of the proposed method than baseline methods in dynamic channel conditions.

In contrast to the remote classification problem addressed in [10], which can be solved locally at the edge devices with enough computational resources, our focus instead lies on the image retrieval problem within the edge computing paradigm. In this context, remote inference becomes imperative, given that the gallery database is exclusively accessible at the edge server.

Multi-source collaborative edge inference Compared to the single-source scenario, the multi-source edge inference problem is much less studied in the literature. Existing studies over

multi-device wireless edge inference mainly focus on the classification [174] and image retrieval [57] tasks. In the context of the multi-source remote classification task, a task-oriented communication scheme for the multi-device edge inference task is proposed in [174], which achieves SOTA performance by utilizing the IB principle in the feature extraction process and the deterministic distributed information bottleneck (DDIB) principle in distributed feature encoding process. The model demonstrates superior performance compared to alternative feature encoding algorithms. For a more challenging multi-source retrieval problem, the authors in [86] explore the cooperation between different devices for the collaborative image retrieval task. This extension introduces an innovative approach aimed at optimizing performance in both OMA and NOMA channels. Empirical evaluations demonstrate that leveraging a distributed DeepJSCC pipeline for the correlated data within the NOMA transmission approach yields superior performance.

Nevertheless, no specific paradigm exists to explicitly exploit the data correlation between devices as prior knowledge in multi-source collaborative semantic communication systems. It is intuitive that incorporating such knowledge should lead to significant enhancements in inference accuracy, which motivates for our study in developing CL technologies for this problem.

6.3 System model

We consider an edge network comprising M distinct devices, each equipped to capture images of a common identity from various angles and potentially at differing levels of quality. These devices are interconnected through a shared wireless MAC to an edge server, as depicted in Fig. 6.2. The primary task of the edge server is to perform inference by matching the images received from these devices against a locally stored database to accurately identify and retrieve the image of the same identity.

The edge devices Consider the image captured by the i -th device, $\mathbf{s}_i \in \mathbb{R}^{h \times w \times c}$, where h , w and c denote the height, width, and number of color channels of the image, respectively, for

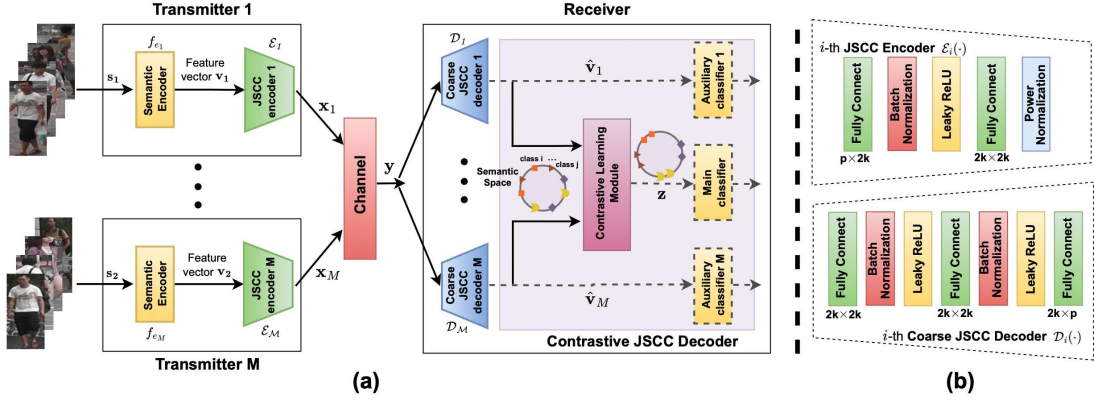


Figure 6.2: (a) Schematic of the proposed CL-SC scheme. This framework aims to refine the discriminative power of noisy received features by optimizing their representations in the semantic feature space. The optimization process specifically focuses on minimizing the feature distance between samples that share identical identities, while simultaneously maximizing the distance between samples that have different identities, enhancing the robustness and accuracy of identity matching. (b) Detailed architecture of the JSCC encoder and decoder used in our system.

$i \in [1, \dots, M]$. The i -th device first encodes its observed image s_i into a semantic feature vector $v_i \in \mathbb{R}^p$, represented as:

$$v_i = f_{e_i}(s_i) : \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{R}^p. \quad (6.1)$$

The resultant v_i is then transformed into the complex channel symbols $x_i \in \mathbb{C}^k$ as:

$$x_i = \mathcal{E}_i(v_i) : \mathbb{R}^p \rightarrow \mathbb{C}^k, \quad (6.2)$$

where k represents the number of channel subcarriers allocated to each edge device. This transformation can be executed through one of two primary encoding schemes: the separation-based digital scheme or the JSCC scheme. The separation-based digital scheme employs a sequence of feature compression, quantization, and arithmetic coding to encode the feature vectors into channel symbols. Contrarily, the JSCC scheme utilizes a specialized JSCC encoder that directly maps the semantic feature vector into channel symbols. Overall, the mapping from the source image to channel symbols in the i -th edge device can be written as:

$$x_i = \mathcal{T}_i(s_i) : \mathbb{R}^{h \times w \times c} \rightarrow \mathbb{C}^k. \quad (6.3)$$

In particular, the channel input $\mathbf{x}_i$ of each transmitter is subject to the following power constraint: $\frac{1}{k} \|\mathbf{x}_i\|_2^2 \leq P_s$, where we set $P_s = 1$ without loss of generality.

Channel model We consider two multiple access transmission schemes, characterized by a total bandwidth of B channel uses: OMA and NOMA transmission schemes. The general channel model can be represented as

$$\mathbf{y} = \mathcal{H}(\mathbf{x}_1, \dots, \mathbf{x}_M), \quad (6.4)$$

where $\mathbf{y} \in \mathbb{C}^B$ denotes the channel output vector.

We consider TDMA to realize OMA transmission, in which case each device transmits signals over orthogonal channels in a time-division manner. We assume each device is assigned an equal channel bandwidth, where the encoded channel symbols are represented as $\mathbf{x}_1, \dots, \mathbf{x}_M \in \mathbb{C}^k$ with $k = \frac{B}{M}$. We consider Rayleigh slow fading channel and denote the channel output by $\mathbf{y} \triangleq [\mathbf{y}_1; \dots; \mathbf{y}_M]^T \in \mathbb{C}^B$, wherein $\mathbf{y}_i \in \mathbb{C}^{\frac{B}{M}}$ is the channel output vector from the i -th edge device, given as:

$$\mathbf{y}_i = h_i \mathbf{x}_i + \mathbf{w}_i, \quad (6.5)$$

where $h_i \in \mathbb{C}$ is the channel gain. It remains constant for k channel uses and is drawn from a complex Gaussian distribution: $h_i \sim \mathcal{CN}(0, \sigma_h^2)$. $\mathbf{w}_i \in \mathbb{C}^k$ is the AWGN term, where each element of which is sampled from a complex Gaussian distribution $\mathcal{CN}(0, \sigma_w^2)$ in an independent and identically distributed (i.i.d) fashion. In particular, for the AWGN channel case, we set $h_i = 1$.

Within the alternative NOMA approach, distinct devices transmit the encoded channel symbols superimposed on each other over the shared wireless channel. Each transmitter will be allocated the full bandwidth of B channel uses, in which case the encoded channel symbols are represented as $\mathbf{x}_1, \dots, \mathbf{x}_M \in \mathbb{C}^k$ with $k = B$. It is worth noting that as input vectors are superimposed on each other in the transmission, the interference between them has to be considered, and there is only one channel output $\mathbf{y} \in \mathbb{C}^B$ from the channel instead of multiple channel outputs. Thus,

we have

$$\mathbf{y} = \sum_{i=1}^M h_i \mathbf{x}_i + \mathbf{w}, \quad (6.6)$$

where $\mathbf{w} \in \mathbb{C}^k$ is the additive AWGN vector sampled from $\mathcal{CN}(0, \sigma_w^2)$. The channel gains $h_i \sim \mathcal{CN}(0, \sigma_h^2)$, and remain constant throughout the duration of B channel uses.

The edge server Upon receiving the noisy channel outputs $\mathbf{y}$, the edge server undertakes the image retrieval task by decoding $\mathbf{y}$ to ascertain the identity of the source. Following methodologies from [57, 86], the receiver initially reconstructs the feature vector for each transmitting device from $\mathbf{y}$. This coarse decoding process is formalized as

$$\hat{\mathbf{v}}_i = \mathcal{D}_i(\mathbf{y}) : \mathbb{C}^B \rightarrow \mathbb{C}^p, \quad (6.7)$$

where $\mathcal{D}_i$ denotes the dedicated decoder for the i -th device. This reconstructed feature vector, $\hat{\mathbf{v}}_i$, encapsulates the semantic content extracted from the transmitted signal after channel distortions have been mitigated. Subsequently, the edge server employs a retrieval matching algorithm to these recovered feature vectors. This algorithm compares the semantic features $\hat{\mathbf{v}}_i$ against a local database to predict the most likely identity of the source $\hat{r}$.

6.4 Methodology

This section presents our Contrastive Learning-based Semantic Communication (CL-SC) scheme designed for cooperative edge inference. The overall workflow of the CL-SC scheme is depicted in Fig. 6.2(a), with a detailed algorithmic breakdown provided in Algorithm 5. The CL-SC approach initiates with the extraction of semantic features at the transmitter, which are then encoded for transmission via a JSCC encoder. At the receiver, the noisy received features undergo enhancement through a CL module, which optimizes these features to improve the efficacy of the image retrieval process.

Algorithm 5 The CL-SC paradigm for M devices**Input:**The observed image: $(s_1, \dots, s_M)$;Feature extractors: $(f_{e_1}, \dots, f_{e_M})$;JSCC encoder and decoder pairs: $(\mathcal{E}_1, \mathcal{D}_1), \dots, (\mathcal{E}_M, \mathcal{D}_M)$;Channel Model: $\mathcal{H}(\cdot)$;Contrastive learning module: $\mathcal{C}(\cdot)$;Image retrieval matching method: $\mathcal{M}(\cdot)$;**Output:** Object identity: r .**Step 1: Semantic Feature Encoding**1: **for** $i=1:M$ **do**2: $v_i = f_{e_i}(s_i)$

▸ Feature Encoding

3: **end for****Stage 2: JSCC encoding and coarse decoding**1: **for** $i=1:M$ **do**2: $x_i = \mathcal{E}_i(v_i)$

▸ JSCC Encoding

3: **end for**4: $y = \mathcal{H}(x_1, \dots, x_M)$

▸ Noisy MAC Channel

5: **for** $i=1:M$ **do**6: $\hat{v}_i = \mathcal{D}_i(y)$

▸ Coarse JSCC Decoding

7: **end for****Stage 3: Contrastive learning module**1: $z_{\theta_1}, \dots, z_{\theta_M} = \mathcal{C}(\hat{v}_1, \dots, \hat{v}_M)$

▸ Contrastive learning

Stage 4: Image retrieval1: $\hat{r} = \mathcal{M}(z_{\theta_1}, \dots, z_{\theta_M})$

▸ Nearest neighbour search

In our proposed system architecture, the transmitter incorporates a semantic feature extractor combined with a JSCC encoder; the receiver features corresponding coarse JSCC decoders and a subsequent CL module, designed to refine and utilize the correlated data for precise edge inference. The design of the JSCC encoder-decoder setup is illustrated in Fig. 6.2(b). Detailed functionalities and the operational principle of the CL module are elucidated in Fig 6.3. In the ensuing discussions, we will delve into five pivotal aspects of the CL-SC framework: overall workflow, JSCC transmission, CL module (for both $M = 2$ and $M > 2$ cases), the training strategy, and the design interpretation.

