Imperial College London
Department of Electrical and Electronic Engineering

Multi-User Semantic Communications

by

Selim Firat Yilmaz

September 2025

Supervised by
Prof. Deniz Gündüz

A thesis submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College London
and the Diploma of Imperial College London

Abstract

This thesis presents novel frameworks for multi-user semantic communications that integrate machine learning with wireless communications to enable efficient, private, and semantically aware information exchange at the wireless edge. Traditional communication systems, built on Shannon's separation principle, face growing limitations when dense networks of edge devices must share limited spectral resources to exchange high-dimensional data under strict latency constraints. In such settings, separate source and channel coding is known to be suboptimal in the finite blocklength regime, and conventional orthogonal access schemes fail to exploit the correlation structure across users' signals. We address these limitations by developing semantic communication systems for scalable multi-user communications and privacy-preserving edge inference.

We first address perceptual quality in wireless image transmission by integrating denoising diffusion probabilistic models into the deep joint source-channel coding (DeepJSCC) framework, achieving enhanced perceptual quality under severe channel conditions while establishing principled perceptual-distortion trade-offs.

We then extend distributed source coding theory to modern deep learning frameworks, focusing on low-latency image transmission with decoder-only side information. Our architectures integrate side information at multiple stages, delivering superior performance across all channel conditions, with particular gains at low signal-to-noise ratios.

Next, we develop non-orthogonal approaches to multi-user semantic communications over multiple access channels using DeepJSCC. Our scheme enables devices to simultaneously transmit compressed image representations, achieving substantial improvements over orthogonal transmission. Furthermore, we introduce a method that unifies compression and channel coding through multi-view autoencoders, enabling non-orthogonal multiple access (NOMA). Our algorithm scales to 64 users or more, surpassing state-of-the-art NOMA methods.

Finally, we introduce a privacy-preserving collaborative inference framework at the wireless edge, where clients' independently trained models participate in ensemble inference. Leveraging over-the-air computation, we develop schemes that exploit channel superposition for bandwidth-efficient

transmission. Our multi-view classification framework achieves statistically significant improvements over orthogonal approaches while ensuring robust privacy protection.

©

Copyright

Declaration of Originality

I, Selim Firat Yilmaz, declare that this thesis titled, 'Multi-User Semantic Communications' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at Imperial College of Science, Technology and Medicine.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed: Selim Firat Yilmaz

Date: September 2025

Acknowledgements

My deepest gratitude goes to my supervisor, Dr. Deniz Gündüz, whose exceptional guidance, expertise, and enthusiasm for research have been instrumental in shaping both this work and my development as a researcher. His mentorship has been an extraordinary privilege and constant source of inspiration.

I would like to express my sincere gratitude to Dr. Bruno Clerckx and Dr. Jakob Hoydis for serving as examiners of this thesis and for their valuable time, expertise, and constructive feedback that have significantly improved the quality of this work.

I am grateful to the GreenEdge Marie Skłodowska-Curie Innovative Training Network (Grant Agreement No. 953775) for providing the framework and financial support that made this research possible. Special thanks to my fellow GreenEdge researchers and supervisors, including but definitely not limited to Dr. Michele Rossi, Dr. Paolo Dini, and Dr. Marco Miozzo, whose guidance, collaboration, and friendship created an inspiring international research environment. I would also like to acknowledge Dr. Elza Erkip at New York University, Dr. Ayfer Özgür at Stanford University, and Dr. Xueyan Niu at Huawei Technologies R&D for their valuable insights and mentorship during my secondments and collaborations, which greatly enriched my research perspective. Furthermore, I extend my appreciation to all members of the Information Processing and Communications Lab for their invaluable friendship and support.

My heartfelt appreciation goes to my parents, Ömer and Kadriye, whose unconditional love, wisdom, and unwavering belief in my abilities have been the foundation of my academic journey. Their sacrifices and support have guided me through every challenge.

To my beloved wife, Tuğçe, thank you for your endless patience, understanding, and encouragement. Your presence has been my anchor, making this demanding journey not only bearable but truly meaningful. Your celebration of every small victory and your support through difficult times have meant everything to me.

This thesis represents not only my individual effort but the collective support and inspiration I have received from all these wonderful people and institutions. To each of them, I offer my sincere gratitude.

To my family

"If you ask me what makes me most happy, number one would be somebody saying 'I learned something from you.' Number two would be somebody saying 'I used your software.'"

Donald Knuth

Contents

List of Tables

List of Figures

List of Algorithms

Glossary

AF attention feature 38, 64, 71, 73

AWGN additive white Gaussian noise 18, 19, 36, 59, 60, 69, 71, 73, 85, 94, 108, 110, 112, 113, 117, 122, 130, 137, 167, 170

BA-OAC belief averaging with over-the-air computation (OAC) 138, 139, 161

BR bandwidth ratio 18, 69, 74, 75, 76

CD critical distance 161

CIS central inference server 133, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145

CL curriculum learning 54, 80, 85, 86, 87

CNN convolutional neural network 37, 63, 102

CSI channel state information 163, 167, 170

DDNM denoising diffusion null-space model 62

DDPM denoising diffusion probabilistic models 58, 61, 62, 64, 66

DeepJSCC deep joint source-channel coding 18, 33, 34, 37, 50, 57, 58, 59, 61, 63, 65, 66, 68, 69, 71, 74, 76, 79, 80, 81, 82, 83, 85, 86, 87, 89, 94, 96

DML deep metric learning 53

DNN deep neural network 46, 61, 71, 79, 89, 101, 119

DP differential privacy 19, 48, 49, 133, 134, 142, 152, 157, 158, 159

FDMA frequency division multiple access 42

FL federated learning 48, 133, 169

GAN generative adversarial network 58, 65, 66

GT ground truth 61

i.i.d. independent and identically distributed 36, 80, 92, 94, 137, 163

IoT Internet of Things 47, 78, 132, 163, 171

IR image restoration 61

JSCC joint source-channel coding 21, 33, 34, 35, 50, 57, 60, 67, 76, 78, 79, 91, 95, 108, 166, 167, 168, 170

LPIPS learned perceptual image patch similarity 17, 19, 41, 55, 56, 59, 64, 65, 72, 74, 95, 108, 109, 117, 130

MAC multiple access channel 17, 43, 45, 48, 78, 79, 80, 83, 91, 94, 95, 96, 100, 101, 109, 110, 113, 117, 134, 135, 144, 158, 160

MIMO multiple-input multiple-output 167

MS-SSIM multi-scale structural similarity index measure (SSIM) 19, 55, 56, 72, 74, 95, 108, 109, 117, 130

MSE mean squared error 38, 51, 63, 72

MTL multi-task learning 52, 99

MV-OAC majority voting with OAC 138, 139, 158, 161

NN neural network 63

NOMA non-orthogonal multiple access 15, 17, 26, 31, 33, 43, 44, 45, 46, 78, 80, 81, 88, 91, 93, 119

OAC over-the-air computation 23, 24, 28, 47, 48, 132, 134, 135, 138, 154, 155, 156, 160, 161, 162, 163, 164, 166

OMA orthogonal multiple access 42, 45

PSNR peak signal-to-noise ratio 17, 18, 54, 55, 56, 59, 64, 65, 71, 74, 81, 87, 94, 108, 116, 117, 118, 122

RR randomized response 158

SGD stochastic gradient descent 153

SIC successive interference cancellation 17, 43, 44, 45, 46, 86, 94, 95

SNR signal-to-noise ratio 17, 34, 36, 38, 54, 58, 61, 64, 65, 69, 71, 74, 75, 78, 79, 80, 81, 86, 87, 93, 94, 107, 108, 109, 118, 152, 156, 167, 168

SSIM structural similarity index measure 19, 24, 55, 56, 95, 117

SVD singular value decomposition 61

TDMA time division multiple access 17, 18, 42, 44, 45, 79, 87, 90, 91, 92, 93, 102, 107

WBA-OAC weighted belief averaging with OAC 138, 139, 161

Chapter 1

Introduction

1.1 Motivation and Objectives

The advent of the Internet of Things (IoT), mobile edge computing, and ubiquitous connectivity has transformed the landscape of wireless communications from simple data transmission to intelligent, semantic-aware information exchange. Traditional communication systems, designed around Shannon's separation principle [1], treat source compression and channel coding as independent optimization problems. While this approach has served well for decades, it increasingly falls short in meeting the demands of modern applications that require real-time processing, energy efficiency, and intelligent decision-making at the network edge.

The proliferation of machine learning applications in wireless environments has created new paradigms where the ultimate goal is not perfect bit-level reconstruction, but rather the preservation of semantic information relevant to downstream tasks [2, 3]. This shift from syntactic to semantic communication opens unprecedented opportunities for joint optimization of learning and communication processes. However, it also introduces significant challenges: even in the single-user case, reconstructions under severe channel conditions suffer from perceptual quality degradation, motivating the use of advanced generative models [4, 5].

As communication systems scale from single-user to multi-user scenarios, new challenges emerge

at each level of complexity. In distributed settings with correlated sources, exploiting decoder-only side information can substantially improve transmission efficiency, connecting to classical distributed source coding theory [6, 7]. When multiple users transmit simultaneously over shared channels, non-orthogonal multiple access (NOMA) techniques offer significant spectral efficiency gains over traditional orthogonal approaches, but their integration with semantic communication requires fundamentally new approaches to interference management [8, 9]. At the largest scale, collaborative intelligence across many distributed devices requires not only efficient communication but also privacy preservation, as devices must share sensitive information while protecting individual privacy [10, 11, 12].

Traditional approaches to privacy preservation often come at the cost of significant communication overhead or performance degradation, creating a fundamental tension between privacy, efficiency, and accuracy [13, 14]. Modern wireless applications increasingly require solutions that address all these dimensions simultaneously, from perceptual quality in point-to-point links to collaborative inference with privacy guarantees across many devices.

Objectives: In light of these challenges and opportunities, this thesis aims to develop novel multi-user semantic communication frameworks that address the following key objectives:

1. **Perceptual Quality Optimization:** Integrate advanced generative models into wireless communication systems to achieve superior perceptual quality in transmitted content, moving beyond traditional distortion-based optimization criteria.

2. **Distributed Source Coding with Side Information:** Extend classical distributed source coding principles to deep learning-based JSCC systems, particularly addressing scenarios where decoder-only side information can be exploited to improve transmission efficiency.

3. **Non-Orthogonal Semantic Communications:** Design and analyze NOMA schemes for semantic communications that outperform traditional orthogonal approaches in terms of spectral efficiency and reconstruction quality, particularly in resource-constrained environments.

4. **Adaptive Interference Management:** Develop learning-based approaches for strategic interference management in multi-user semantic communication systems, where interference can be leveraged as a beneficial feature rather than simply being treated as a detrimental effect.

5. **Privacy-Preserving Collaborative Inference:** Develop efficient protocols for collaborative inference in wireless networks that preserve privacy, leveraging the natural properties of wireless channels to provide formal privacy guarantees while maintaining high inference accuracy and communication efficiency.

These objectives collectively aim to establish a comprehensive framework for next-generation wireless communication systems that are intelligent, efficient, private, and semantically aware. By addressing these challenges, this thesis contributes to the foundation of future wireless networks that can support the growing demands of machine learning applications while ensuring privacy, efficiency, and high-quality user experiences.

1.2 Contributions

This thesis presents novel contributions to multi-user semantic communications, privacy-preserving distributed inference, and wireless machine learning. The technical chapters are organized along a progression of increasing multi-user complexity, from a single-user point-to-point link, through two-user distributed and multiple access settings, to large-scale multi-device collaborative inference, as detailed below:

Chapter 3 begins with the single-user point-to-point setting and introduces novel approaches to wireless image transmission that prioritize perceptual quality. We integrate denoising diffusion models into the DeepJSCC framework, achieving improved perceptual quality under severe channel conditions while establishing theoretical frameworks for perceptual-distortion trade-offs and demonstrating robustness to channel impairments.

Chapter 4 extends the single-user setting to a two-user distributed scenario, bridging classical

distributed source coding theory with modern deep learning frameworks. We develop the first DeepJSCC architecture that effectively exploits decoder-only side information, extending the Wyner-Ziv problem to the deep learning domain. We propose architectural variants including multi-scale feature fusion at the decoder, a parameter-efficient shared-encoder design, and an upper bound with side information at both ends, demonstrating significant gains over both point-to-point DeepJSCC and separation-based schemes across a range of channel conditions and fidelity metrics.

Chapters 5 and 6 move to settings where multiple users transmit simultaneously over a shared channel, presenting non-orthogonal approaches to multi-user semantic communications. Chapter 5 introduces a NOMA framework for two-user DeepJSCC over a multiple access channel, demonstrating performance gains over separation-based approaches in practical finite blocklength regimes. Chapter 6 extends this to many users with learning-based interference management algorithms that enable encoders to strategically leverage interference patterns, demonstrated with up to 64 users in our experiments, while establishing foundations for joint source-channel-multiple access optimization.

Chapter 7 concludes the multi-user progression in this thesis by addressing collaborative inference across many distributed devices. It introduces a privacy-preserving collaborative inference framework leveraging over-the-air computation (OAC) for ensemble inference, developing novel schemes that exploit wireless channel superposition properties for bandwidth-efficient transmission while providing natural differential privacy guarantees through channel noise.

Throughout this thesis, we provide rigorous theoretical foundations including information-theoretic analysis establishing fundamental limits and performance bounds, formal privacy analysis with differential privacy guarantees, convergence and stability analysis of learning algorithms, and comprehensive computational and communication complexity analysis.

These contributions collectively advance the development of next-generation semantic communication systems that are efficient, private, and intelligent. Our work challenges fundamental assumptions in wireless communication design, demonstrating, for example, that interference can be harnessed as a beneficial feature, and that privacy and efficiency can be compatible objectives when

leveraging the natural properties of wireless channels. The proposed techniques, validated through extensive experiments, pave the way for advanced wireless applications in the era of ubiquitous machine learning, enabling collaborative intelligence while respecting privacy constraints and resource limitations.

1.3 Associated Publications

Parts of the work presented in this thesis, along with other research conducted during my PhD, have been published in the following journals, conferences, and preprints, listed in chronological order. Publications J3, C1, C2, C4, C5, and P1, where I am the sole first author, constitute the core contributions of this thesis.

Journal Publications:

J1 F. Wilhelmi, J. Hribar, **S. F. Yilmaz**, E. Ozfatura, K. Ozfatura, O. Yildiz, D. Gündüz, H. Chen, X. Ye, L. You, Y. Shao, P. Dini, B. Bellalta, "Federated Spatial Reuse Optimization in Next-Generation Decentralized IEEE 802.11 WLANs," **2nd ITU Journal on Future and Evolving Technologies (ITU J-FET) on AI/ML Solutions in 5G**, 2022. `https://itu.int/pub/S-JNL-VOL3.ISSUE2-2022-A11` [15]

J2 N. Pavlidis, V. Perifanis, **S. F. Yilmaz**, F. Wilhelmi, M. Miozzo, P. S. Efraimidis, R.-A. Koutsiamanis, P. Mulinka and P. Dini, "Federated Learning in Mobile Networks: A Comprehensive Case Study on Traffic Forecasting," **IEEE Transactions on Sustainable Computing**, 2024. `https://ieeexplore.ieee.org/abstract/document/10759814/` [16]

J3 **S. F. Yilmaz**, B. Hasircioglu, L. Qiao and D. Gündüz, "Private Collaborative Edge Inference via Over-the-Air Computation," **IEEE Transactions on Machine Learning in Communications and Networking**, 2025. `https://ieeexplore.ieee.org/document/10829586/` [17]

Conference Publications:

C1 **S. F. Yilmaz**, B. Hasırcıoğlu, D. Gündüz, "Over-the-Air Ensemble Inference with Model Privacy," **IEEE International Symposium on Information Theory (ISIT)**, 2022. `https://ieeexplore.ieee.org/document/9834591` [18]

C2 **S. F. Yilmaz**, C. Karamanlı, D. Gündüz, "Distributed Deep Joint Source-Channel Coding over a Multiple Access Channel," **IEEE International Conference on Communications (ICC)**, 2023. `https://arxiv.org/abs/2211.09920` [19]

C3 V. Perifanis, N. Pavlidis, **S. F. Yilmaz**, F. Wilhelmi, E. Guerra, M. Miozzo, P. S. Efraimidis, P. Dini, R.-A. Koutsiamanis, "Towards Energy-Aware Federated Traffic Prediction for Cellular Networks," **International Symposium on Federated Learning Technologies and Applications**, 2023. `https://arxiv.org/pdf/2309.10645.pdf` [20]

C4 **S. F. Yilmaz**, E. Ozyilkan, D. Gündüz, E. Erkip, "Distributed Deep Joint Source-Channel Coding with Decoder-Only Side Information," **IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)**, 2024. `https://arxiv.org/abs/2310.04311` [21]

C5 **S. F. Yilmaz**, X. Niu, B. Bai, W. Han, L. Deng and D. Gündüz, "High Perceptual Quality Wireless Image Delivery with Denoising Diffusion Models," **IEEE INFOCOM Workshops: Deep Learning for Wireless Communications, Sensing, and Security** (DeepWireless), 2024. `https://arxiv.org/pdf/2309.15889.pdf` [22]

Preprints:

P1 **S. F. Yilmaz**, C. Karamanli, D. Gündüz, "Learning to Interfere in Non-Orthogonal Multiple Access Joint Source-Channel Coding," 2025. `https://arxiv.org/abs/2504.03690` (revised) [23]

P2 J. Chen*, **S. F. Yilmaz***, D. You*, P. L. Dragotti, D. Gündüz, "SING: Semantic Image Communications using Null-Space and INN-Guided Diffusion Models," 2025. `https://arxiv.org/abs/2503.12484` (under review) [24]

P3 A. Kurmukova, **S. F. Yilmaz**, E. Özfatura, D. Gündüz, "TransCoder: Code-Adaptive Neural Channel Modulation," 2025. `https://arxiv.org/abs/2511.22539` (under review) [25]

* indicates equal contribution.

1.4 Thesis Organization

This thesis is organized as follows. This chapter concludes with the common notation (Section 1.5). The technical chapters progress from single-user to increasingly complex multi-user settings:

- **Chapter 2: Background and Related Work.** Provides comprehensive theoretical foundations covering semantic communication, DeepJSCC, distributed inference, NOMA theory, Wyner-Ziv coding, deep learning paradigms, and privacy mechanisms in distributed systems.
- **Chapter 3: High Perceptual Quality Wireless Image Delivery with Denoising Diffusion Models.** Addresses the single-user point-to-point setting, integrating diffusion models into DeepJSCC for superior perceptual quality transmission under severe channel conditions. Based on [C5].
- **Chapter 4: Distributed Deep Joint Source-Channel Coding with Decoder-Only Side Information.** Extends to a two-user distributed scenario where one source transmits and a correlated second source serves as decoder-only side information, applying Wyner-Ziv theory to DeepJSCC frameworks. Based on [C4].
- **Chapter 5: Distributed Deep Joint Source-Channel Coding over a Multiple Access Channel.** Considers two users transmitting simultaneously, presenting a NOMA framework for multi-user image transmission that demonstrates significant gains over separation-based approaches in finite blocklength regimes. Based on [C2].
- **Chapter 6: Learning to Interfere in Non-Orthogonal Multiple-Access Joint Source-Channel Coding.** Scales to many users (demonstrated with up to 64 in our

experiments), extending NOMA-DeepJSCC with learning-based interference management where encoders strategically create beneficial interference patterns. Based on [P1].

- **Chapter 7: Private Collaborative Edge Inference via Over-the-Air Computation.** Addresses large-scale collaborative inference across many distributed devices, introducing privacy-preserving ensemble and multi-view classification with differential privacy guarantees via over-the-air computation. Based on [J3].

- **Chapter 8: Conclusion.** Summarizes key contributions and outlines future research directions in multi-user semantic communications, privacy-preserving distributed inference, and perceptual quality optimization.

1.5 Notation

Unless stated otherwise, we adopt the following notational conventions throughout this thesis: boldface lowercase letters denote vectors and tensors (e.g., $\mathbf{p}$), boldface uppercase letters denote matrices (e.g., $\mathbf{P}$), non-boldface letters denote scalars (e.g., p or P), and uppercase calligraphic letters denote sets (e.g., $\mathcal{P}$). Blackboard bold letters denote function domains and number sets (e.g., $\mathbb{P}$). $\mathbb{R}$, $\mathbb{N}$, $\mathbb{C}$, and $\mathbb{Z}$ denote the sets of real, natural, complex, and integer numbers, respectively. $|\mathcal{P}|$ denotes the cardinality of set $\mathcal{P}$. We define $[n] \triangleq \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}^+$, and $[i, j] \triangleq \{i, i+1, \ldots, j\}$ for $i, j \in \mathbb{Z}$ and $i \leq j$. We define $\mathbb{I} \triangleq [255]$ for pixel intensity values. For a u-dimensional complex vector $\mathbf{p} \in \mathbb{C}^u$, $\mathbf{p}^H$ denotes its Hermitian transpose, and $\mathbf{I}_u$ denotes the $u \times u$ identity matrix for $u \in \mathbb{N}^+$.

Chapter 2

Background and Related Work

This chapter reviews key literature that underpins the multi-user semantic communication frameworks developed in this thesis. Building on the motivation and objectives in Chapter 1, we explore the convergence of deep learning and wireless communications that enables our novel approaches.

We begin with an overview of semantic communication and joint source-channel coding (JSCC) principles, including the common deep joint source-channel coding (DeepJSCC) neural network architecture employed throughout this thesis (Section 2.1). This is followed by generative models for perceptual quality-oriented communication (Section 2.2), distributed source coding for correlated image transmission (Section 2.3), NOMA schemes for multi-user frameworks (Section 2.4), distributed inference systems and privacy considerations for edge computing (Section 2.5), and deep learning techniques used across our multi-user systems (Section 2.6). We conclude with the image quality evaluation metrics used throughout this thesis (Section 2.7).