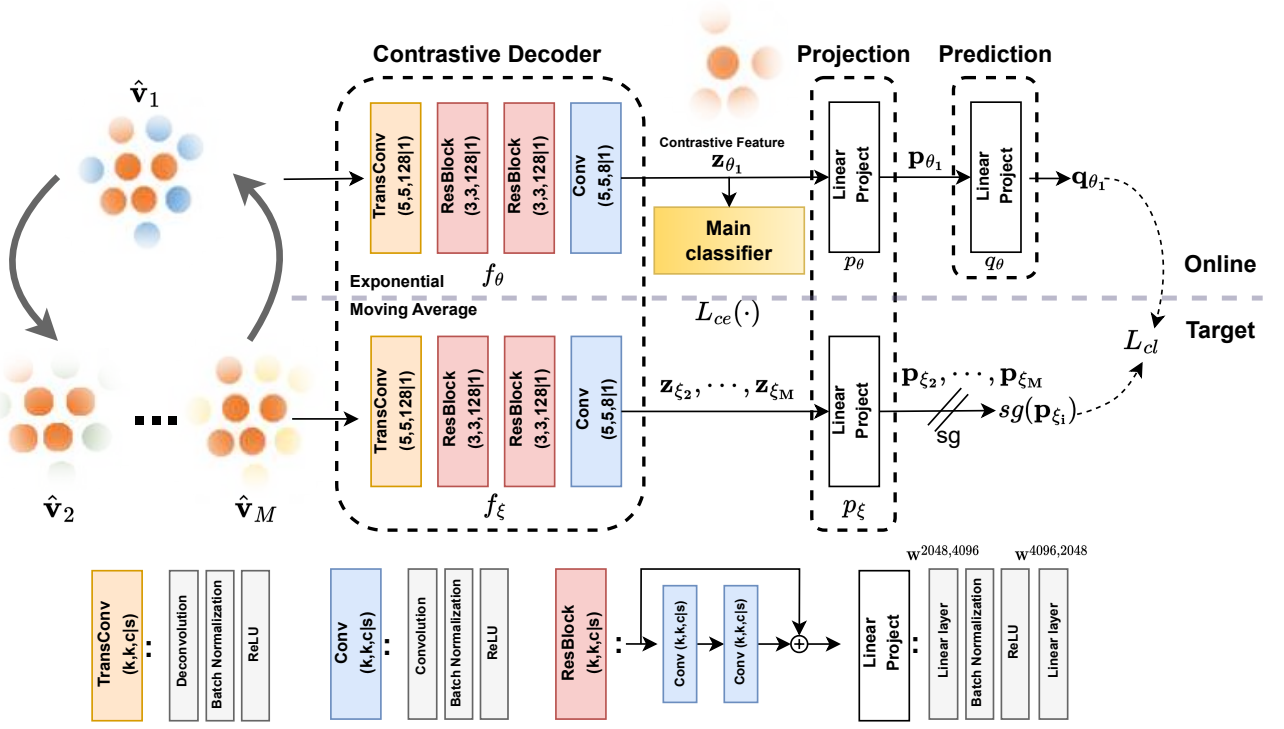


Figure 6.3: Contrastive learning module for M devices, where $(k, k, c|s)$ means convolutional operation with (k, k) kernel, out channel number c , stride s , and sg means stop-gradient operation.

6.4.1 Overall workflow

Aligned with the SOTA retrieval methodologies as detailed in prior studies [57, 86, 168], we adopt a consistent baseline for performance comparison. Specifically, within each transmitter of our system, we utilize a lightweight ResNet50 architecture [175] as the semantic feature extractor. This choice is motivated by ResNet50's balance of efficiency and effectiveness in feature extraction tasks. The operation within the i -th transmitter can be represented as $\mathbf{v}_i = f_{e_i}(\mathbf{s}_i)$, where f_{e_i} denotes the feature extraction function of ResNet50, concluding with an average pooling operation to condense the extracted features into $\mathbf{v}_i$. These features are then transmitted to the edge server via a JSCC coding mechanism, which will be elaborated upon in Section 6.4.2.

At the edge server, the transmitted features are first coarsely reconstructed as $\hat{\mathbf{v}}_i$. This reconstructed data undergoes further refinement through a CL module, detailed in Section 6.4.3, resulting in enhanced feature vectors $\mathbf{z}_{\theta_i}$. These vectors are subsequently input into a retrieval

matching algorithm to predict the identity of the source signals:

$$\hat{r} = \mathcal{M}(\mathbf{z}_{\theta_1}, \dots, \mathbf{z}_{\theta_M}). \quad (6.8)$$

To ensure a fair comparison with existing methodologies, we utilize a nearest neighbors search algorithm, as commonly employed in the literature [57, 86]. This algorithm calculates the Euclidean distances to quantify similarities between the query images and various viewpoints within the gallery. Through this nearest neighbors search, we generate a rank-ordered list of gallery images, sorted by descending similarity to the query image, enabling performance evaluation based on top-1 accuracy.

6.4.2 JSCC transmission

Motivated by the demonstrated effectiveness of JSCC in enhancing semantic data transmission [35, 41, 43, 45, 47, 48], we have incorporated JSCC technologies within our CL-SC paradigm. We introduce a coarse JSCC decoder to reconstruct the semantic feature vector for three reasons: (a) CL-SC models the coarse feature transmission as a disturbance, which a contrastive decoder can later refine; (b) it allows easy extension to multiple transmitters without requiring a joint, tailored design; and (c) it yields better performance, as observed in prior works [57, 86]. Specifically, semantic features are transmitted through a JSCC encoder and then reconstructed via a coarse JSCC decoder, as depicted in Fig. 6.2 (a). The architecture for each JSCC encoder and decoder pair implemented across our system is designed to ensure streamlined processing, with detailed components shown in Fig. 6.2 (b).

- **JSCC encoder.** In each transmitter, the JSCC encoder processes the semantic feature vector $\mathbf{v}_i$ through two FC layers, each followed by batch normalization and Leaky rectified linear unit (ReLU) activation functions. This sequence transforms the semantic features into encoded channel vectors $\mathbf{x}_i \in \mathbb{C}^k$: $\mathbf{x}_i = \mathcal{E}_i(\mathbf{v}_i)$.
- **Coarse JSCC decoder.** Conversely, each coarse JSCC decoder reconstructs the semantic feature vector from the transmitted signals. Specifically, the received signal $\mathbf{y}$ is

processed through three FC layers, each also followed by batch normalization and Leaky ReLU activation, to reconstruct the coarse feature vector as $\hat{\mathbf{v}}_i$: $\hat{\mathbf{v}}_i = \mathcal{D}_i(\mathbf{y})$.

- **Pretraining.** To facilitate effective JSCC transmission, we pre-train the JSCC encoders $\mathcal{E}_1, \dots, \mathcal{E}_M$ and decoders $\mathcal{D}_1, \dots, \mathcal{D}_M$ using a MSE loss function. This function measures the fidelity of reconstructed features compared to the original features:

$$\mathcal{L}_{mse} = \sum_{i=1}^M \text{MSE}(\mathbf{v}_i, \hat{\mathbf{v}}_i) \triangleq \sum_{i=1}^M \mathbb{E}[\|\mathbf{v}_i - \hat{\mathbf{v}}_i\|_2^2], \quad (6.9)$$

where the expectation accounts for variability in both source and channel distributions. Minimizing $\mathcal{L}_{mse}$ helps refine the parameters of the JSCC system, with the objective of optimizing transmission distortion. This pretraining process guarantees the transfer and retention of semantic feature information at a coarse level.

6.4.3 The CL module

We propose a cross-view CL module to refine the reconstructed coarse semantic features. To facilitate a comprehensive understanding of the module, we begin by considering the case where $M = 2$, whose architecture and operations are elaborated in Fig. 6.3 and Algorithm 6.

Inspired by [154, 155, 176], our CL module employs a mutual prediction strategy to optimize noisy $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ for more representative semantics. Specifically, we treat $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ as distinct noisy observations of the source identity, as in remote source coding problems [161], and minimize the conditional entropy between them by cross-prediction operations.

The proposed CL module comprises an online network and a target network. The online network, taking $\hat{\mathbf{v}}_i$ as input, is optimized to predict the targets produced by the target network with input $\hat{\mathbf{v}}_j$. Concurrently, the target network is optimized using an average moving strategy to avert network collapse. In the subsequent sections, we will detail the CL module through two parts: cross-view prediction and loss function.

Algorithm 6 Cross-view contrastive learning algorithm**Input:**

Reconstructed semantic feature vectors: $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2$
 One-hot identity label vector and main classifier: $\mathbf{r}, I_\theta$
 Online decoder, projector, predictor: $f_\theta(\cdot), p_\theta(\cdot), q_\theta(\cdot)$
 Target decoder, projector: $f_\xi(\cdot), p_\xi(\cdot)$
 Optimizer, target decay rate: Opt_1, Opt_2, τ

Output:

Optimized contrastive feature: $\mathbf{z}_{\theta_1}, \mathbf{z}_{\theta_2}$

Training phase:

```

1: for each epoch do
2:    $\mathbf{z}_{\theta_1} = f_\theta(\hat{\mathbf{v}}_1), \mathbf{z}_{\theta_2} = f_\theta(\hat{\mathbf{v}}_2)$  ▷ Online decoding
3:    $\mathcal{L}_{ce} = CE(I_\theta(\mathbf{z}_{\theta_1}), \mathbf{r}) + CE(I_\theta(\mathbf{z}_{\theta_2}), \mathbf{r})$  ▷ Classification loss
4:    $\theta \leftarrow Opt_1(\theta, \nabla_\theta \mathcal{L}_{ce})$  ▷ Update online network
5:   if epoch%2 = 0 then
6:      $\mathbf{z}_{\xi_1} = f_\xi(\hat{\mathbf{v}}_1), \mathbf{z}_{\xi_2} = f_\xi(\hat{\mathbf{v}}_2)$  ▷ Target decoding
7:      $\mathbf{p}_{\xi_1} = p_\xi(\mathbf{z}_{\xi_1}), \mathbf{p}_{\xi_2} = p_\xi(\mathbf{z}_{\xi_2})$  ▷ Target projection
8:      $\mathbf{p}_{\theta_1} = p_\theta(\mathbf{z}_{\theta_1}), \mathbf{p}_{\theta_2} = p_\theta(\mathbf{z}_{\theta_2})$  ▷ Online projection
9:      $\mathbf{q}_{\theta_1} = q_\theta(\mathbf{p}_{\theta_1}), \mathbf{q}_{\theta_2} = q_\theta(\mathbf{p}_{\theta_2})$  ▷ Online prediction
10:     $\mathcal{L}_{cl} = 4 - 2(\frac{\langle \mathbf{q}_{\theta_1}, \mathbf{p}_{\xi_2} \rangle}{\|\mathbf{q}_{\theta_1}\|_2 \|\mathbf{p}_{\xi_2}\|_2} + \frac{\langle \mathbf{q}_{\theta_2}, \mathbf{p}_{\xi_1} \rangle}{\|\mathbf{q}_{\theta_2}\|_2 \|\mathbf{p}_{\xi_1}\|_2})$  ▷ Cross-view prediction loss
11:     $\theta \leftarrow Opt_2(\theta, \nabla_\theta \mathcal{L}_{cl})$  ▷ Update online network
12:     $\xi \leftarrow \tau \xi + (1 - \tau) \theta$  ▷ Update target network
13:   end if
14: end for

```

Inference phase:

```

1:  $\mathbf{z}_{\theta_1} = f_\theta(\hat{\mathbf{v}}_1), \mathbf{z}_{\theta_2} = f_\theta(\hat{\mathbf{v}}_2)$  ▷ Optimized feature
2:  $\mathbf{r} = \mathcal{M}(\mathbf{z}_{\theta_1}, \mathbf{z}_{\theta_2})$  ▷ Retrieval matching

```

Cross-view prediction As illustrated in Fig. 6.3, online and target networks share the same structure, but with different parameters θ and ξ . Both networks comprise a contrastive decoder f_θ and f_ξ , alongside a projection layer p_θ and p_ξ , respectively. Additionally, the online network integrates a prediction layer q_θ .

Through the cross-view prediction training, the online contrastive decoder aims to optimize $\hat{\mathbf{v}}_i$ into a contrastive feature $\mathbf{z}_{\theta_i}$ with more representative semantics. The operations of the contrastive decoders in the online and target networks, applied to the input $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{v}}_j$, can be represented as:

$$\begin{aligned} \mathbf{z}_{\theta_i} &= f_\theta(\hat{\mathbf{v}}_i), \\ \mathbf{z}_{\xi_j} &= f_\xi(\hat{\mathbf{v}}_j). \end{aligned} \tag{6.10}$$

Inspired from [154, 155], we empirically design an additional projection layer before the prediction operation to enhance performance. The operations of the projection layer for the online and target networks are denoted as:

$$\begin{aligned} \mathbf{p}_{\theta_i} &= p_{\theta}(\mathbf{z}_{\theta_i}), \\ \mathbf{p}_{\xi_j} &= p_{\xi}(\mathbf{z}_{\xi_j}). \end{aligned} \tag{6.11}$$

Then, the result of the prediction layer in the online network is denoted as:

$$\mathbf{q}_{\theta_i} = q_{\theta}(\mathbf{p}_{\theta_i}). \tag{6.12}$$

which is optimized as the prediction of the $\mathbf{p}_{\xi_j}$.

Considering the bi-view input data, during the training phase, each view $\hat{\mathbf{v}}_i$ is fed into the online network as $\mathbf{q}_{\theta_i}$ to predict the regression target $\mathbf{p}_{\xi_j}$ from the target network, which employs another complementary view $\hat{\mathbf{v}}_j$ as the input. Specifically, we interchange the roles of $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{v}}_j$ by presenting them as inputs to both the online and target networks during the training phase.

Loss function To train the CL module, we introduce a classification loss $\mathcal{L}_{ce}$ and a cross-view prediction loss $\mathcal{L}_{cl}$, where we optimize $\mathcal{L}_{ce}$ every epoch and $\mathcal{L}_{cl}$ every two epochs.

For the $\mathcal{L}_{ce}$, we introduce a ‘main classifier’, which is an FC layer followed by a softmax operation and denoted as $I_{\theta}(\cdot)$. The loss function $\mathcal{L}_{ce}$ is given as:

$$\mathcal{L}_{ce} \triangleq \frac{1}{2} (CE(I_{\theta}(\mathbf{z}_{\theta_i}), \mathbf{r}) + CE(I_{\theta}(\mathbf{z}_{\theta_j}), \mathbf{r})), \tag{6.13}$$

where $\mathbf{r} \in \mathbb{R}^D$ is the one-hot identity label vector, D is the number of classes, and $CE(\mathbf{x}, \mathbf{r}) \triangleq -\frac{1}{D} \sum_{i=1}^D (r_i \log(x_i) + (1 - r_i) \log(1 - x_i))$ is the cross entropy loss. Note that this main classifier is only introduced in the training phase and will be removed in the inference phase, where $\mathbf{z}_{\theta_i}$ and $\mathbf{z}_{\theta_j}$ are directly used for retrieval. The online network parameters θ are then updated by

the back-propagation with optimizer Opt_1 over $\mathcal{L}_{ce}$ as:

$$\boldsymbol{\theta} \leftarrow Opt_1(\boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{ce}). \quad (6.14)$$

The cross-view prediction loss $\mathcal{L}_{cl}$ measures the distance between the online prediction output and the target projection output, which is defined as:

$$\begin{aligned} \mathcal{L}_{cl} &= \|\bar{\mathbf{q}}_{\boldsymbol{\theta}_1} - \bar{\mathbf{p}}_{\boldsymbol{\xi}_2}\|_2^2 + \|\bar{\mathbf{q}}_{\boldsymbol{\theta}_2} - \bar{\mathbf{p}}_{\boldsymbol{\xi}_1}\|_2^2 \\ &= 4 - 2\left(\frac{\langle \mathbf{q}_{\boldsymbol{\theta}_1}, \mathbf{p}_{\boldsymbol{\xi}_2} \rangle}{\|\mathbf{q}_{\boldsymbol{\theta}_1}\|_2 \|\mathbf{p}_{\boldsymbol{\xi}_2}\|_2} + \frac{\langle \mathbf{q}_{\boldsymbol{\theta}_2}, \mathbf{p}_{\boldsymbol{\xi}_1} \rangle}{\|\mathbf{q}_{\boldsymbol{\theta}_2}\|_2 \|\mathbf{p}_{\boldsymbol{\xi}_1}\|_2}\right), \end{aligned} \quad (6.15)$$

where $\bar{\mathbf{q}}_{\boldsymbol{\theta}_i} \triangleq \frac{\mathbf{q}_{\boldsymbol{\theta}_i}}{\|\mathbf{q}_{\boldsymbol{\theta}_i}\|_2}$ is the normalized online prediction output, and $\bar{\mathbf{p}}_{\boldsymbol{\xi}_i} \triangleq \frac{\mathbf{p}_{\boldsymbol{\xi}_i}}{\|\mathbf{p}_{\boldsymbol{\xi}_i}\|_2}$ is the normalized target projection output.

To avoid collapse and trivial results during training, we adopt an exponential moving average updating strategy. Specifically, we use an optimizer Opt_2 to minimize $\mathcal{L}_{cl}$ only with respect to the online network parameters $\boldsymbol{\theta}$ as:

$$\boldsymbol{\theta} \leftarrow Opt_2(\boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{cl}), \quad (6.16)$$

where only $\boldsymbol{\theta}$ is updated while freezing $\boldsymbol{\xi}$, and there is a stop-gradient operation for the output of the target network. The target network $\boldsymbol{\xi}$ is then updated by the exponential moving average as:

$$\boldsymbol{\xi} \leftarrow \tau \boldsymbol{\xi} + (1 - \tau) \boldsymbol{\theta}, \quad (6.17)$$

where $\tau = 0.99$ is set as the target delay rate.

Intuitively, training $\mathcal{L}_{ce}$ prevents vanishing gradients and improves the performance, while training $\mathcal{L}_{cl}$ encourages more discriminative features with better representations and semantics after noisy channels. Our numerical results, which are shown later, will confirm these intuitions.

Multiple sources ($M > 2$) While the above discussions have primarily assumed a two-source setup, the proposed CL-SC scheme is adaptable to scenarios involving more than two

sources. In configurations where $M > 2$, the basic optimization strategy, including the use of the exponential moving average updating, remains unchanged. However, the loss functions are specifically modified to effectively handle the complexity introduced by multiple sources.

First, the classification loss $\mathcal{L}_{ce}(\cdot)$ is expanded to optimize the online network across M devices in each epoch:

$$\mathcal{L}_{ce} = \frac{1}{M} \sum_{i=1}^M CE(I_{\theta}(\mathbf{z}_{\theta_i}), \mathbf{r}). \quad (6.18)$$

Second, the cross-view prediction loss $\mathcal{L}_{cl}(\cdot)$ is modified to encompass every distinct pair of views to ensure robustness against variance in source perspectives:

$$\mathcal{L}_{cl} = \sum_{i=1}^M \sum_{j=1, j \neq i}^M \|\bar{\mathbf{q}}_{\theta_i} - \bar{\mathbf{p}}_{\xi_j}\|_2^2. \quad (6.19)$$

6.4.4 Training strategies

In contrast to the adoption of an end-to-end training approach, our proposed CL-SC paradigm employs a four-step training strategy: the semantic feature encoders (T_1), the JSCC module (T_2), the CL module (T_3), and an end-to-end joint training phase (T_4). This training strategy involves the separate pre-training of each network component before the final joint training stage, facilitating a more systematic and efficient training process.

Specifically, the semantic feature encoder is pre-trained on the ImageNet dataset in step T_1 , where we incorporate auxiliary classifiers during the training phase, which is implemented as one FC layer and a softmax operation. The auxiliary loss function is introduced during the pre-training as:

$$\mathcal{L}_{ce_{aux}}(\mathbf{v}) = \frac{1}{M} \sum_i^M CE(I_i(\mathbf{v}_i), \mathbf{r}), \quad (6.20)$$

where $I_1(\cdot)$ and $I_2(\cdot)$ are the operations performed by the auxiliary classifiers. In step T_2 , the pre-trained feature encoders are frozen, and only the JSCC autoencoders are trained with the average MSE loss (Eqn. 6.9) between the transmitted and reconstructed feature vectors. In step T_3 , the semantic feature encoders and JSCC autoencoders are frozen, and only the CL

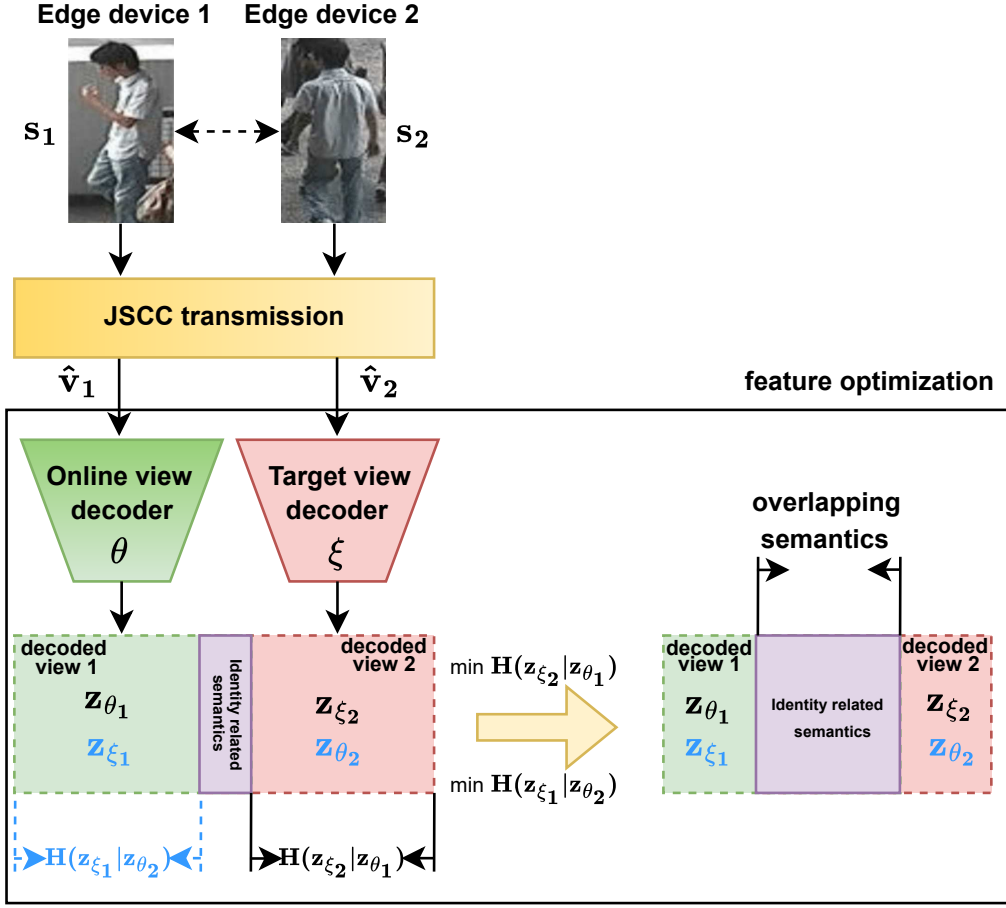


Figure 6.4: Design interpretation of our cross-view CL module, where the dotted rectangles denote the information (feature space) contained in the decoded view z_{θ_1} and z_{ξ_2} , the solid rectangle denotes the overlapped information (identity-related semantics). The blue notation means the operations after exchanging the input views for the online and target networks during the training phase.

module is trained with the $\mathcal{L}_{ce}$ and $\mathcal{L}_{cl}$, as shown in Algorithm 6. In step T_4 , we jointly train all components with $\mathcal{L}_{ce_{aux}}(\hat{v})$, $\mathcal{L}_{ce}$, and $\mathcal{L}_{cl}$.