2.1 Semantic Communication and DeepJSCC

Traditional communication systems follow Shannon's separation theorem [1], which establishes that source coding and channel coding can be optimized separately without loss of optimality in

the asymptotic regime with infinite blocklengths. However, in practical scenarios characterized by finite blocklengths, stringent latency constraints, and dynamic channel conditions, joint optimization of source and channel coding can provide significant performance gains. This limitation of the separation theorem in finite blocklength regimes has been a well-known challenge in information theory for several decades [26], yet no practically feasible scheme achieving performance even close to the theoretical separation bound has been developed until recent advances in deep learning. This longstanding gap has motivated the development of JSCC approaches.

Semantic communication represents a paradigm shift from bit-perfect transmission to task-oriented communication. Rather than focusing on exact symbol reconstruction, semantic communication systems are designed to meet new requirements where the reconstruction alphabet differs from the source alphabet. This approach is particularly valuable for applications where the end goal is task completion (e.g., classification, object detection) rather than perfect reconstruction, and these requirements can be considered as special distortion measures that prioritize task-relevant information over pixel-level fidelity.

DeepJSCC [2] represents a significant advancement in this direction, employing deep neural networks to jointly optimize source compression and channel coding in an end-to-end manner. This data-driven approach has demonstrated superior performance compared to traditional Separation methods, particularly for image transmission tasks [27, 19]. The key insight behind DeepJSCC is that neural networks can learn task-specific representations that are inherently robust to channel impairments, achieving graceful degradation with varying channel quality while preserving semantically important features.

The advantages of DeepJSCC become particularly pronounced under challenging conditions: (1) low signal-to-noise ratio (SNR) environments where traditional systems experience cliff effects, (2) bandwidth-constrained scenarios where efficient compression is crucial, and (3) time-varying channels where adaptive representations provide robustness. DeepJSCC has been successfully extended to various scenarios including different channel models [27], multi-modal sources [28, 29], inference-oriented transmission [30], and multi-user scenarios [19, 31]. The

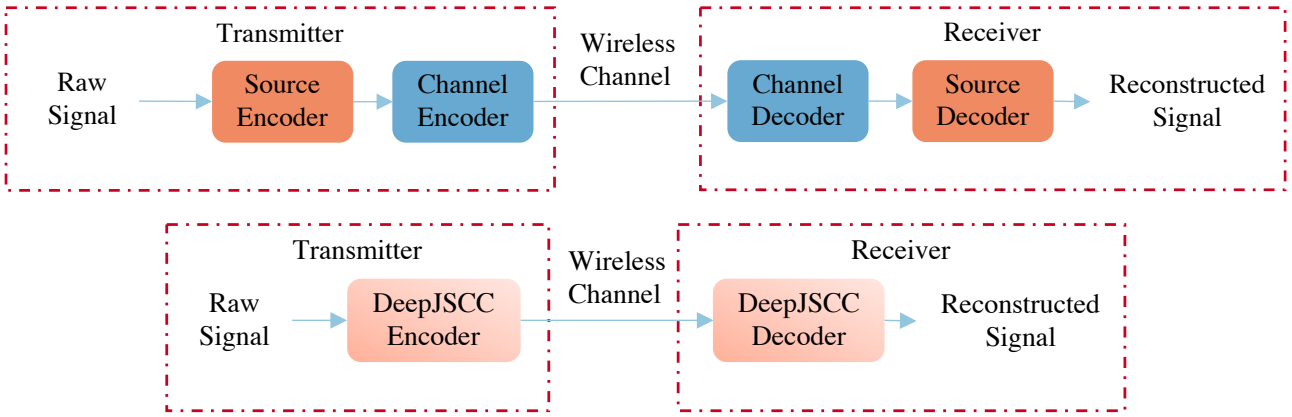


Figure 2.1: Separation and DeepJSCC-based data transmission schemes

continuous-amplitude nature of DeepJSCC transmissions enables superior adaptation to varying channel conditions compared to discrete digital schemes, making it particularly suitable for dynamic wireless environments where channel conditions fluctuate rapidly.

On the other hand, JSCC is not limited to continuous-amplitude channel inputs. Even when transmission is constrained to discrete constellations or the use of conventional beamforming and channel coding techniques, compression and channel coding/modulation can be optimized in an end-to-end fashion with the introduction of modules that are parameterized as neural networks as shown in [32] and [33].

Figure 2.1 illustrates the structural difference between a conventional separation-based architecture and the DeepJSCC approach. While the separation approach utilizes distinct, independently optimized modules for source coding and channel coding, the DeepJSCC architecture replaces these sequential blocks with a single joint encoder neural network at the transmitter and a joint decoder neural network at the receiver, optimizing the entire pipeline end-to-end for the specific channel conditions.

2.1.1 Point-to-Point System Model

We consider wireless image transmission over a noisy channel. The transmitter maps an input image $\mathbf{x} \in \mathbb{I}^{C_{\text{in}} \times W \times H}$, where W and H denote the width and height, and C_{in} represents the number of color channels (e.g., $C_{\text{in}} = 3$ for RGB images), into a complex-valued latent vector

using a nonlinear encoding function $E_{\mathbf{\Theta}} : \mathbb{I}^{C_{\text{in}} \times W \times H} \to \mathbb{C}^k$, parameterized by $\mathbf{\Theta}$:

$$\mathbf{z} = E_{\mathbf{\Theta}}(\mathbf{x}, \sigma), \tag{2.1}$$

where k denotes the available channel bandwidth (number of complex channel uses) and σ is the channel noise standard deviation. In fading channels, the encoder additionally takes the channel gain h as input. The transmitted signal $\mathbf{z}$ is subject to the average power constraint:

$$\frac{1}{k} \|\mathbf{z}\|_2^2 \leq P_{\text{avg}}, \tag{2.2}$$

where k is the number of channel uses and P_{avg} is the average power budget.

For additive white Gaussian noise (AWGN) channels, the received signal is:

$$\mathbf{y} = \mathbf{z} + \mathbf{n}, \tag{2.3}$$

where $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_k)$ is independent and identically distributed (i.i.d.) complex Gaussian noise. We assume the noise variance σ^2 is known at both the transmitter and the receiver.

A nonlinear decoding function $D_{\mathbf{\Phi}} : \mathbb{C}^k \to \mathbb{I}^{C_{\text{in}} \times W \times H}$, parameterized by $\mathbf{\Phi}$, reconstructs the image from the channel output:

$$\hat{\mathbf{x}} = D_{\mathbf{\Phi}}(\mathbf{y}, \sigma). \tag{2.4}$$

The *bandwidth ratio* ρ characterizes the available channel resources:

$$\rho \triangleq \frac{k}{C_{\text{in}} W H} \text{ channel symbols/pixel.} \tag{2.5}$$

The channel SNR is defined as:

$$\text{SNR} \triangleq 10 \log_{10} \left(\frac{P_{\text{avg}}}{\sigma^2} \right) \text{ dB.} \tag{2.6}$$

Given ρ and SNR, the goal is to minimize the average distortion between the original image $\mathbf{x}$

and the reconstruction $\hat{\mathbf{x}}$:

$$\arg\min_{\boldsymbol{\Theta},\boldsymbol{\Phi}} \mathbb{E}\left[d(\mathbf{x},\hat{\mathbf{x}})\right], \tag{2.7}$$

where $d(\cdot,\cdot)$ is a differentiable distortion measure and the expectation is over the source distribution and channel noise.

2.1.2 DeepJSCC Neural Network Architecture

In Chapters 3 to 6, we build upon a common convolutional neural network (CNN)-based autoencoder architecture for DeepJSCC. We describe this shared architecture here and detail chapter-specific modifications in the respective chapters.

We employ the encoder and decoder architectures from [32], removing the quantization stage as DeepJSCC operates with continuous-amplitude channel inputs. The resulting architecture features nearly symmetric networks with progressive spatial downsampling at the encoder and corresponding upsampling at the decoder. CNNs enable parameter-efficient extraction of high-level features by exploiting spatial structures within images, and have been shown to perform well for various vision-related tasks, including DeepJSCC for image transmission [2]. The architecture is composed of the following building blocks:

Residual blocks. The backbone is constructed from residual blocks [34], each consisting of two convolutional layers with a skip connection that facilitates gradient flow during training. Downsampling residual blocks use a stride of s=2 to halve the spatial dimensions, while upsampling blocks employ sub-pixel convolution (pixel shuffle) to double them. The internal channel count is set to N=256 by default, except for the final encoder layer, whose output channel count C_{out} controls the bandwidth ratio, and the final decoder layer, which outputs C_{in} channels for image reconstruction.

Attention mechanism. The architecture incorporates the computationally efficient non-local attention module from [35], placed at strategic points in both the encoder and decoder. This module captures long-range spatial dependencies, allowing the network to allocate representational capacity to semantically important regions of the input.

Attention feature (AF) module. To enable a single trained model to operate across a range of channel conditions, the architecture leverages SNR adaptivity through the AF module proposed in [36]. The AF module takes the channel SNR as an additional input and produces channel-wise scaling factors via a small multi-layer perceptron, allowing the network to learn SNR-dependent feature modulations. By training with randomly sampled SNR values and channel gains for fading channels, the model adapts to different operating points without retraining. AF modules are inserted throughout both the encoder and decoder.

Power normalization. A power normalization layer at the encoder output enforces the average transmit power constraint in Eq. (2.2). The raw encoder output $\tilde{\mathbf{z}}$ is normalized as $\mathbf{z} = \sqrt{kP_{\text{avg}}/\left\|\tilde{\mathbf{z}}\right\|_2^2}\,\tilde{\mathbf{z}}$, ensuring the constraint is satisfied with equality.

The number of output channels C_{out}, combined with the spatial downsampling factor, determines the bandwidth ratio ρ (defined in Section 2.1.1). Lower bandwidth ratios correspond to more aggressive compression, while higher ratios allow the encoder to preserve finer details.

Individual chapters adapt this backbone by varying the number of downsampling stages to produce lower-resolution reconstructions suitable for generative restoration (Chapter 3), augmenting the encoder input dimensions to incorporate device-specific embeddings (Chapter 5), or introducing dedicated modules such as multi-scale side information fusion (Chapter 4) and learnable user-specific projection layers (Chapter 6).

2.2 Generative Models for Semantic Communication

While the DeepJSCC framework described above optimizes for distortion metrics such as mean squared error (MSE), many applications demand high perceptual quality at the receiver. Recent advances in generative models have opened new possibilities for perceptual quality-oriented communication systems. These models can generate semantically consistent reconstructions even under severe channel impairments, prioritizing perceptual quality over traditional distortion metrics. Rather than pursuing exact reconstruction, this line of work aims to preserve semantic meaning at the receiver [4, 37].

Several key generative frameworks have been developed and applied to communication systems. Variational Autoencoders (VAEs), introduced in 2013, provide a probabilistic generative framework that can be adapted for communication systems [38]. VAEs learn compressed latent representations that can be efficiently transmitted, with the decoder generating reconstructions from the received latent codes. Their probabilistic nature makes them particularly suitable for handling channel uncertainty and noise [39].