During the inference phase, all classifiers are discarded, and we directly utilize the optimized feature $z_{\theta_1}, \dots, z_{\theta_M}$ obtained from the CL module for the edge retrieval task.

6.4.5 Design interpretation

In this section, we elucidate the intuition and interpretation of the proposed CL module. The proposed cross-view CL module is intricately tailored to enhance the optimization of the received channel symbols of the source signals by considering their correlations.

As depicted in Fig. 6.4 when $M = 2$, following the JSCC transmission over a MAC, a pair of correlated source signals ($\mathbf{s}_1$ and $\mathbf{s}_2$) undergoes a transformation within two distinct feature spaces, resulting in their respective counterparts: $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$. The correlation between two sources implies that the $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ should exhibit overlap or shared consistent semantics in the feature space. In contrast, the divergence of information within these feature spaces arises from dissimilar distributions attributed to varying backgrounds, variations from noisy channels, and the distributed nature inherent in separate JSCC networks. The objective of the CL module is to refine two feature spaces $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2$ into $\mathbf{z}_1, \mathbf{z}_2$, such that they encompass a maximal amount of overlapped information, especially the semantics associated with the shared identity. In practice, we propose a cross-view prediction strategy to minimize the conditional entropy between generated distinct views, $H(\mathbf{z}_2|\mathbf{z}_1)$ and $H(\mathbf{z}_1|\mathbf{z}_2)$, to encourage above optimization.

Specifically, the proposed cross-view prediction method utilizes an online-target twin network configuration, where the online network optimizes feature $\hat{\mathbf{v}}_i$ through minimizing the conditional entropy between the resultant features $\mathbf{z}_{\theta_i}$ and $\mathbf{z}_{\xi_j}$, as visualized in Fig. 6.4. A target network with the same structure as the online network is designed to form a contrastive learning mode and prevent trivial solutions. The underlying intuition is that when two views are optimally encoded in this problem, their encoded feature representations should completely overlap, indicating the elimination of all uncorrelated semantics. Consequently, we have $H(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i}) = -\mathbb{E}_{p_{\mathbf{z}_{\theta_i}, \mathbf{z}_{\xi_j}}} [\log p(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})] = 0$, yielding the most representative feature that integrates information from both views for the edge retrieval.

This optimization objective can be solved by introducing a variational distribution $q(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})$ and maximizing the lower bound $\mathbb{E}_{p_{\mathbf{z}_{\theta_i}, \mathbf{z}_{\xi_j}}} [\log q(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})]$, that is:

$$\begin{aligned} \mathbb{E}_{p_{\mathbf{z}_{\theta_i}, \mathbf{z}_{\xi_j}}} [\log p(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})] &= \mathbb{E}_{p_{\mathbf{z}_{\theta_i}, \mathbf{z}_{\xi_j}}} [\log q(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})] + \\ &\quad D_{KL}(p(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i}) \| q(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})), \end{aligned} \quad (6.21)$$

where above variational distribution $q(\mathbf{z}_{\xi_j}|\mathbf{z}_{\theta_i})$ is normally assumed with a Gaussian distribution $\mathcal{N}(\mathbf{z}_{\xi_j}; G(\mathbf{z}_{\theta_i}), \sigma_q^2 \mathbf{I})$, where $G(\cdot)$ is a parameterized model, mapping the $\mathbf{z}_{\theta_i}$ to $\mathbf{z}_{\xi_j}$ and $\sigma_q^2 \mathbf{I}$ is

the variance matrix. In practice [155, 176], after ignoring the constant part derived from the Gaussian process during optimization, maximizing the $\mathbb{E}_{p_{\mathbf{z}_{\theta_i}, \mathbf{z}_{\xi_j}}} [\log q(\mathbf{z}_{\xi_j} | \mathbf{z}_{\theta_i})]$ is equivalent to the minimization of

$$\mathbb{E}_{p_{\mathbf{z}_{\theta_i}, \mathbf{z}_{\xi_j}}} [\|\mathbf{z}_{\xi_j} - G(\mathbf{z}_{\theta_i})\|_2^2]. \quad (6.22)$$

Considering a symmetric bi-view training data $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2$, we can derive the optimization task as:

$$\min_{\theta, \xi} \mathbb{E}_{\theta, \xi} [\|\mathbf{z}_{\xi_1} - G(\mathbf{z}_{\theta_2})\|_2^2 + \|\mathbf{z}_{\xi_2} - G(\mathbf{z}_{\theta_1})\|_2^2]. \quad (6.23)$$

Motivated by previous CL literature [155, 176, 177], we undertake an empirical design for $G(\cdot)$, which incorporates an additional projection layer and l -2 normalization operations prior to the prediction. This design choice leads to the formulation of our model as $G(\cdot) \triangleq q_{\theta}(p_{\theta}(\mathbf{z}_{\theta_i}))$, as depicted in Fig. 6.3, and the corresponding loss function, as defined by Eqn. (6.15).

Based on the aforementioned derivations, the proposed CL module facilitates the reduction of conditional entropy among distinct feature views, thereby encouraging shared semantic information while discarding inconsistent information from the background and noisy channels. Consequently, our well-trained model exhibits the ability to effectively transfer features across diverse ‘views’ from different source distributions and noisy channel disturbance. This capability leads to the generation of more representative and discriminative features over the wireless edge, thereby enhancing the coherence of the semantic identity and improving the edge inference performance.

6.5 Training and evaluation

This section presents numerical experiments to evaluate the performance of our proposed CL-SC scheme across diverse channel conditions and bandwidths. We employ the single-source JSCC scheme [57], two-source JSCC scheme [86] and separation-based digital scheme [57] as the benchmark. In particular, the two-source separation-based digital scheme incorporates a feature compressor with quantization and arithmetic coding operations to encode feature vectors for

communication. Note that we assume MAC capacity-achieving codes with equal rate allocation for each transmitter in the digital scheme. Therefore, the reported performance of the digital scheme is not achievable in practice, particularly for the very low channel bandwidth considered here.

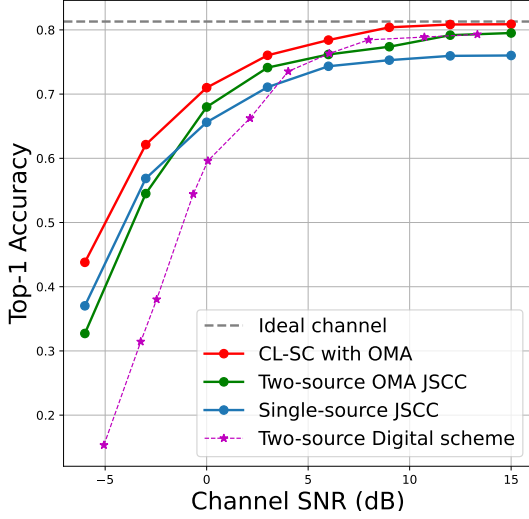
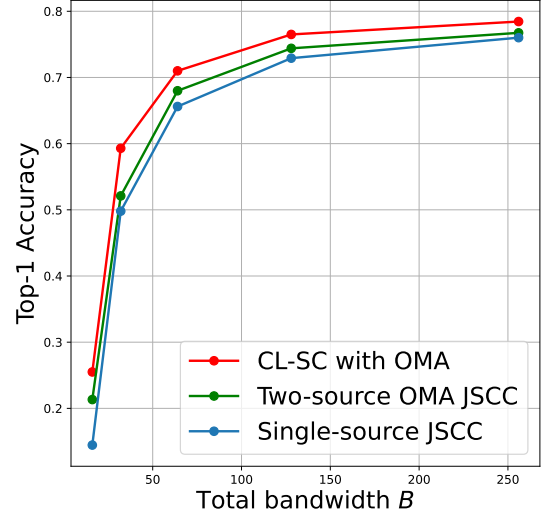
We investigate both OMA and NOMA transmission schemes with AWGN and Fading channels to evaluate the performance gains from the proposed CL module. To note, there is no channel state information available in the Fading channel scenario. It is noteworthy that the NOMA-based approach attains SOTA performance, emphasizing the benefits derived from our CL module and its integration of feature-over-the-air design considerations.

6.5.1 Experimental setup

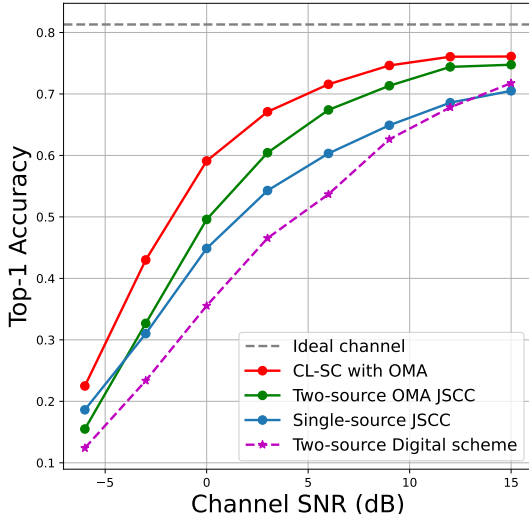
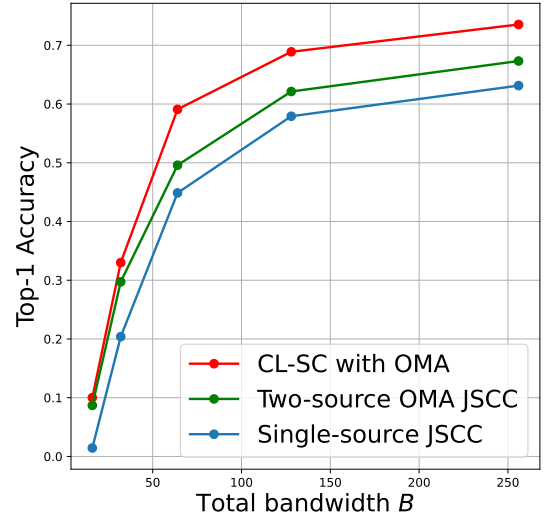
Unless stated otherwise, all models are trained and tested over the same channel SNRs with a total channel bandwidth of $B = 64$ symbols, and we set $M = 2$. We consider the Market-1501 [165] dataset. To align with the previous benchmarks and form a balanced dataset, we adopt the same camera views as in [86]: $[0, 2], [0, 3], [1, 3], [1, 5], [2, 4], [4, 5]$ and resize the input images into the 256×128 resolution. Performance is evaluated with top-1 accuracy [165]. Our model is implemented in Pytorch with two GTX 3090Ti GPUs. Each fading gain follows $h_i \sim \mathcal{CN}(0, \sigma_h^2)$ with $\sigma_h = 1$. To measure the channel quality, the average channel SNR μ is defined as

$$\mu \triangleq 10 \log_{10} \frac{\sigma_h^2}{\sigma_w^2} \text{dB}. \quad (6.24)$$

In the T_1 , we pre-train the semantic feature encoder for 30 epochs with a batch size of 16 by a stochastic gradient descent (SGD) optimizer. The learning rate is set to 0.01, while momentum is set to 0.9. The output dimension of the feature encoder is configured as 2048, utilizing an adaptive average pooling operation. In the T_2 , we train the JSCC autoencoders by the SGD optimizer for 100 epochs with a learning rate of 0.1. For the T_3 , we train the contrastive module with 40 epochs and a batch size of 512. Two separate SGD optimizers are employed to optimize the $\mathcal{L}_{cl}$ and $\mathcal{L}_{ce}$ with the same learning rate of 0.01. In the T_4 , the entire network is jointly

(a) AWGN-OMA with $B = 64$.

(b) AWGN-OMA with SNR=0 dB.

(c) Fading-OMA with $B=64$.

(d) Fading-OMA with SNR=0 dB.

Figure 6.5: Top-1 retrieval accuracy of the different schemes for the OMA transmission over the AWGN and Fading channel across various SNR and bandwidth values.

trained for 40 epochs. The learning rates for the individual components, namely the classifier, JSCC autoencoders, and the remaining networks, are set to 0.01, 0.001, and 0.01, respectively.

6.5.2 CL-SC scheme in OMA transmission

We first conduct a comparative analysis of the proposed CL-SC scheme against existing single-source, two-source JSCC schemes, and a separation-based digital scheme within the OMA transmission. The performance of different models versus diverse channel SNR values in both

AWGN and fading channels is plotted in Fig. 6.5a and Fig. 6.5c. Additionally, Fig. 6.5b and Fig. 6.5d illustrate the performance of different models across varying bandwidth values B for both AWGN and fading channels under a fixed $SNR = 0$ dB.

In Fig. 6.5a, we plot the top-1 accuracy of different models over the AWGN-OMA channel. In general, the proposed CL-SC scheme outperforms all other existing methods over the AWGN channel across all SNR values, especially in the low SNR regimes. The enhancements in performance through the transition from single-source strategies to two-device JSCC schemes underscore the advantages of integrating multiple devices for collaborative edge inference. Furthermore, there exists significant further performance gain (up to 11.08%) when progressing from two-device JSCC schemes to the proposed CL-SC scheme. This observation highlights that incorporating multi-source correlations via a contrastive approach can be advantageous for edge inference, particularly in challenging channel conditions and resource-constrained scenarios.

We can also observe that the performance gain of the CL-SC scheme decreases as the channel conditions get better. This decline can be attributed to the fact that under favorable channel conditions, the transmission distortion of signals becomes acceptable for edge inference, leading to a reduction in the benefits derived from source correlations. Concurrently, the previous two-source JSCC cannot outperform the traditional digital scheme with capacity-achieving channel codes in the high SNRs. Moreover, it is outperformed by the single-source JSCC scheme in low SNRs, where it might be more beneficial to allocate all the channel resources to a single transmitter rather than transmitting two query features, both with low quality. Our proposed CL-SC scheme resolves the above issues and attains a SOTA performance across all SNRs within the OMA transmission scenario. This superiority is attributed to the additional CL module, enabling the model to exploit signal correlations to enhance the inference performance.

As shown in Fig. 6.5b, our CL-SC consistently outperforms the one-source and two-source JSCC schemes, where improvements of up to 7.2% and 11.04% for all bandwidth B values can be observed. We can also observe that our CL-SC can significantly improve performance, especially in low bandwidth regimes. However, CL-JSCC cannot bring significant improvements in high bandwidth scenarios. It is because, with enough bandwidth, the two-source JSCC

scheme can also perform well and preserve enough semantics for the edge retrieval task.

Fig. 6.5c and Fig. 6.5d consider the slow fading channel within the OMA transmission. In comparison to alternative methods for fading channels in OMA transmission, our proposed CL-SC scheme can still achieve SOTA performance over all SNRs. Notably, our CL-SC exhibits additional advantages in the low SNR regime, where the CL-SC can improve the top-1 accuracy up to 9.51% compared with the previous best methods. The findings illustrated in Fig. 6.5d demonstrate consistent enhancements (up to 9.51% and 14.23%) from our CL-SC for all bandwidth values over the fading channel, when comparing with two-source and single-source JSCC schemes. Particularly, our CL-SC can significantly improve performance in low bandwidth regimes, such as $q = 32, 64, 128$. Intriguingly, our CL-SC also exhibits performance improvements in certain high-bandwidth regions for the fading channel, showing its advantage in addressing challenges posed by the fading channel across various bandwidths. In the assessment of various models within both AWGN and fading channels, the CL-SC can improve the performance more significantly in the fading channel. As the fading channel may result in more distortions and diversities during the transmission, the additional optimization steps from the contrastive learning module are expected to yield more improvements. This observation underscores that CL-SC is a more appropriate method for dealing with the challenging fading channels.

In summary, our CL-SC scheme achieves SOTA performance in both AWGN and fading channels in the OMA transmission across all SNR and bandwidth values. We can conclude that exploiting contrastive learning over multi-source correlations can significantly benefit edge inference for the OMA transmission, especially in poor channel conditions and limited resources.

6.5.3 CL-SC scheme in NOMA transmission

Motivated by the enhanced performance observed in the NOMA-JSCC transmission scheme [86], our investigation extends the proposed CL-SC scheme to the NOMA transmission with OTA technologies, namely as feature-over-the-air. The feature-over-the-air design of the NOMA

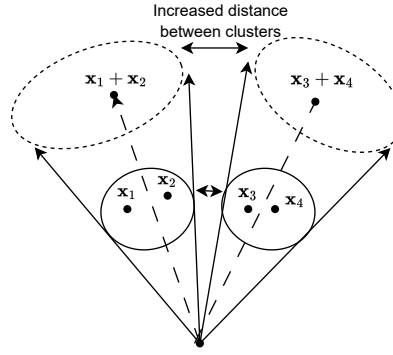


Figure 6.6: Over-the-air superposition of feature vectors from a cluster increases the contrast between different class clusters, where $\mathbf{x}_1$, $\mathbf{x}_2$ are two data points belonging to the same cluster, while $\mathbf{x}_3$ and $\mathbf{x}_4$ correspond to two other data points in another cluster.

scheme and the proposed CL module are two sides of one coin, both of which encourage increasing the feature distance between samples of different identities and minimizing the distance between samples of the same identity.

We first present the intuitive explanations regarding the intrinsic complementary relationship between the feature-over-the-air and CL designs. Considering the sources are correlated and associated with a common identity, the extracted semantic features and the two transmitted signals should be similar and “close” in the latent space. If we visualize the semantic features/channel symbols associated with different identities mapped to different clusters in a latent space, adding pairs of vectors from the same cluster will increase the feature distance between the new clusters formed of the added vectors, as shown in Fig. 6.6. By enhancing the contrast between distinct clusters within latent space, the feature-over-the-air design facilitates the preservation of overlapped semantics and heightens the discriminability of features for the receiver. This intrinsic characteristic aligns seamlessly with the objectives of the CL module. Consequently, the superposition of the correlated signal from another device for transmission fits the CL-SC framework, contributing to the improvements of inference accuracy rather than acting as interference.

Subsequently, we conduct extensive numerical experiments to evaluate the performance of the CL-SC scheme in NOMA transmission. Specifically, we plot the top-1 accuracy performance of various models across different SNR values for both AWGN and slow fading channels in Fig. 6.7a and Fig. 6.7c. In addition, Fig. 6.7b and Fig. 6.7d present the performance of

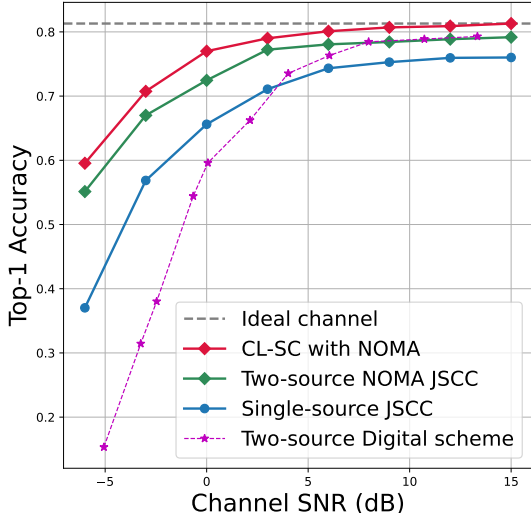
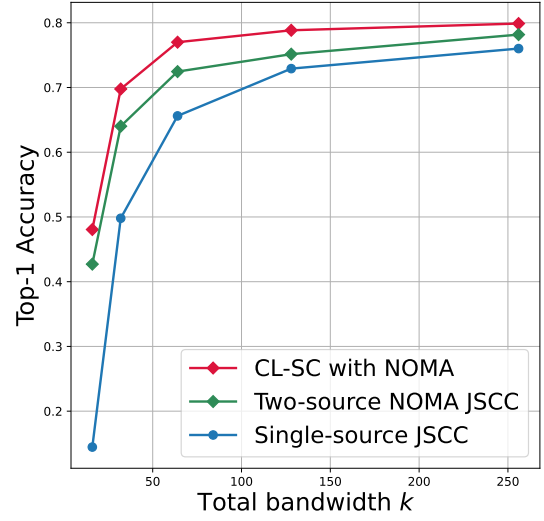
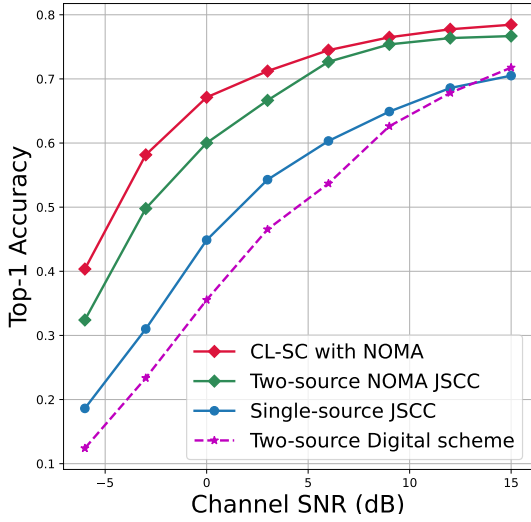
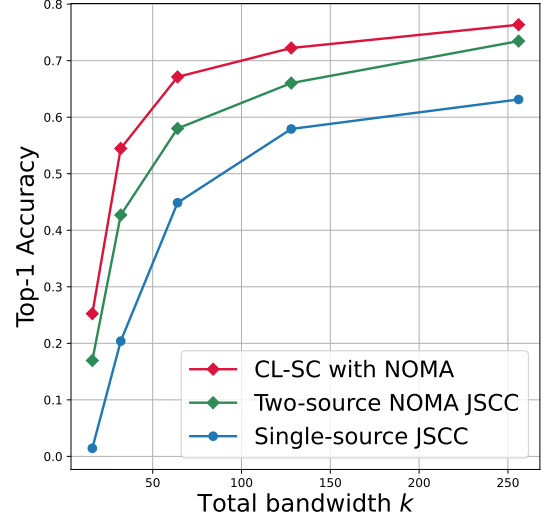
(a) AWGN-NOMA with $B=64$.(b) AWGN-NOMA with $\text{SNR}=0\text{dB}$ (c) Fading-NOMA with $B=64$.(d) Fading-NOMA with $\text{SNR}=0\text{dB}$.

Figure 6.7: Top-1 retrieval accuracy of the different schemes for the NOMA transmission over the AWGN and Fading channel across various SNR and bandwidth values.

different models across various bandwidth values B for $\text{SNR}=0\text{dB}$ in the AWGN and slow fading channels, respectively.