Generative Adversarial Networks (GANs), proposed in 2014, have found applications in communication systems, particularly for image transmission where perceptual quality is crucial [40, 41]. GAN-based approaches can learn to generate realistic reconstructions that maintain semantic content even when traditional metrics like PSNR indicate poor reconstruction quality.

Diffusion models operate by gradually adding noise to data during the forward process and learning to reverse this process to generate new samples [42]. In communication systems, diffusion models can be integrated into the decoding process to generate high-quality reconstructions from corrupted received signals. The key advantage is their ability to capture complex data distributions and generate diverse, semantically consistent outputs [5].

2.2.1 Inverse Problem Formulation

The connection between communication systems and inverse problems has spurred the application of these powerful generative models to joint source-channel coding (JSCC). Pre-trained generative models have been utilized to solve classical inverse problems in compressed sensing [43], while similar approaches have delivered remarkable results in single-image super-resolution [44], demonstrating their capability in tackling practical inverse challenges. This parallel has inspired innovative JSCC methods, including privacy-preserving techniques [45], adversarial approaches [46], and generative frameworks [47].

A notable contribution is InverseJSCC [47], which frames the restoration of high-fidelity source images from degraded channel outputs as an inverse problem. By employing a pre-trained StyleGAN-2 generator alongside intermediate latent optimization (ILO) [48], this method attains

impressive perceptual quality for image recovery in harsh channel environments. Building on this foundation, the work in [49] extends the concept to adaptive generative scenarios, where the receiver continuously refines its model using previously received images, enabling ongoing adaptation to evolving image distributions.

2.2.2 Diffusion Models for Communication

Diffusion models have revolutionized visual generation and are increasingly applied to inverse problems, especially image restoration. Zero-shot techniques are particularly promising, as they enable the direct use of unconditionally trained diffusion models for restoration tasks without task-specific fine-tuning. For instance, the method in [50] steers diffusion model outputs toward restored images by substituting low-frequency components in the denoised result with those from the degraded reference. Furthermore, the approach in [51] advances diffusion model solvers via approximate posterior sampling, addressing broad noisy image restoration challenges.

The robust generative power of diffusion models has led to their integration in semantic communications. In [37], a hybrid JSCC approach merges conventional transmission with diffusion-based noise addition to input images, followed by DeepJSCC transmission, yielding superior decoder reconstruction. The denoising capabilities of diffusion models are harnessed in [52] to counteract channel noise at the receiver in wireless systems. Under stringent bandwidth limitations, the work in [53] models channel transmission as the diffusion forward process, employing VAE-based compression to minimize bandwidth while capitalizing on VAEs' advanced encoding efficiency. Conditional diffusion guided by semantic cues, such as textual descriptions or edge maps, is explored in [54], with a semantics-driven channel denoising strategy that dynamically adjusts to varying channel states. Beyond image transmission, diffusion models have been applied to extreme video compression, where pre-trained conditional diffusion models at the decoder generate subsequent frames from compressed references, achieving superior perceptual quality at very low bit rates compared to traditional codecs [55].

CommIN [56] presents a JSCC framework that conceptualizes image reconstruction as an inverse problem, combining invertible neural networks (INNs) and diffusion models to boost perceptual

fidelity, surpassing DeepJSCC in extreme scenarios. The INN-based degradation modeling in CommIN separates coarse and fine image components, empowering diffusion models to enhance reconstructions effectively.

In semantic communication systems, generative models enable the use of perceptual quality metrics such as learned perceptual image patch similarity (LPIPS) [57], Fréchet Inception Distance (FID) [58], and Kernel Inception Distance (KID) [59], which better align with human perception than traditional pixel-wise metrics.

Recent work has shown that generative approaches achieve superior performance in challenging scenarios, including very low SNR and bandwidth-constrained settings [60, 37]. These methods mark a shift from reconstruction-oriented to generation-oriented communication, where the receiver generates content that preserves semantic meaning rather than attempting pixel-perfect recovery [61].

Implementation considerations for generative models in communication systems include training stability, computational complexity, and mode collapse in GANs. Recent advances in conditional generation and latent diffusion models [62] have addressed many of these challenges, enabling more efficient and stable training for communication applications [63].

2.3 Distributed Source Coding

When multiple users transmit correlated sources, compression can benefit from exploiting inter-source correlation even when encoders cannot communicate with each other. Distributed source coding addresses precisely this scenario, where multiple correlated sources are encoded separately but decoded jointly. The theoretical foundations were established through seminal results by Slepian and Wolf [6], who proved that an encoder that does not observe a correlated side information can asymptotically achieve the same compression rate as the one that does (in both cases, the decoder has access to the side information), if the joint distribution statistics are known and compression is lossless. The Wyner-Ziv theorem [7] extended this result to lossy compression, characterizing the rate-distortion function with side information available at both

the encoder and decoder, or only at the decoder. Surprisingly, they showed that there is no rate loss in the latter scenario compared to the former, if the sources are Gaussian and mean-squared error is set as the distortion criterion. Practical research effort for the Wyner-Ziv setup has been spearheaded by distributed source coding using syndromes (DISCUS) [64], which formulated the compression problem as a dual source-channel coding problem.

Recent neural approaches to distributed source coding [65, 66, 67] have demonstrated that neural networks can learn optimal binning structures reminiscent of classical information-theoretic constructions. These methods exploit correlation between sources to achieve compression gains beyond point-to-point schemes. These works, inspired by the Slepian-Wolf and Wyner-Ziv results, are concerned with exploiting the side information in order to further compress the primary input source, compared to the point-to-point (having no side information) compression scenario. In particular, recent work has demonstrated that neural Wyner-Ziv compressors can learn "random binning" structures, akin to the achievability part of the Wyner-Ziv theorem.

2.4 Non-Orthogonal Multiple Access (NOMA)

Beyond exploiting source correlation, multi-user systems must also address the challenge of sharing the wireless channel among multiple transmitters.

2.4.1 Orthogonal vs. Non-Orthogonal Access

In conventional wireless systems, orthogonal multiple access (OMA) schemes, such as time division multiple access (TDMA), frequency division multiple access (FDMA), and orthogonal frequency division multiple access (OFDMA), divide the available radio resources (time, frequency, or code) among users so that each transmission occupies an orthogonal, non-overlapping portion of the resource space. By design, OMA eliminates inter-user interference entirely: each user transmits over a dedicated resource block. While this simplifies receiver design, it limits each user to a fraction of the total resources.

NOMA, by contrast, allows multiple users to share the *same* time-frequency resources simultaneously, deliberately introducing controlled inter-user interference that is resolved at the receiver through advanced signal processing. To formalize, consider a K-user uplink multiple access channel (MAC) where user k transmits signal $\mathbf{z}_k$ with average transmit power P_k^{tx}. The received signal at the base station is

$$\mathbf{y} = \sum_{k=1}^{K} h_k \, \mathbf{z}_k + \mathbf{n},$$

where $h_k \in \mathbb{R}$ denotes the real-valued channel coefficient for user k and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, N_0\mathbf{I})$ is additive white Gaussian noise. Throughout this section, $P_k \triangleq h_k^2 P_k^{\text{tx}}$ denotes the received power of user k, absorbing the channel gain. In OMA, only one user transmits at any given time-frequency slot, so the receiver sees $\mathbf{y} = h_k\mathbf{z}_k + \mathbf{n}$ without any inter-user interference. In NOMA, all users transmit concurrently, and the receiver must disentangle the superposed signals.

2.4.2 Capacity Region of the Multiple Access Channel

The capacity region of the K-user real-valued Gaussian MAC was established independently by Ahlswede [68] and Liao [69] (see also [70, 71]). It is the set of non-negative rate tuples $(R_1, \ldots, R_K)$, in bits per channel use, achievable with vanishingly small error probability, satisfying

$$\sum_{k\in\mathcal{S}} R_k \le \tfrac{1}{2}\log_2\left(1 + \frac{\sum_{k\in\mathcal{S}} P_k}{N_0}\right), \quad \forall\, \mathcal{S} \subseteq \{1, \ldots, K\},\ \mathcal{S} \neq \emptyset. \tag{2.8}$$

This yields $2^K - 1$ constraints (one per non-empty subset) and defines a polymatroidal region. Each vertex of the dominant face corresponds to a successive interference cancellation (SIC) decoding order (a permutation of the K users), where user $\pi(k)$ is decoded at step k while treating the signals of users $\{\pi(k+1), \ldots, \pi(K)\}$ as noise. There are $K!$ such vertices, and the sum capacity is

$$C_\Sigma = \tfrac{1}{2}\log_2\left(1 + \frac{\sum_{k=1}^{K} P_k}{N_0}\right).$$

Since C_Σ depends on the total received power, each additional user strictly increases the sum capacity; however, the growth is logarithmic. When all users have equal received power P, the sum capacity $C_\Sigma = \frac{1}{2}\log_2(1 + KP/N_0)$ scales as $\mathcal{O}(\log_2(K))$, exhibiting diminishing per-user

marginal gains.

For $K=2$, the three constraints ($2^2-1=3$) specialize to

$$R_1 \le \tfrac{1}{2}\log_2(1+P_1/N_0) \triangleq C_1,$$
$$R_2 \le \tfrac{1}{2}\log_2(1+P_2/N_0) \triangleq C_2,$$
$$R_1+R_2 \le \tfrac{1}{2}\log_2(1+(P_1+P_2)/N_0) \triangleq C_\Sigma.$$

As illustrated in Fig. 2.2, this region forms a *pentagon* in the (R_1, R_2) plane. The dominant face, i.e., the segment between corner points A and B, is governed by the sum-rate constraint $R_1+R_2=C_\Sigma$. Corner point $A=\left(\tfrac{1}{2}\log_2(1+P_1/(P_2+N_0)),\ C_2\right)$ is achieved by SIC that decodes user 1 first while treating user 2's signal as noise, then subtracts user 1's contribution and decodes user 2 interference-free. Corner point $B=\left(C_1,\ \tfrac{1}{2}\log_2(1+P_2/(P_1+N_0))\right)$ corresponds to the reverse decoding order: user 2 is decoded first (treating user 1 as noise), then user 1 is decoded free of interference. Any point on the segment AB can be reached via time-sharing between the two SIC orders or via joint decoding [70, 71].

Under TDMA with a peak-power constraint, user k is allocated a fraction α_k of the total time (with $\alpha_1+\alpha_2 \le 1$), achieving rate $R_k=\alpha_k C_k$. The resulting TDMA rate region is the triangle with vertices $(0,0)$, $(C_1,0)$, and $(0,C_2)$, which is strictly contained within the MAC capacity pentagon (Fig. 2.2). The green-shaded area in the figure represents rate pairs achievable *only* through non-orthogonal transmission. In particular, the maximum TDMA sum rate is $\max(C_1,C_2)$, while NOMA achieves C_Σ; the gap $C_\Sigma-\max(C_1,C_2)>0$ is strictly positive whenever both users are active.