From Figs. 6.7a and 6.7c, we can observe that two-source NOMA schemes outperform the single-source JSCC scheme at all channel SNRs considered, showing that incorporating multiple views to make a collaborative decision is essential to improve the retrieval accuracy. Compared with the two-source digital scheme in high SNR scenarios, the CL-SC scheme demonstrates superior performance, whereas the previous JSCC schemes cannot outperform. Compared with the NOMA JSCC benchmark from [86], we can observe that the proposed CL-SC scheme can

significantly improve the top-1 accuracy for all channel SNRs, especially in the low SNR regime, where the CL-SC can improve the top-1 accuracy up to 4.41% and 8.38% for the AWGN and fading channels, respectively. This observation shows that incorporating multi-source correlations with the CL mechanism can significantly improve remote retrieval performance over a wide range of SNRs, especially in bad channel conditions. We also observe that the CL-based scheme can maintain an improvement of 2.12% at high SNRs, showing the superiority of CL in the multi-view inference problem. Comparing the performance gain over the AWGN channel and fading channels in Fig. 6.7a and Fig. 6.7c, respectively, it is observed that the CL-SC yields more significant performance improvements in the fading channel. This can be attributed to the fact that the fading channel is more challenging, which introduces greater distortions during the transmission disturbance phase, where the additional optimization provided by the CL module is expected to bring more improvements.

In Figs. 6.7b and 6.7d, the proposed CL-SC scheme outperforms the NOMA JSCC scheme for all bandwidth values with consistent improvements up to 5.75% and 11.77% for AWGN and fading channels, respectively. In particular, the CL-SC scheme significantly improves the performance in the low bandwidth regimes, such as $k = 16, 32, 64, 128$. We emphasize that accuracy over 48% can be achieved for a blocklength of slightly higher than $k = 16$, which is extremely short for channel coding, and reliable communication at $= 0\text{dB}$ would not be possible. This shows that JSCC is essential for latency-constrained inference at the edge [4].

In comparison to its performance in OMA transmission, as shown in Fig. 6.8, the CL-SC demonstrates superior efficacy in NOMA transmission, achieving a performance enhancement of up to 15.74% and 17.85% for AWGN and slow fading channels, respectively. The NOMA transmission scheme facilitates the transmission of features by leveraging the superposition property, promoting feature overlap. Subsequently, the CL-SC utilizes the contrastive learning method to optimize the overlapped features to be more compact and discriminative, aiming at improving inference performance.

In summary, our CL-SC scheme achieves SOTA performance in both AWGN and fading channels with the NOMA transmission under various SNRs and bandwidth values. We can conclude

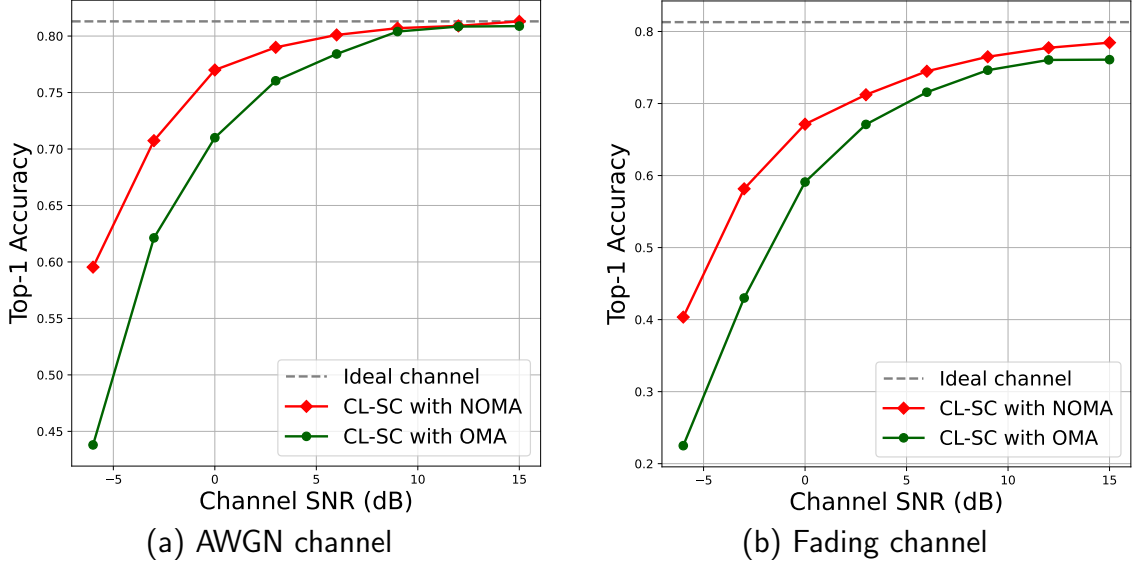


Figure 6.8: Performance comparison between OMA and NOMA schemes for the CL-SC.

that exploiting contrastive learning with the feature-over-the-air design can significantly benefit edge inference tasks, especially in lousy channel conditions and limited resources.

6.5.4 CL-SC scheme for $M=3$

This section outlines the experiments conducted to evaluate the performance of the CL-SC scheme over an AWGN channel with parameters $M = 3$ and $B = 60$. The results are elaborated in Table 6.1. Notably, to ensure a balanced dataset for evaluation, specific camera indices were selected: $[0, 1, 2]$, $[0, 2, 4]$, $[1, 3, 5]$, and $[3, 4, 5]$. We can observe consistent performance improvements from the proposed CL-SC scheme across both OMA and NOMA transmission strategies. This observation is especially pronounced in low SNR scenarios. Compared to $M = 2$, the performance improvement of CL-SC with NOMA over multi-source JSCC can be more pronounced, highlighting the effectiveness of the CL scheme across multiple viewpoints. For larger numbers of transmitters, performance saturation is expected; however, this could not be fully validated due to the limited availability of datasets with more than three views. Given the additional resource requirements and the clear performance advantages already observed, we suggest that $M = 2$ or 3 is sufficient in most cases. These findings underscore the potential of our proposed CL-SC scheme to be applied in broader distributed semantic communication

scenarios.

Transmission	Models	-3 dB	0 dB	3 dB
OMA	multi-source JSCC	40.90%	53.78%	61.11%
	CL-SC	45.95%	59.34%	66.41%
NOMA	multi-source JSCC	52.27%	62.88%	66.78%
	CL-SC	59.35%	66.67%	69.94%

Table 6.1: Top1-accuracy of different schemes across various SNR values, where $B = 60$ and $M = 3$.

6.5.5 Result visualization

To comprehensively demonstrate the effect of the proposed CL module in optimizing more discriminative query features in the feature space, Fig. 6.9 presents the t-SNE visualization [178] of the feature spaces for two-source JSCC and CL-SC schemes with NOMA transmission, where $B = 64$ and $SNR = 0$ dB. Each distinct color corresponds to a unique class identity, selected from the dataset. Specifically, Fig. 6.9a and Fig. 6.9b depict the feature spaces of the original two-source JSCC scheme and the proposed CL-SC scheme, respectively.

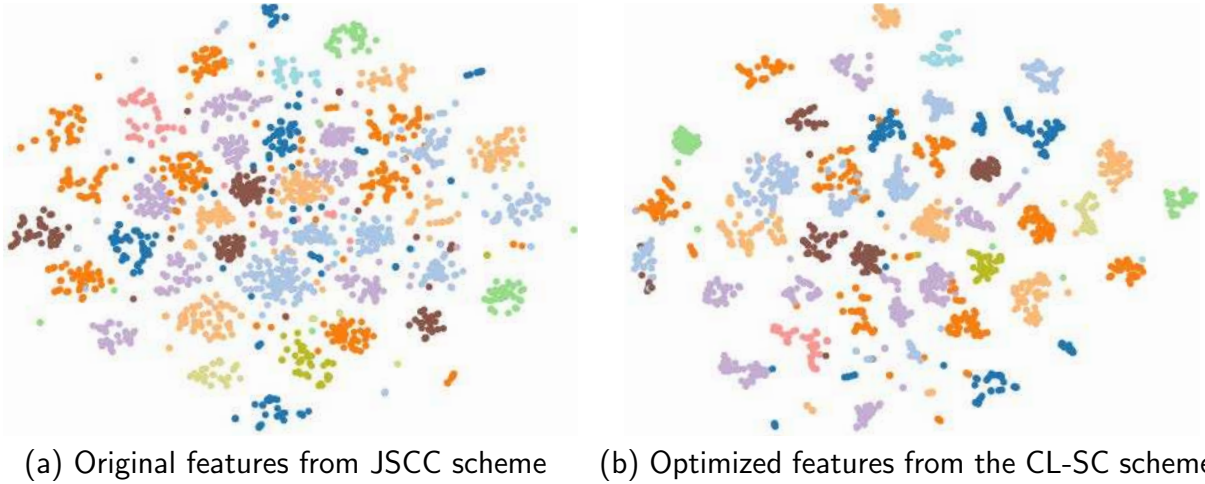


Figure 6.9: Visualization of the features from two-source JSCC and CL-SC module, where $B = 64$ and $SNR = 0$ dB.

From Fig. 6.9a, it becomes apparent that the features acquired through the two-source JSCC scheme lack discriminative and representative qualities. The distance between samples of the same class is notably distant, and the proximity of samples from different classes often leads to

overlap. In contrast, Figure 6.9b clearly illustrates that the CL-SC scheme has optimized and learned discriminative features from the additional contrastive learning module. The distances between samples of the same class are minimal, while those between samples of different classes are greater. Moreover, there are fewer instances of overlapping samples from distinct classes, compared with the results in Fig. 6.9a. This observation emphasizes that the CL-SC scheme can effectively enhance retrieval performance by learning discriminative features.

6.6 Conclusion

In this chapter, we introduced a novel contrastive learning-based semantic communication framework designed to address collaborative edge retrieval challenges. Our approach leverages signal correlations effectively and optimizes the transmission of features to enhance retrieval accuracy within the constraints of available bandwidth and SNRs. The proposed CL-SC framework is applicable to both OMA and NOMA transmission schemes and benefits significantly from the integration of JSCC with over-the-air technologies. Through comprehensive numerical experiments, we demonstrated that incorporating CL can significantly boost retrieval performance across various channel conditions and bandwidths in both AWGN and slow-fading channels, particularly under low SNRs and restricted bandwidth scenarios. The versatility of our proposed framework makes it suited for any remote communication context where exploiting multiple correlated signals is critical.

6.6.1 Future work

Based on the findings of this chapter, there are several potential avenues for future research:

- Contrastive JSCC coding [160]. Although our CL-SC scheme is designed with an efficient cross-view CL fashion, alternative JSCC schemes based on the CL algorithm also exist [160]. Future research could explore alternative CL algorithms incorporating negative samples for collaborative edge inference tasks, especially the trade-off between per-

formance improvement and the additional training complexity introduced by negative samples.

- Multimodal contrastive learning. The success of CL in multimodal scenarios [179] has paved the way for practical multimodal semantic systems. Our proposed CL-SC framework can be seamlessly integrated with recently popular pre-trained large language models, such as CLIP [179], enabling the development of a versatile CL-enabled semantic communication system capable of accommodating diverse modalities and different edge tasks.

Chapter 7

Conclusion

Throughout this thesis, we comprehensively explored DeepJSCC, a pioneering technology that fundamentally challenges and extends beyond the traditional paradigms established by Shannon’s separation theorem. By integrating source and channel coding into a unified framework facilitated by deep learning, DeepJSCC heralds a new era of communication systems characterized by enhanced efficiency, adaptability, and resilience.

The journey through the realms of OFDM, MIMO, feedback, broadcasting, and multiple access channels, encompassing tasks from source transmission to edge inference, has highlighted the versatility and transformative potential of DeepJSCC across diverse semantic communication scenarios. Each of these scenarios presented unique challenges, serving as a testament to the limitations of conventional communication systems and underscoring the necessity for innovative approaches like DeepJSCC. It is particularly noteworthy that, in several cases, the proposed DeepJSCC method significantly outperforms the theoretical upper bounds of specific traditional separation-based schemes, even when capacity-achieving channel codes are assumed. Moreover, DeepJSCC addresses critical research gaps highlighted in previous studies by providing practical and effective coding solutions. For instance, despite substantial progress in channel coding, practical implementations for feedback channels remain underexplored, whereas DeepJSCC presents a competitive alternative. We believe that the flexibility offered by DeepJSCC, and its data-driven nature make it a promising alternative for 6G and future communication

networks.