2.4.3 Trade-offs Between Orthogonal and Non-Orthogonal Access

NOMA is advantageous in several settings:

- **Spectral efficiency:** NOMA achieves the full MAC capacity region, whereas OMA is limited to a strict subset. Since $\log_2(1+x)$ is strictly concave, $C_\Sigma < C_1+C_2$ holds, meaning

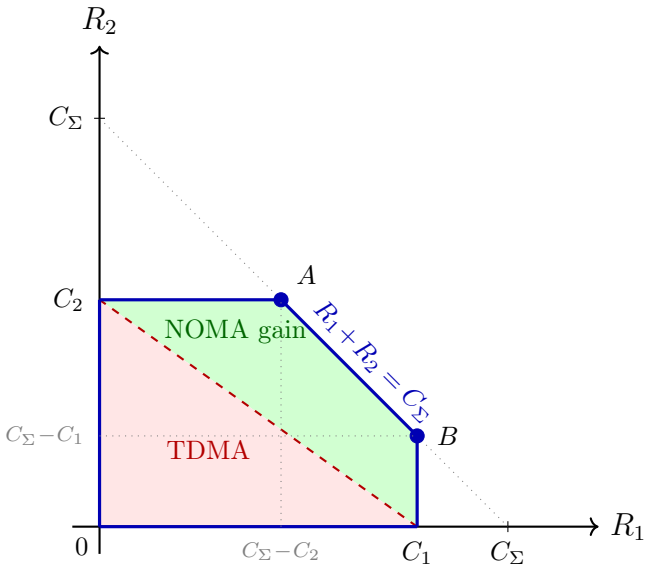


Figure 2.2: Capacity region of a two-user real-valued Gaussian MAC. The pentagon (blue boundary) is the full capacity region, achievable through NOMA with joint decoding or SIC. Corner point A corresponds to SIC that decodes user 1 first (user 2 interference-free); corner point B to the reverse order. The dashed red line shows the TDMA boundary. The green shaded area highlights rate pairs achievable only through non-orthogonal transmission. The intercepts C_Σ on both axes mark the maximum achievable sum rate, which strictly exceeds the TDMA sum rate $\max(C_1, C_2)$.

the sum-rate constraint is tighter than the individual bounds combined.

- **Overloaded systems:** When the number of users exceeds the available orthogonal resource blocks, OMA cannot serve all users simultaneously; NOMA naturally accommodates any number of users through power-domain multiplexing.

- **Heterogeneous channel conditions:** When users have disparate received powers (large P_1/P_2 ratio), SIC can effectively separate the strong user's signal before decoding the weak user, yielding substantial rate gains over equal resource partitioning.

Conversely, OMA remains attractive when:

- **Low complexity is a priority:** OMA avoids the need for SIC or joint decoding, simplifying the receiver architecture.

- **Channel estimation is imperfect:** SIC performance degrades with channel estimation errors, since residual interference from imperfect cancellation accumulates across decoding stages. OMA sidesteps this issue entirely.

Table 2.1: Comparison of Deep Learning-Based Transmission Methods for NOMA

Paper	Demonstrated Max # Users	Dataset(s)	Channel(s)	Coding	Modality	Uplink/Downlink
Ours (Chapter 6) [23]	64	CIFAR,TinyImagenet, Cityscapes,Kodak	AWGN, Rayleigh	JSCC	Image	Uplink
Ours (Chapter 5) [19]	2	CIFAR	AWGN	JSCC	Image	Uplink
DBC Aware JSCC [12]	2	CIFAR	AWGN	JSCC	Image	Downlink
NOMASC [11]	3	MNIST,CIFAR, Europarl	AWGN, Rayleigh	JSCC	Image, Text	Downlink
DeepSCM [72]	2	CIFAR	AWGN	JSCC	Image	Downlink
IS-SNOMA [73]	2	Cityscapes	Rician	JSCC	Image	Downlink
VAE-MAC [39]	2	Gaussian	AWGN	JSCC	Bits	Downlink
MDC-NOMA [74]	2	Baboon,Lena	AWGN, Rayleigh	Separation	Image	Downlink
MDC-NOMA-STBC [75]	2	Baboon,Lena	AWGN, Rayleigh	Separation	Image	Downlink
DeepSIC [76]	2	Gaussian	AWGN, Poisson	Separation	Bits	Uplink
CNN-SIC [77]	2	Gaussian	AWGN, Rayleigh	Separation	Bits	Downlink
SICNet [78]	2	Gaussian	AWGN	Separation	Bits	Downlink

- **Strict per-user fairness is required:** Orthogonal resource allocation provides deterministic, per-user rate guarantees, whereas NOMA rate regions depend on the decoding order and power allocation.

2.4.4 Deep Learning-Based NOMA Transmission

As discussed in Section 2.4.2, multiple strategies can achieve the MAC capacity region, including joint decoding and SIC combined with rate-splitting [79, 80, 81, 82, 83, 84]. Recently, deep neural networks (DNNs) have been employed to implement SIC for digital NOMA systems [76, 77, 78], showing promising results. However, the integration of NOMA with DeepJSCC presents a unique challenge: since input signals are directly mapped to continuous-amplitude channel inputs without constellation constraints, the decoder functions as an estimator and always has some noise in its reconstruction. Unlike in digital communication, the decoder cannot perfectly recover the transmitted codeword, making perfect interference cancellation impossible. For this reason, joint decoding approaches become more attractive for continuous-amplitude transmissions, as they can handle the inherent estimation uncertainty.

Most existing deep learning-based NOMA schemes are limited to two-user scenarios as shown in Table 2.1, creating a significant gap between theoretical potential and practical applicability. The challenge lies in managing increasing interference levels as the number of users grows, while maintaining computational tractability during training. Recent work has begun addressing

these scalability limitations through progressive training approaches and orthogonal user-specific projections.

2.5 Distributed Inference and Privacy

The preceding sections focused on point-to-point or small-scale multi-user transmission; we now turn to systems where many edge devices collaborate for a common inference task. The proliferation of Internet of Things (IoT) devices and edge computing applications has fundamentally transformed how inference tasks are performed in modern distributed systems. Traditional cloud-centric approaches face limitations in dynamic environments due to latency constraints, privacy requirements, bandwidth limitations, and energy considerations [85]. Edge inference addresses these challenges by distributing computations across multiple local devices, enabling real-time decision-making while preserving data privacy.

The fundamental challenge in distributed inference lies in efficiently aggregating information from distributed sources while handling the inherent heterogeneity in device capabilities, data distributions, channel conditions, and participation patterns. Static environments, such as surveillance camera networks or sensor deployments, benefit from edge inference due to latency and privacy constraints [86, 87]. Dynamic environments present even greater challenges, as devices like autonomous vehicles, drones, and mobile phones must adapt to changing conditions while maintaining inference accuracy [88, 85].

2.5.1 Over-the-Air Computation (OAC)

OAC leverages the superposition property of wireless channels to perform computation directly over the air [89]. In conventional communication systems, the superposition of signals is viewed as interference to be mitigated. OAC, however, exploits this natural property to perform distributed computation efficiently. In OAC schemes, multiple devices transmit their local computations simultaneously, and the receiver obtains the aggregated result corrupted by

channel noise. This approach provides significant bandwidth efficiency compared to orthogonal transmission schemes and has natural privacy-preserving properties due to the aggregation process.

The mathematical foundation of OAC relies on the linearity of certain computations and the additive nature of wireless channels. For a MAC, when K devices transmit signals x_k simultaneously, the received signal is $y = \sum_{k=1}^{K} h_k x_k + n$, where h_k represents the channel coefficient for device k and n is the additive noise. When the desired computation is a linear combination of the transmitted values, OAC can directly compute the result without requiring individual signal recovery.

OAC has found significant applications in federated learning (FL) [90, 91, 92, 93, 94], where it provides both communication efficiency and differential privacy (DP) amplification through the natural aggregation and channel noise. The privacy amplification effect arises because individual contributions are masked by both the aggregation process and the wireless channel noise, making it difficult for adversaries to infer individual model updates. This natural privacy protection is particularly valuable in scenarios where explicit privacy-preserving mechanisms would be computationally expensive or reduce utility.

For edge inference applications, OAC enables efficient ensemble learning where multiple local models can contribute to a global decision [18]. The key advantage is that devices can participate without revealing their individual model parameters or local data, addressing privacy concerns while achieving superior inference accuracy through model diversity. However, OAC faces challenges including channel estimation errors, synchronization requirements, and the need for careful power control to ensure fair aggregation.

2.5.2 Federated Learning and Distributed Inference

While FL [95] focuses on collaboratively training a single global model across distributed devices, distributed inference aims to combine predictions from independently trained local models at inference time. FL requires multiple rounds of communication and coordination, making it

suitable for scenarios where long-term model improvement is prioritized. Distributed inference, in contrast, provides immediate results with minimal coordination, making it more suitable for latency-critical applications [85].

The integration of FL with wireless communication techniques, particularly OAC, has emerged as a promising research direction [90, 91, 96]. OAC can provide natural privacy amplification in FL through the aggregation of model updates directly in the wireless channel, combined with channel noise effects [89, 97]. FL has found applications across diverse domains including finance [98], medicine [99, 100], and wireless communications [15, 101].

2.5.3 Privacy in Distributed Wireless Systems

Privacy preservation in distributed systems has become increasingly critical with the widespread adoption of edge computing and the growing awareness of data protection requirements [85]. The challenge lies in enabling collaborative computation while protecting individual users' sensitive information from both external adversaries and honest-but-curious system participants [86, 87]. DP [102] provides formal privacy guarantees through controlled noise addition, establishing a mathematical framework for quantifying the privacy-utility trade-off.

The concept of differential privacy is particularly relevant in wireless distributed systems, where multiple mechanisms can contribute to privacy protection [89]. Classical differential privacy requires explicit noise addition to query responses, but wireless systems offer natural sources of privacy protection that can be leveraged and amplified [93, 94]. Understanding these inherent privacy-preserving properties enables the design of systems that achieve strong privacy guarantees without significant utility degradation [18].

In wireless distributed inference, privacy amplification can occur naturally through several mechanisms: (1) aggregation of multiple users' contributions, which masks individual inputs through the summation process [89], (2) channel noise, which adds natural randomness to transmitted signals [97], and (3) quantization effects in digital systems, which introduce additional noise to the communication process [90]. The combination of these effects can provide substantial

privacy amplification, meaning that the effective privacy protection is stronger than what would be achieved by each mechanism individually [18].

The intersection of communication efficiency, inference accuracy, and privacy protection creates complex trade-offs that require careful system design [88, 85]. Over-the-air computation naturally provides privacy protection through aggregation, as individual contributions are inherently mixed in the wireless channel [89]. JSCC can potentially provide additional privacy benefits by learning representations optimized for transmission rather than interpretability, making it difficult for adversaries to extract sensitive information from intercepted signals. Recent advances in privacy-aware communication systems leverage DeepJSCC and generative models to selectively protect sensitive information while preserving the utility of non-sensitive features [45, 60]. In these approaches, sensitive features are deliberately hidden under noise or transformed into representations that are difficult to reconstruct, while public or non-sensitive features are transmitted with minimal protection to maintain communication efficiency. This selective mechanism enables fine-grained control over the privacy-utility trade-off, achieving strong privacy guarantees for critical information while maintaining high performance for task-relevant data.

2.6 Deep Learning Techniques for Multi-User Systems

Several deep learning techniques are used across the multi-user semantic communication systems developed in this thesis. This section reviews these cross-cutting methods.

2.6.1 Ensemble and Multi-View Classification

Ensemble learning methods combine multiple hypotheses instead of constructing a single best hypothesis to model the data [103]. In ensemble learning, each hypothesis votes for the final decision, where votes can have weights depending on their confidence. It is generally intractable to find the optimal hypothesis, and choosing a model among a set of equally good models has

the risk of choosing the model that has worse generalization performance; however, averaging these models would reduce this risk [103, 104]. Furthermore, weighted or voting-based ensemble methods have theoretical guarantees; for example, it can be shown that the expected error of an averaging ensemble of models is not greater than the average of the expected errors of individual models under the MSE loss [104].

The effectiveness of ensemble methods depends critically on model diversity [105]. Two important ensemble paradigms relevant to distributed settings are *bagging* [106] and *dagging* [107]. Bagging trains models on bootstrap samples drawn with replacement from the original dataset, while dagging trains individual models on disjoint datasets. The dagging approach is particularly relevant for distributed edge scenarios where devices naturally possess non-overlapping local datasets, improving both diversity and privacy of local models.

In wireless communication systems, ensemble methods find particular application in distributed inference scenarios where multiple edge devices collaborate to make decisions [18, 85]. The ensemble approach provides robustness against individual device failures and channel impairments, as the collective decision-making process can compensate for poor performance from any single device. Mathematically, if we denote the prediction from the k-th device as $p_k(\mathbf{x})$, the ensemble prediction can be formulated as:

$$\hat{y} = \arg\max_{y} \sum_{k=1}^{K} w_k p_k(y|\mathbf{x})$$

where w_k represents the confidence weight for device k, and K is the number of participating devices.

Multi-view classification deals with multiple feature perspectives, while ensemble classification performs classification on the same whole instance. Both approaches aim to improve classification performance through information fusion. Multi-view classification utilizes correlations among different views [108], employing data-level, feature-level, or decision-level fusion techniques [109, 110]. Decision-level fusion is particularly attractive for distributed settings as it requires no data or feature sharing between models, improving both latency and privacy.

Common decision-level fusion methods include averaging, weighted averaging, and voting [103]. Empirical studies show that simple averaging often performs comparably to weighted averaging [111, 112], particularly when individual models have similar performance levels. Weighted averaging becomes advantageous when models exhibit significantly different performance characteristics [113], though determining reliable weights can be challenging with noisy or insufficient data.

2.6.2 Multi-Task Learning

Multi-task learning (MTL) [114, 115] enables networks to learn multiple related tasks simultaneously, often improving performance on individual tasks through shared representations. In multi-user communication systems, MTL principles can be applied to jointly learn encoding and decoding functions for multiple users, potentially reducing complexity while maintaining performance.

MTL architectures typically employ parameter sharing strategies, including hard sharing where layers are completely shared between tasks, and soft sharing where task-specific parameters are learned with regularization to encourage similarity [116]. In communication systems, hard parameter sharing is particularly effective for learning joint representations that can be efficiently transmitted over bandwidth-constrained channels [19].

For multi-user JSCC systems, MTL can be applied to learn a unified encoder that maps different users' data to a shared latent space, while maintaining task-specific decoders for individual reconstruction [19]. This approach reduces the total number of parameters compared to separate single-user systems, enabling more efficient deployment on resource-constrained edge devices.

The effectiveness of MTL in communication systems stems from the shared statistical properties across different users and tasks. For example, in image transmission scenarios, different users may transmit images from similar domains (e.g., surveillance footage, medical images), allowing the system to learn shared features that are robust to channel impairments. The multi-task objective can be formulated as:

$$\mathcal{L}_{MTL} = \sum_{t=1}^{T} \alpha_t \mathcal{L}_t + \lambda \Omega(\theta),$$

where $\mathcal{L}_t$ is the loss for task t, α_t are task weights, $\Omega(\theta)$ is a regularization term encouraging parameter sharing, and λ controls the regularization strength.

Advanced MTL techniques include gradient-based task weighting [117] and uncertainty-based weighting [118], which dynamically adjust task importance during training. In communication contexts, these methods adapt to varying channel conditions and user requirements, making MTL particularly suited for dynamic wireless environments [119].

2.6.3 Deep Metric Learning

Deep metric learning (DML) techniques learn representations that preserve important relationships between data points, which is valuable for systems that need to handle multiple users or sources. DML approaches typically use contrastive or triplet loss functions to learn embeddings where similar samples are close in the latent space while dissimilar samples are pushed apart [120]. In communication systems, DML can be applied to learn robust representations that are resilient to channel distortions. For instance, triplet loss can be formulated as:

$$\mathcal{L}_{triplet} = \max(0, d(\mathbf{z}_a, \mathbf{z}_p) - d(\mathbf{z}_a, \mathbf{z}_n) + m)$$

where $\mathbf{z}_a$, $\mathbf{z}_p$, and $\mathbf{z}_n$ are the anchor, positive, and negative embeddings, respectively, and m is a margin parameter that controls the separation between positive and negative pairs. The margin ensures that the distance between anchor-positive pairs is at least m units smaller than the distance between anchor-negative pairs, preventing trivial solutions where all embeddings collapse to the same point.

Advanced DML methods include N-pair loss [121] and lifted structured loss [122], which provide more efficient sampling strategies for large-scale datasets. In multi-user communication scenarios, these methods can learn user-specific embeddings that preserve semantic relationships while being robust to channel variations [10].

2.6.4 Curriculum Learning

Curriculum learning (CL) [123] trains complex systems by starting with simpler tasks and gradually increasing difficulty, which has proven effective for scaling neural communication systems to larger numbers of users. In communication systems, curriculum strategies can be applied to progressively increase the number of users, channel complexity, or data diversity during training. For example, a curriculum for multi-user JSCC might start with single-user transmission under ideal channel conditions, then gradually introduce multiple users and more challenging channel models [27, 124].

Recent work has shown that curriculum learning can significantly improve convergence and final performance in large-scale communication systems [125]. The curriculum can be designed based on various criteria including SNR, number of users, or data complexity, with automatic curriculum generation methods showing particular promise for adaptive system design [126].

2.7 Image Quality Evaluation Metrics

Throughout this thesis, we evaluate image reconstruction quality using several complementary metrics that capture different aspects of image fidelity. We define these metrics here for reference across subsequent chapters.

2.7.1 Peak Signal-to-Noise Ratio (PSNR)

The peak signal-to-noise ratio (PSNR) measures pixel-level reconstruction fidelity and is defined as:

$$\mathrm{PSNR}(\mathbf{x}, \hat{\mathbf{x}}) = 10 \log_{10} \left(\frac{A^2}{\frac{1}{C_{\mathrm{in}} HW} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2} \right) \ \mathrm{dB}, \tag{2.9}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the original and reconstructed images, respectively, C_{in}, H, and W denote the number of color channels, height, and width of the image, and A is the maximum possible pixel value (e.g., $A = 255$ for 8-bit images). Higher PSNR indicates better reconstruction quality.

2.7.2 Structural Similarity Index (SSIM)

The structural similarity index measure (SSIM) [127] captures structural information beyond pixel-level differences by comparing luminance, contrast, and structure between two images. It is defined as:

$$\mathrm{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{(2\mu_{\mathbf{x}}\mu_{\hat{\mathbf{x}}} + c_1)(2\sigma_{\mathbf{x}\hat{\mathbf{x}}} + c_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\hat{\mathbf{x}}}^2 + c_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\hat{\mathbf{x}}}^2 + c_2)}, \tag{2.10}$$

where $\mu_{\mathbf{x}}$ and $\mu_{\hat{\mathbf{x}}}$ are the means, $\sigma_{\mathbf{x}}^2$ and $\sigma_{\hat{\mathbf{x}}}^2$ are the variances, $\sigma_{\mathbf{x}\hat{\mathbf{x}}}$ is the covariance between the two images, and c_1, c_2 are stabilization constants. SSIM values range from -1 to 1, with 1 indicating perfect structural similarity.

2.7.3 Multi-Scale Structural Similarity (MS-SSIM)

Multi-scale SSIM (MS-SSIM) [128] extends SSIM by evaluating structural similarity across multiple spatial scales. Images are iteratively downsampled, and SSIM is computed at each scale j. The final score is a weighted geometric mean:

$$\text{MS-SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = \left[\prod_{j=1}^{M} \mathrm{SSIM}(\mathbf{x}^j, \hat{\mathbf{x}}^j)^{w_j}\right]^{\frac{1}{\sum_{j=1}^{M} w_j}}, \tag{2.11}$$

where M is the number of scales, $\mathbf{x}^j$ and $\hat{\mathbf{x}}^j$ denote the images at scale j, and w_j are the per-scale weights. We use default weights from [128], adjusting the filter size to the maximum value that does not exceed the image resolution or the default value of 11. MS-SSIM has been shown to correlate better with human perception than PSNR [128].

2.7.4 Learned Perceptual Image Patch Similarity (LPIPS)

LPIPS [57] measures perceptual similarity by computing distances in the feature space of a pre-trained deep neural network (e.g., VGG or AlexNet). Unlike pixel-wise metrics, LPIPS captures high-level perceptual differences that align more closely with human judgement. Lower LPIPS scores indicate greater perceptual similarity. Throughout this thesis, we employ

a pre-trained VGG network for LPIPS evaluation. These metrics provide complementary perspectives: PSNR quantifies pixel-level fidelity, SSIM and MS-SSIM capture structural similarity at single and multiple scales, and LPIPS measures perceptual quality as judged by learned features.

The background presented in this chapter forms the foundation for our novel contributions in multi-user semantic communication, as detailed in subsequent chapters: generative approaches for perceptual quality optimization (Chapter 3), distributed source coding with decoder-only side information (Chapter 4), scalable NOMA-based systems (Chapters 5 and 6), and the privacy-preserving ensemble inference framework (Chapter 7).

Chapter 3

High Perceptual Quality Wireless Image Delivery with Denoising Diffusion Models

3.1 Introduction

As outlined in Chapter 1, this thesis addresses the first research objective by developing JSCC techniques with generative models for enhanced perceptual quality. Building upon the DeepJSCC background established in Chapter 2, this chapter considers a single-user point-to-point setting and introduces a novel diffusion-based DeepJSCC framework that leverages controlled degradation and restoration for improved image transmission quality.