In conclusion, DeepJSCC stands out as a fundamental innovation in the field of semantic communications, extending the boundaries of what is achievable with JSCC. Its importance is underscored not only by its challenge to theoretical limits but also by its ability to make JSCC practical and applicable across a diverse range of sources, channels, and application scenarios. As research continues to push these frontiers, DeepJSCC presents a promising avenue toward developing highly efficient and reliable wireless communication systems, which can be adaptable to the increasing requirements and constraints imposed by emerging applications. This thesis has established a foundation for further exploration and application of DeepJSCC, demonstrating its potential to optimize the design of semantic communication systems. These findings pave the way for future advancements and position DeepJSCC as a promising solution for the next generation of communication systems.

7.1 Future work

- **Explicit exploitation of source semantic relevance.**

In the development of DeepJSCC for semantic communication, channel characteristics play a pivotal role and form the central focus of this thesis. Additionally, the attributes of the source can be explicitly integrated into the design of more advanced DeepJSCC systems. For example, incorporating innovations such as DeepJSCC with variable code lengths optimized for different source instances and exploiting correlations between different semantic objects in the image or video can significantly improve coding efficiency.

- **New communication paradigms arising from emerging technologies.**

While the DeepJSCC frameworks proposed in this thesis can be plugins for various semantic communication paradigms, the advent of new technologies, such as large language models and prompt engineering, can introduce new challenges in the future. For instance, token-based communication may become a new transmission unit, and optimizing prompts in conjunction with DeepJSCC can be a promising avenue for future research.

More recently, overfitted compression approaches, including our own work [7, 61], have introduced new paradigms to address source generalizability while reducing decoding complexity, offering new directions for both source coding and DeepJSCC schemes.

- **Theoretical basis of semantic communication.**

Although this thesis presents several practical DeepJSCC frameworks for semantic communication, there is still no established theoretical foundation for semantic communication. A well-developed theory could guide future research directions, provide valuable bounds, and enhance interpretability for the system. For instance, in source delivery tasks, a theoretical framework could provide a metric to quantify how much semantic information is preserved.

- **New communication paradigm involves interactions**

This thesis provides a practical feedback coding method involving interactions between transmitter and receiver. However, the current modern communication system has not yet developed an interactive mechanism between different transceivers, which can refine and optimize the communication objectives. This topic involves the design of active/passive feedback coding, a cooperative algorithm between transceivers, and efficient bandwidth resource allocation for downlink and uplink transmissions.

- **New communication paradigm without explicit channels**

Communication is crucial for coordination in multi-agent systems, but explicit channels may be unavailable, insecure, or unreliable. Our recent work [180] explores joint task and implicit communication protocol optimization, enabling communication via agent actions. This new paradigm is particularly well-suited to JSCC technologies due to their low-bandwidth constraints, and can serve as a plug-in for DeepJSCC-based semantic communication by integrating semantics into task execution and framing it as a joint source-channel coding problem.

- **Multi-modality semantic communication**

Future wireless systems are expected to increasingly support multimodal tasks and transmission scenarios. Integrating multiple modalities into these systems will be essential for improving both network efficiency and robustness. Semantic communication, capable of seamlessly handling various modalities such as text, audio, and video, will be developed to exploit their inherent correlations to reduce bandwidth consumption and enhance resilience to noise.

Bibliography

- [1] L. Chettri and R. Bera, “A comprehensive survey on internet of things (IoT) toward 5G wireless systems,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.
- [2] M. Chen, W. Saad, and C. Yin, “Virtual reality over wireless networks: Quality-of-service model and learning-based resource management,” *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5621–5635, 2018.
- [3] M. Xing, S. Xiang, and L. Cai, “A real-time adaptive algorithm for video streaming over multiple wireless access networks,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 795–805, 2014.
- [4] D. Gündüz, D. B. Kurka, M. Jankowski, M. M. Amiri, E. Ozfatura, and S. Sreekumar, “Communicate to learn at the edge,” *IEEE Communications Magazine*, vol. 58, no. 12, pp. 14–19, 2020.
- [5] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: <https://openreview.net/forum?id=rkcQFMZRb>
- [6] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, “Overview of the versatile video coding (vvc) standard and its applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [7] H. Wu, G. Chen, P. L. Dragotti, and D. Gunduz, “Lotterycodec: Searching the implicit

- representation in a random network for low-complexity image compression,” in *Forty-Second International Conference on Machine Learning*, 2025.
- [8] 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.
- [9] M. Rowshan, M. Qiu, Y. Xie, X. Gu, and J. Yuan, “Channel coding towards 6G: Technical overview and outlook,” *IEEE Open Journal of the Communications Society*, 2024.
- [10] J. Shao, Y. Mao, and J. Zhang, “Learning task-oriented communication for edge inference: An information bottleneck approach,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 197–211, 2021.
- [11] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [12] S. Vembu, S. Verdu, and Y. Steinberg, “The source-channel separation theorem revisited,” *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 44–54, 1995.
- [13] A. Goldsmith, “Joint source/channel coding for wireless channels,” in *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, vol. 2. IEEE, 1995, pp. 614–618.
- [14] 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.
- [15] E. Bourtsoulatze, D. Burth 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.
- [16] Y. Shao, Q. Cao, and D. Gündüz, “A theory of semantic communication,” *IEEE Transactions on Mobile Computing*, 2024.

- [17] G. M. Davis and J. M. Danskin, “Joint source and channel coding for image transmission over lossy packet networks,” *Proceedings of SPIE - The International Society for Optical Engineering*, May 1999.
- [18] 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.
- [19] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, “Joint source-channel coding for video communications,” in *Handbook of Image and Video Processing*, 2005.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] M. Fresia, F. Pérez-Cruz, H. V. Poor, and S. Verdú, “Joint source and channel coding,” *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 104–113, 2010.

- [27] 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.
- [28] 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.
- [29] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.
- [30] 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.
- [31] E. Ozfatura, Y. Shao, A. Ghazanfari, A. Perotti, B. Popovic, and D. Gunduz, "Feedback is good, active feedback is better: Block attention active feedback codes," in *IEEE International Conference on Communications (ICC)*, 2023.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

- [36] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, “Neural joint source-channel coding,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 1182–1192.
- [37] Y. M. Saidutta, A. Abdi, and F. Fekri, “Joint source-channel coding over additive noise analog channels using mixture of variational autoencoders,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2000–2013, 2021.
- [38] E. Erdemir, P. L. Dragotti, and D. Gündüz, “Privacy-aware communication over a wiretap channel with generative networks,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 2989–2993.
- [39] “Better portable graphics 0.9.8,” URL <https://bellard.org/bpg/>, 2018.
- [40] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.

- [45] 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.
- [46] S. F. Yilmaz, C. Karamanlı, and D. Gündüz, “Distributed deep joint source-channel coding over a multiple access channel,” in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 1400–1405.
- [47] 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.
- [48] T.-Y. Tung and D. Gündüz, “Deepwive: Deep-learning-aided wireless video transmission,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2570–2583, 2022.
- [49] Z. Weng and Z. Qin, “Semantic communication systems for speech transmission,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.
- [50] 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.
- [51] H. Wu, N. Mital, K. Mikolajczyk, and D. Gündüz, “Features-over-the-air: Contrastive learning enabled cooperative edge inference,” *arXiv preprint arXiv:2304.08221*, 2023.
- [52] W. Weaver, “Recent contributions to the mathematical theory of communication,” *ETC: a review of general semantics*, pp. 261–281, 1953.
- [53] J. Hu, F. Wang, W. Xu, H. Gao, and P. Zhang, “Semharq: Semantic-aware HARQ for multi-task semantic communications,” *arXiv preprint arXiv:2404.08490*, 2024.
- [54] H. Xie and Z. Qin, “A lite distributed semantic communication system for internet of

- things,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2021.
- [55] P. Zhang, W. Xu, H. Gao, K. Niu, X. Xu, X. Qin, C. Yuan, Z. Qin, H. Zhao, J. Wei *et al.*, “Toward wisdom-evolutionary and primitive-concise 6G: A new paradigm of semantic communication networks,” *Engineering*, vol. 8, pp. 60–73, 2022.
- [56] P. Yi, Y. Cao, X. Kang, and Y.-C. Liang, “Deep learning-empowered semantic communication systems with a shared knowledge base,” *IEEE Transactions on Wireless Communications*, 2023.
- [57] M. Jankowski, D. Gündüz, and K. Mikolajczyk, “Wireless image retrieval at the edge,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 89–100, 2021.
- [58] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet, “Coin: Compression with implicit neural representations,” *ICLR 2021-International Conference on Learning Representations Workshop Neural Compression 2021*, 2021.
- [59] T. Ladune, P. Philippe, F. Henry, G. Clare, and T. Leguay, “Cool-chic: Coordinate-based low complexity hierarchical image codec,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 515–13 522.
- [60] H. Kim, M. Bauer, L. Theis, J. R. Schwarz, and E. Dupont, “C3: High-performance and low-complexity neural compression from a single image or video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9347–9358.
- [61] H. Wu, M. Zhang, Y. Shao, K. Mikolajczyk, and D. Gündüz, “MIMO channel as a neural function: Implicit neural representations for extreme CSI compression,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [62] I. Csiszár and J. Körner, *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

- [63] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, “An overview of JPEG-2000,” in *Proceedings DCC 2000. Data compression conference*. IEEE, 2000, pp. 523–541.
- [64] 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.
- [65] “Versatile video coding reference software version 12.1 (vtm-12.1).” https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM, vol. 6, 2021.
- [66] 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.
- [67] 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.
- [68] D. Divsalar and F. Pollara, “On the design of turbo codes,” *The Telecommunications and Data Acquisition Progress Report 42-123*, 1995.
- [69] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [70] L. Kondi, F. Ishtiaq, and A. Katsaggelos, “Joint source-channel coding for motion-compensated DCT-based SNR scalable video,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1043–1052, 2002.
- [71] 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.

- [72] N. Phamdo and U. Mittal, “A joint source-channel speech coder using hybrid digital-analog (HDA) modulation,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 4, pp. 222–231, 2002.
- [73] T. Holliday, A. J. Goldsmith, and H. V. Poor, “Joint source and channel coding for MIMO systems: Is it better to be robust or quick?” *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1393–1405, 2008.
- [74] O. Fresnedo, F. J. Vazquez-Araujo, L. Castedo, and J. Garcia-Frias, “Analog joint source-channel coding for OFDM systems,” in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2013, pp. 704–708.
- [75] V. Kafedziski and D. Morrell, “Joint source-channel coding over frequency selective fading channels with feedback using OFDM,” in *1997 IEEE 47th Vehicular Technology Conference. Technology in Motion*, vol. 3. IEEE, 1997, pp. 1390–1394.
- [76] E. Köken and E. Tuncel, “Joint source-channel coding for broadcasting correlated sources,” *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 3012–3022, 2017.
- [77] T. P. Coleman, E. Martinian, and E. Ordentlich, “Joint source-channel coding for transmitting correlated sources over broadcast networks,” *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3864–3868, 2009.
- [78] V. Kostina, Y. Polyanskiy, and S. Verd, “Joint source-channel coding with feedback,” *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3502–3515, 2017.
- [79] D. Gunduz and E. Erkip, “Source and channel coding for cooperative relaying,” *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3454–3475, 2007.
- [80] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
- [81] M. Ruf and J. Modestino, “Operational rate-distortion performance for joint source and channel coding of images,” *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 305–320, 1999.