Standard DeepJSCC for images [2] focused only on the distortion of the reconstructed image with respect to (w.r.t.) the input. On the other hand, there has been significant progress in recent years in generative models that generate realistic images with better perception qualities. In [27], adversarial loss has been used for DeepJSCC to improve the perceptual quality of the reconstructed images; however, this has achieved limited success due to the difficulties in training, such as maintaining a balance between the loss components and ensuring stable training dynamics. Alternatively, the authors of [47] employed StyleGAN2, a powerful pre-trained generative model to improve the perceptual quality of the image reconstructed by DeepJSCC by modeling the whole encoder-channel-decoder pipeline as a forward process, and

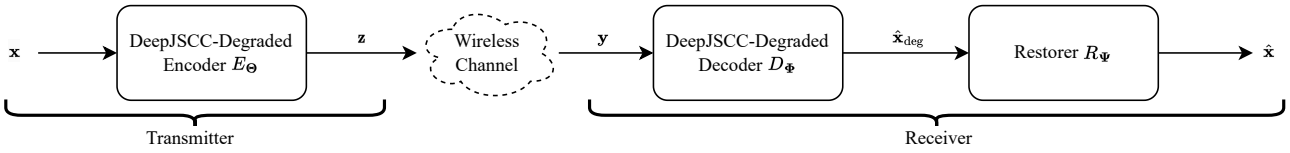


Figure 3.1: Overview of the image transmission procedure using our method.

modeling the image reconstruction as an inverse problem. With this approach, it is possible to deploy a DeepJSCC encoder–decoder pair trained over a generic image dataset, and refine its output by employing a generative model solely at the receiver. In this chapter, we follow a similar approach, but use a diffusion-based generative model at the decoder. Our method outperforms both standard DeepJSCC and generative adversarial network (GAN)-based DeepJSCC of [47] on both perception-oriented and distortion-oriented metrics.

Our main contributions are summarized as follows:

1. We introduce the first denoising diffusion probabilistic models (DDPM)-based DeepJSCC scheme for wireless image delivery that utilizes a novel controlled degradation and restoration-based formulation.

2. Through an extensive set of experiments, we demonstrate that our method outperforms conventional DeepJSCC and previous state-of-the-art generative learning-based DeepJSCC for all evaluated SNR and bandwidth conditions.

3. To facilitate further research and reproducibility, we provide the source code of our framework and simulations on github.com/ipc-lab/deepjscc-diffusion.

3.2 System Model and Problem Definition

Here, we describe the problem and our novel methodology that combines diffusion models with modified DeepJSCC. We decompose our problem into two stages: (1) autoencoding stage, and (2) restoration stage, which are summarized in Figure 3.1.

3.2.1 System Model

We build upon the point-to-point DeepJSCC system model described in Section 2.1.1, considering wireless image transmission over an AWGN channel. The transmitter maps an input image $\mathbf{x} \in \mathbb{I}^{C_{\text{in}} \times W \times H}$ into a complex-valued latent vector $\tilde{\mathbf{z}} = E_{\boldsymbol{\Theta}}(\mathbf{x}, \sigma)$ subject to the power constraint in Eq. (2.2), and transmits it over the channel as described in Eq. (2.3).

Unlike the standard DeepJSCC decoder, a nonlinear decoding function $D_{\boldsymbol{\Phi}} : \left(\mathbb{C}^k, \mathbb{R}\right) \rightarrow \mathbb{I}^{C'_{\text{in}} \times W' \times H'}$ at the receiver, parameterized by $\boldsymbol{\Phi}$, reconstructs a *degraded* image from the channel output, i.e., $\hat{\mathbf{x}}_{\text{deg}} = D_{\boldsymbol{\Phi}}(\mathbf{y}, \sigma)$. A restorer $R_{\boldsymbol{\psi}}$ then enhances the image via $\hat{\mathbf{x}} = R_{\boldsymbol{\psi}}\left(\hat{\mathbf{x}}_{\text{deg}}\right)$. Thus, given the channel output $\mathbf{y}$, the flow of data at the receiver becomes $\mathbf{y} \xrightarrow{D_{\boldsymbol{\Phi}}} \hat{\mathbf{x}}_{\text{deg}} \xrightarrow{R_{\boldsymbol{\psi}}} \hat{\mathbf{x}}$. Note that, in general, these two steps can be combined into a single decoding step. However, similarly to [47], we are interested in refining the degraded image reconstructed by a generic DeepJSCC decoder modeled by $D_{\boldsymbol{\Phi}}$ to improve its perceptual quality.

Given the bandwidth ratio ρ and SNR (defined in Section 2.1.1), the goal in general is to minimize the average distortion between the original image $\mathbf{x}$ at the transmitter and the reconstructed image $\hat{\mathbf{x}}$ at the receiver, i.e.,

$$\underset{\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{\psi}}{\arg\min}\, \mathbb{E}_{r(\mathbf{x}, \hat{\mathbf{x}})}\left[d(\mathbf{x}, \hat{\mathbf{x}})\right], \tag{3.1}$$

where $r(\mathbf{x}, \hat{\mathbf{x}})$ is the joint distribution of original image and the reconstructed image and $d(\mathbf{x}, \hat{\mathbf{x}})$ can be any distortion metric. As mentioned, we employ DeepJSCC as the code block, which has been trained to maximize the average PSNR (defined in Section 2.7.1). It is known that PSNR is not aligned with human perception in general. Hence, we will also consider the LPIPS metric [57] (see Section 2.7.4), which has been shown to better match human perception.

3.3 Methodology

In this section, we describe our novel methodology to exploit image restoration for efficient image transmission using DeepJSCC. Algorithm 1 summarizes the training methodology of the

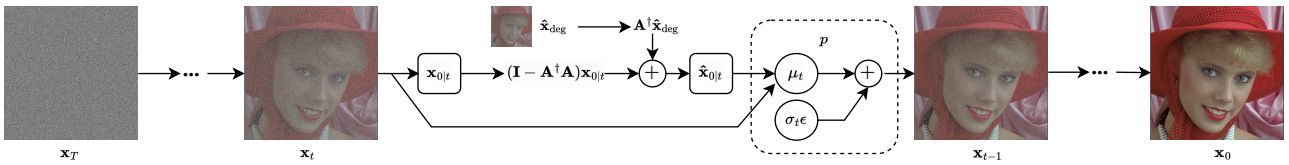


Figure 3.2: Overview of the image restoration procedure by $R_{\boldsymbol{\psi}}$.

Algorithm 1: Our JSCC-based image transmission procedure

▷ *At the transmitter* ◁

$\tilde{\mathbf{z}} = E_{\boldsymbol{\Theta}}(\mathbf{x}, \sigma)$ ▷ *Encoding*

$\mathbf{z} = \sqrt{kP_{\text{avg}}/\left\|\tilde{\mathbf{z}}\right\|_2^2}\tilde{\mathbf{z}}$ ▷ *Power normalization*

▷ *AWGN channel* ◁

$\mathbf{y} = \mathbf{z} + \mathbf{n}$ ▷ *Received signal*

▷ *At the receiver* ◁

$\hat{\mathbf{x}}_{\text{deg}} = D_{\boldsymbol{\Phi}}(\mathbf{y}, \sigma)$ ▷ *Reconstruction of the degraded image*

$\hat{\mathbf{x}} = R_{\boldsymbol{\psi}}(\hat{\mathbf{x}}_{\text{deg}})$ ▷ *Restoration of the degraded image*

return $\hat{\mathbf{x}}$

introduced method, named DeepJSCC-Diffusion.

3.3.1 Transmission of the Degraded Image

We treat the process from the input image to the reconstructed degraded image as a forward process, and we formulate the restoration as an inverse problem. We would like to approximate the impact of the forward process as a known linear transform that we can invert, $\mathbf{A} \in \mathbb{R}^{d\times D}$.[1] Hence, the aim of DeepJSCC-Degraded is to transmit a degraded version of $\mathbf{x}$, denoted by $\mathbf{x}_{\text{deg}} \in \mathbb{R}^{d\times 1}$, where

$$\mathbf{x}_{\text{deg}} = \mathbf{A}\mathbf{x}.$$

Degradation matrix $\mathbf{A}$ can be any linear operator, such as decolorization or mean-pooling-based downsampling [5]. Ideal $\mathbf{A}$ should have the following properties:

1. It should reduce the amount of information to be transmitted so that $\mathbf{x}_{\text{deg}}$ can be recovered at the receiver with minimal reconstruction error, denoted by $\hat{\mathbf{x}}_{\text{deg}}$.

[1] We consider flattened versions of $\mathbf{x}$, $\mathbf{x}_{\text{deg}}$, $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}$ for simplicity of the notation, e.g., we map $\mathbf{x} \in \mathbb{R}^{C_{\text{in}}\times W\times H}$ to $\mathbf{x} \in \mathbb{R}^{D\times 1}$, where $D = C_{\text{in}}WH$.

2. It should preserve perceptual invariances of the image so that $\hat{\mathbf{x}}_{\text{deg}}$ can be restored by $R_{\boldsymbol{\psi}}$ yielding a consistent and perceptually high-quality image $\hat{\mathbf{x}}$.

When $\mathbf{A} = \mathbf{I}$, our scheme for autoencoding stage is equivalent to standard DeepJSCC. However, note that, when the channel SNR and bandwidth ratio ρ are low, the receiver cannot always recover the desired $\mathbf{x}_{\text{deg}}$, and additional reconstruction error is introduced, following an unknown distribution. Our goal by introducing $\mathbf{A}$ is to obtain a known degradation operator to remove the additional noise and to allow good restoration. For instance, a lower resolution image can be transmitted more reliably on the same channel w.r.t. its higher resolution version, yet, it requires further restoration at the receiver to recover its high-resolution version with better perceptual quality.

3.3.2 Restoration of the Degraded Image

Here, we describe the restoration stage that reverts the known degradation modelled by $\mathbf{A}$ and improves its perceptual quality while remaining as consistent with $\hat{\mathbf{x}}_{\text{deg}}$ as possible.

We denote the pseudo-inverse of $\mathbf{A}$ via $\mathbf{A}^{\dagger}$, which satisfies $\mathbf{A}\mathbf{A}^{\dagger}\mathbf{A} \equiv \mathbf{A}$, and can be solved in matrix form using singular value decomposition (SVD). For instance, $\mathbf{A}^{\dagger}$ corresponds to upsampling matrix for the downsampling-based degradation operator $\mathbf{A}$. Given degraded image $\mathbf{x}_{\text{deg}}$, image restoration (IR) aims to yield $\hat{\mathbf{x}}$ that does not introduce significant distortion while increasing the perceptual quality of the image [5]. For the former, we introduce a consistency constraint that requires $\mathbf{A}\hat{\mathbf{x}} \equiv \mathbf{x}_{\text{deg}}$, whereas the perceptual quality requires $\hat{\mathbf{x}} \sim q(\mathbf{x})$, where $q(\mathbf{x})$ represents the ground truth (GT) image distribution.

We employ DDPM, denoted by $Z_{\boldsymbol{\psi}}$, which is trained through a progressive denoising task. DDPM consists of T-step forward and backward processes. The forward process gradually introduces random noise into the data, whereas the reverse process generates desired data samples from noise. Let $\boldsymbol{\epsilon}_t$ denote the noise in $\mathbf{x}_t$ at time step t. DDPM utilizes a DNN, $Z_{\boldsymbol{\psi}}$, to predict the noise, $\boldsymbol{\epsilon}_t$, i.e., $\boldsymbol{\epsilon}_t = Z_{\boldsymbol{\psi}}(\mathbf{x}_t, t)$ is the estimation of $\boldsymbol{\epsilon}_t$ at time-step t. We refer the reader to [5, 4] for further details on DDPMs. Although DDPMs are powerful unconditional

Algorithm 2: Image restoration procedure R_{ψ}.