- [82] I. Kozintsev and K. Ramchandran, “Robust image transmission over energy-constrained time-varying channels using multiresolution joint source-channel coding,” *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1012–1026, 1998.
- [83] V. Stankovic, R. Hamzaoui, Y. Charfi, and Z. Xiong, “Real-time unequal error protection algorithms for progressive image transmission,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1526–1535, 2003.
- [84] 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.
- [85] 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.
- [86] 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.
- [87] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, “Task-oriented multi-user semantic communications,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, 2022.
- [88] 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. [Online]. Available: <https://cmsworkshops.com/ICASSP2020/Papers/ViewPaper.asp?PaperNum=6212>
- [89] 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.

- [90] 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.
- [91] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, “Deep learning enabled semantic communication systems,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.
- [92] C. Bian, Y. Shao, and D. Gunduz, “Wireless point cloud transmission,” *arXiv preprint arXiv:2306.08730*, 2023.
- [93] R. Mao, H. Wu, Y. Jia, Z. Nan, Y. Sun, S. Zhou, D. Gündüz, and Z. Niu, “DiffCP: Ultra-low bit collaborative perception via diffusion model,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.
- [94] C. Shannon and W. Weaver, “The mathematical theory of communication,” *Urbana, IL, USA: Univ. Illinois Press*, 1949.
- [95] N. Wiener, “The human use of human beings,” *Urbana, IL, USA: Houghton Mifflin*, 1950.
- [96] C. E. Shannon *et al.*, “Coding theorems for a discrete source with a fidelity criterion,” *IRE Nat. Conv. Rec*, vol. 4, no. 142-163, p. 1, 1959.
- [97] E. Grassucci, C. Marinoni, A. Rodriguez, and D. Comminiello, “Diffusion models for audio semantic communication,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 13 136–13 140.
- [98] Y. Shao, Q. Cao, and D. Gündüz, “A theory of semantic communication,” *IEEE Transactions on Mobile Computing*, pp. 1–18, 2024.
- [99] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, “What is semantic communication? a view on conveying meaning in the era of machine intelligence,” *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, 2021.
- [100] E. C. Strinati and S. Barbarossa, “6G networks: Beyond shannon towards semantic and goal-oriented communications,” *Computer Networks*, vol. 190, p. 107930, 2021.

- [101] M. Kalfa, M. Gok, A. Atalik, B. Tegin, T. M. Duman, and O. Arikan, "Towards goal-oriented semantic signal processing: Applications and future challenges," *Digital Signal Processing*, vol. 119, p. 103134, 2021.
- [102] G. Shi, Y. Xiao, Y. Li, and X. Xie, "From semantic communication to semantic-aware networking: Model, architecture, and open problems," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 44–50, 2021.
- [103] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Communications Magazine*, vol. 59, no. 6, pp. 96–102, 2021.
- [104] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2021.
- [105] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.
- [106] M. Yang, C. Bian, and H.-S. Kim, "Deep joint source channel coding for wireless image transmission with OFDM," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.
- [107] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531–532, 1972.
- [108] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.
- [109] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>

- [110] J. Zhou, C. T. Leong, and C. Li, “Multi-scale and attention residual network for single image dehazing,” in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2021, pp. 483–487.
- [111] L. Sun, J. Liu, Y. Zhu, and Z. Jiang, “Local to global with multi-scale attention network for person re-identification,” in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 2254–2258.
- [112] L. Yu, J. Zhang, and Q. Wu, “Dual attention on pyramid feature maps for image captioning,” *IEEE Transactions on Multimedia*, vol. 24, pp. 1775–1786, 2022.
- [113] N. Samuel, T. Diskin, and A. Wiesel, “Learning to detect,” *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.
- [114] 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.
- [115] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Deep learning based MIMO communications,” *arXiv preprint arXiv:1707.07980*, 2017.
- [116] 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.
- [117] 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.
- [118] 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.
- [119] 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.

- [120] 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.
- [121] 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.
- [122] 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.
- [123] 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.
- [124] 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.
- [125] 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.
- [126] 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.
- [127] 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.
- [128] 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.

- [129] 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.
- [130] C. Oestges, “Validity of the kronecker model for MIMO correlated channels,” in *2006 IEEE 63rd Vehicular Technology Conference*, vol. 6. IEEE, 2006, pp. 2818–2822.
- [131] T. M. Cover, *Elements of information theory*. Hoboken, NJ, USA: Wiley, 1999.
- [132] C. Shannon, “The zero error capacity of a noisy channel,” *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 8–19, 1956.
- [133] R. L. Dobrushin, “Information transmission in a channel with feedback,” *Theory of Probability & Its Applications*, vol. 3, no. 4, pp. 367–383, 1958.
- [134] J. Schalkwijk, “A coding scheme for additive noise channels with feedback–II: Band-limited signals,” *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 183–189, 1966.
- [135] J. Schalkwijk and L. Bluestein, “Transmission of analog waveforms through channels with feedback (corresp.),” *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 617–619, 1967.
- [136] V. Chande, H. Jafarkhani, and N. Farvardin, “Joint source-channel coding of images for channels with feedback,” *Proc. IEEE Inf. Theory Workshop*, Feb. 1998.
- [137] J. Lu, A. Nosratinia, and B. Aazhang, “Progressive joint source-channel coding in feedback channels,” in *Proc. Data Compression Conf. (DCC)*, Mar. 1999, pp. 140–148.
- [138] V. Kafedziski, “Joint source channel coding of images over frequency selective fading channels with feedback using DCT and multicarrier block pulse amplitude modulation,” in *Proc. Asilomar Conf. Signals, Syst. and Comput.*, vol. 1, Nov. 1998, pp. 37–41 vol.1.
- [139] R. Puri, K. Ramchandran, and A. Ortega, “Joint source channel coding with hybrid FEC/ARQ for buffer constrained video transmission,” in *1998 IEEE Second Workshop on Multimedia Signal Processing (Cat. No. 98EX175)*. IEEE, 1998, pp. 567–572.

- [140] P. Chou, A. Mohr, A. Wang, and S. Mehrotra, “FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video,” in *Proceedings DCC 2000. Data Compression Conference*, 2000, pp. 440–449.
- [141] J. Chakareski, P. Chou, and B. Aazhang, “Computing rate-distortion optimized policies for streaming media to wireless clients,” in *Proceedings DCC 2002. Data Compression Conference*, 2002, pp. 53–62.
- [142] D. Taubman and J. Thie, “Optimal erasure protection for scalably compressed video streams with limited retransmission,” *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1006–1019, 2005.
- [143] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *Proc. ICLR*, 2021.
- [144] G. Shi, Y. Xiao, Y. Li, and X. Xie, “From semantic communication to semantic-aware networking: Model, architecture, and open problems,” *IEEE Communications Magazine*, vol. 59, no. 8, pp. 44–50, 2021.
- [145] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, “Deepcode: Feedback codes via deep learning,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 194–206, 2020.
- [146] T. Kailath, “An application of shannon’s rate-distortion theory to analog communication over feedback channels,” *Proceedings of the IEEE*, vol. 55, no. 6, pp. 1102–1103, 1967.
- [147] L. Ozarow and S. Leung-Yan-Cheong, “An achievable region and outer bound for the Gaussian broadcast channel with feedback,” *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 667–671, 1984.
- [148] T. M. Cover, “Comments on broadcast channels,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2524–2530, 1998.

- [149] C. E. Shannon, “Two-way communication channels,” in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, vol. 4. University of California Press, 1961, pp. 611–645.
- [150] T. Cover, A. Gamal, and M. Salehi, “Multiple access channels with arbitrarily correlated sources,” *IEEE Transactions on Information Theory*, vol. 26, no. 6, pp. 648–657, 1980.
- [151] A. Lapidoth and S. Tinguely, “Sending a bivariate gaussian over a gaussian mac,” *IEEE Trans. Info. Theory*, vol. 56, no. 6, pp. 2714–2752, 2010.
- [152] W. Zhang, Y. Wang, M. Chen, T. Luo, and D. Niyato, “Optimization of image transmission in a cooperative semantic communication networks,” *IEEE Transactions on Wireless Communications*, 2023.
- [153] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
- [154] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [155] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, “Bootstrap your own latent—a new approach to self-supervised learning,” *Advances in Neural Inf. Proc. Sys. (NeurIPS)*, vol. 33, pp. 21 271–21 284, 2020.
- [156] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo, “Detco: Unsupervised contrastive learning for object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8392–8401.
- [157] X. Zhao, R. Vemulapalli, P. A. Mansfield, B. Gong, B. Green, L. Shapira, and Y. Wu, “Contrastive learning for label efficient semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 623–10 633.

- [158] P. Wang, K. Han, X.-S. Wei, L. Zhang, and L. Wang, “Contrastive learning based hybrid networks for long-tailed image classification,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 943–952.
- [159] C. Chaccour and W. Saad, “Disentangling learnable and memorizable data via contrastive learning for semantic communications,” in *2022 56th Asilomar Conference on Signals, Systems, and Computers*, 2022, pp. 1175–1179.
- [160] S. Tang, Q. Yang, L. Fan, X. Lei, Y. Deng, and A. Nallanathan, “Contrastive learning based semantic communication for wireless image transmission,” in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*. IEEE, 2023, pp. 1–6.
- [161] R. Dobrushin and B. Tsybakov, “Information transmission with additional noise,” *IRE Transactions on Information Theory*, vol. 8, no. 5, pp. 293–304, 1962.
- [162] M. M. Amiri and D. Gündüz, “Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.
- [163] S. F. Yilmaz, B. Hasircioğlu, and D. Gündüz, “Over-the-air ensemble inference with model privacy,” in *IEEE Int’l Symp. Info. Theory*, 2022.
- [164] M. Krouka, A. Elgabli, C. b. Issaid, and M. Bennis, “Communication-efficient split learning based on analog communication and over the air aggregation,” in *IEEE GLOBECOM*, 2021.
- [165] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.
- [166] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, “Deep learning for person re-identification: A survey and outlook,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

- [167] F. Yang, K. Yan, S. Lu, H. Jia, X. Xie, and W. Gao, “Attention driven person re-identification,” *Pattern Recognition*, vol. 86, pp. 143–155, Feb 2019.
- [168] Y. Fu, Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao, and T. Huang, “Horizontal pyramid matching for person re-identification,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 8295–8302.
- [169] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8844–8854.
- [170] T.-W. Huang, J. Cai, H. Yang, H.-M. Hsu, and J.-N. Hwang, “Multi-view vehicle re-identification using temporal attention model and metadata re-ranking,” in *CVPR Workshops*, vol. 2, 2019, p. 3.
- [171] X. Liu, W. Liu, J. Zheng, C. Yan, and T. Mei, “Beyond the parts: Learning multi-view cross-part correlation for vehicle re-identification,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 907–915.
- [172] H. Li, C. Hu, J. Jiang, Z. Wang, Y. Wen, and W. Zhu, “JALAD: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution,” in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, 2018, pp. 671–678.
- [173] A. E. Eshratifar, A. Esmaili, and M. Pedram, “Bottlenet: A deep learning architecture for intelligent mobile cloud computing services,” in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2019, pp. 1–6.
- [174] J. Shao, Y. Mao, and J. Zhang, “Task-oriented communication for multi-device cooperative edge inference,” *IEEE Transactions on Wireless Communications*, 2022.
- [175] 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 (CVPR)*, Jun 2016, pp. 770–778.

- [176] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, “Completer: Incomplete multi-view clustering via contrastive prediction,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2021, pp. 11 174–11 183.
- [177] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [178] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE.” *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [179] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [180] H. Wu, G. Chen, and D. Gunduz, “Actions speak louder than words: Rate-reward trade-off in markov decision processes,” in *The Thirteenth International Conference on Learning Representations*, 2025.