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ *Initialize from pure noise*
for $t = T, ..., 1$ **do**
  $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - Z_{\psi}(\mathbf{x}_t, t) \sqrt{1 - \bar{\alpha}_t} \right)$ ▷ *Denoising step*
  $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^{\dagger} \hat{\mathbf{x}}_{\text{deg}} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A}) \mathbf{x}_{0|t}$ ▷ *Refine null-space*
  $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$ ▷ *Sample from reverse process*
return $\mathbf{x}_0$

image generators, generating consistent images is challenging [4, 5]. We employ zero-shot image restoration method denoising diffusion null-space model (DDNM), which utilizes and guides the DDPM model Z_{ψ} for the restoration task to improve quality while preserving consistency. Note that since DDNM utilizes a pre-trained DDPM, it does not require specific training for restoration task [5].

Any sample $\mathbf{x}$ can be decomposed into $\mathbf{x} \equiv \mathbf{A}^{\dagger}\mathbf{A}\mathbf{x} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{x}$, called range-null space decomposition, where the two terms correspond to range and null space parts of the image, respectively. We denote the estimated $\mathbf{x}_0$ at diffusion time-step t as $\mathbf{x}_{0|t}$. For a degraded image $\mathbf{x}_{\text{deg}}$, we can construct a solution for $\hat{\mathbf{x}}$ that satisfies the consistency constraint:

$$\hat{\mathbf{x}} = \mathbf{A}^{\dagger}\mathbf{x}_{\text{deg}} + (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A})\mathbf{x}_{0|t},$$

where $\mathbf{x}_{0|t}$ is iteratively refined via the procedure shown in Figure 3.2 and Algorithm 2, where $\bar{\alpha}$ is a hyperparameter determining the noise schedule and p is the forward diffusion process distribution (see [5] for more details). Since modifying $\mathbf{x}_{0|t}$ does not affect the consistency constraint, the restoration procedure does not break the consistency of the transmitted image. Therefore, we refine the image to increase its perceptual quality while keeping it consistent to the version reconstructed by the DeepJSCC decoder with a known degradation. Finally, we unflatten $\hat{\mathbf{x}}$ back to dimensions $C_{\text{in}} \times W \times H$.

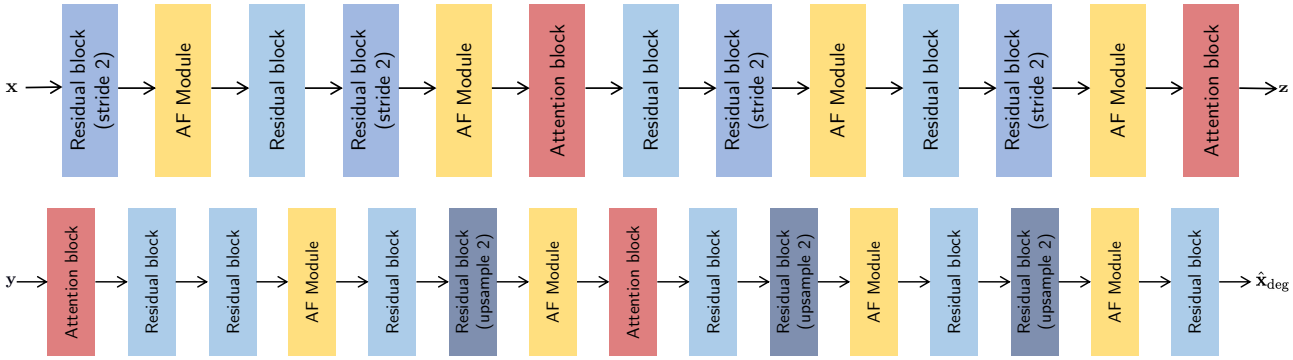


Figure 3.3: The encoder (top) and decoder (bottom) architectures of the employed DeepJSCC scheme.

3.3.3 Autoencoder Network Architecture and Training

Figure 3.3 shows the details of the employed CNN-based autoencoder architecture for DeepJSCC, which builds upon the common backbone described in Section 2.1.2. For fair comparison, we employ the same neural network (NN) architecture with [47] as our autoencoder, except that we replace the first residual upsample block of the decoder with a residual block to produce the degraded image $\hat{\mathbf{x}}_{\text{deg}}$.

Notice that, unlike the current DeepJSCC architectures [2, 32], our CNN-based autoencoder reconstructs the degraded image $\hat{\mathbf{x}}_{\text{deg}}$, which is later restored by $R_{\mathbf{\Psi}}$. We denote degradation matrix $\mathbf{A}$ to be the average pooling matrix (with downsampling factor of 2 on both dimensions), so we have $2W^{'} = W$, $2H^{'} = H$ and $C_{\text{in}} = C_{\text{in}}^{'}$. The reason for this choice is that we can reliably revert this degradation and obtain a high-quality image via restoration as described in Section 3.3.2.

We first train the encoder and decoder parameters $\mathbf{\Theta}$ and $\mathbf{\Phi}$ to minimize the MSE loss:

$$\mathcal{L}\left(\mathbf{x}, \hat{\mathbf{x}}_{\text{deg}}\right) = \frac{1}{C_{\text{in}}WH} \left\|\mathbf{A}\mathbf{x} - \hat{\mathbf{x}}_{\text{deg}}\right\|_2^2,$$

which aids our DeepJSCC-Degraded model to transmit degraded instances with minimal distortion w.r.t. the known degradation modelled by $\mathbf{A}$.

3.4 Numerical Results

In this section, we present our experimental setup and demonstrate the performance gains of our method.

3.4.1 Experimental Setup

We evaluate our method on 512x512 CelebA-HQ dataset, which contains 30 000 high-resolution images [41]. We split the dataset using the same procedure as in [47], i.e., split as $8:1:1$ for training, validation, and testing, respectively. We use the PSNR and LPIPS metrics (defined in Section 2.7) to evaluate the quality of the reconstructed images.

3.4.2 Implementation Details

We have conducted the experiments using Pytorch [129]. We use the same hyperparameters and the same architecture for all the methods. We set the learning rate to 1×10^{-4}, number of filters in middle layers to 128, batch size to 64 and the power constraint to $P_{\text{avg}} = 1$. We use Adam optimizer [130]. We continue training until no improvement is achieved for consecutive 10 epochs. During training and validation, we run the model using different SNR values for each instance, uniformly chosen from $[-5, 5]$ dB. We test and report the results for each SNR value using the same model thanks to the AF module. We shuffle the training pairs or instances randomly before each epoch.

In the restoration step, we employ 512×512 unconditional Imagenet DDPM, which is fine-tuned for 8100 steps from OpenAI's class-conditional DDPM [4]. We set the number of diffusion time steps to $T = 1000$ with linear schedule. We also utilize the time-travel trick in [5] with 100 sampling time steps.

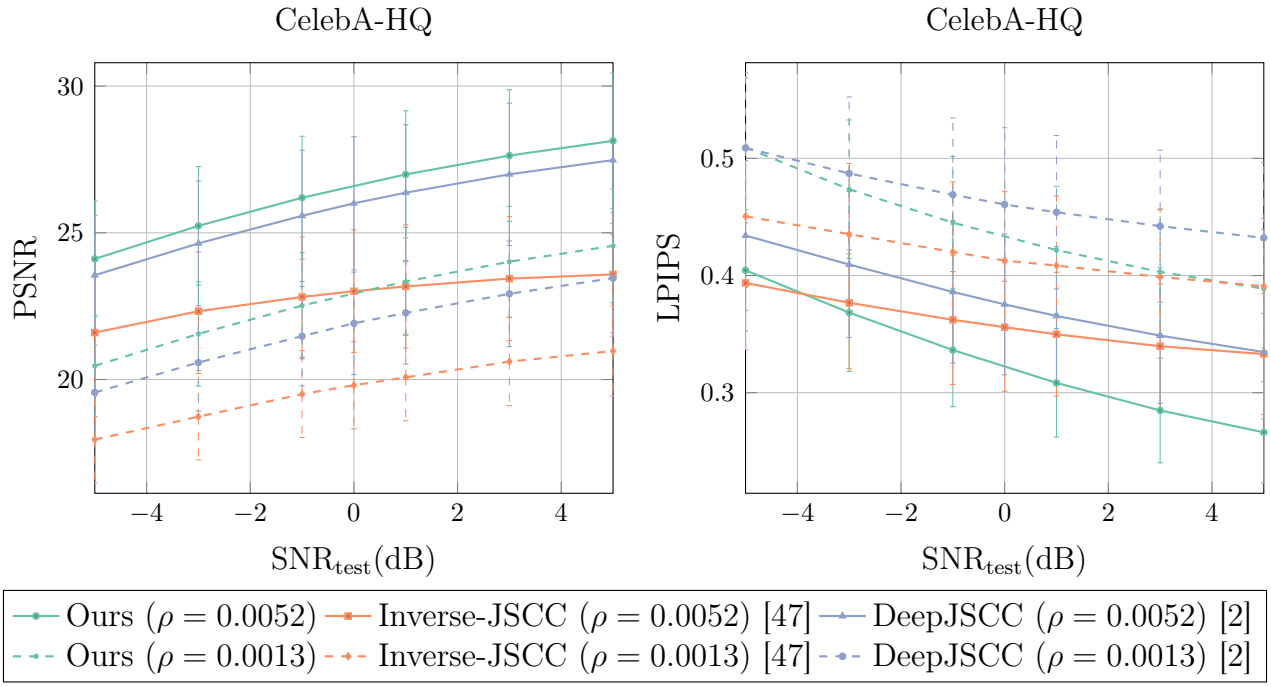


Figure 3.4: PSNR and LPIPS comparison of our method with the baselines for $\rho \in \{0.0013, 0.0052\}$ over different SNRs.

3.4.3 Results

In this section, we show the superiority of our method over standard DeepJSCC and GAN-based DeepJSCC [47], named Inverse-JSCC. Since previous studies have already shown that DeepJSCC outperforms classical separation-based methods, we do not consider them as baselines [2].

Figure 3.4 shows the comparisons in terms of PSNR and LPIPS metrics, respectively. We highlight that we consider an extremely challenging communication scenario with a very low bandwidth ratio. Our method clearly improves w.r.t. both DeepJSCC and Inverse-JSCC at all evaluated values of $\text{SNR}_{\text{test}} \in [-5\,, 5\,]$ dB and $\rho \in \{0.0013, 0.0052\}$. Figure 3.5 shows an example set of reconstructed images for qualitative comparison. It is clear that the proposed method generates more realistic images, but surprisingly, it also achieves a much lower distortion at all channel conditions. We note that our method improves upon DeepJSCC even in terms of the distortion metric, PSNR. This shows that it is beneficial for the DeepJSCC to target a lower resolution version of the image, which can then be improved at the receiver using the diffusion model.

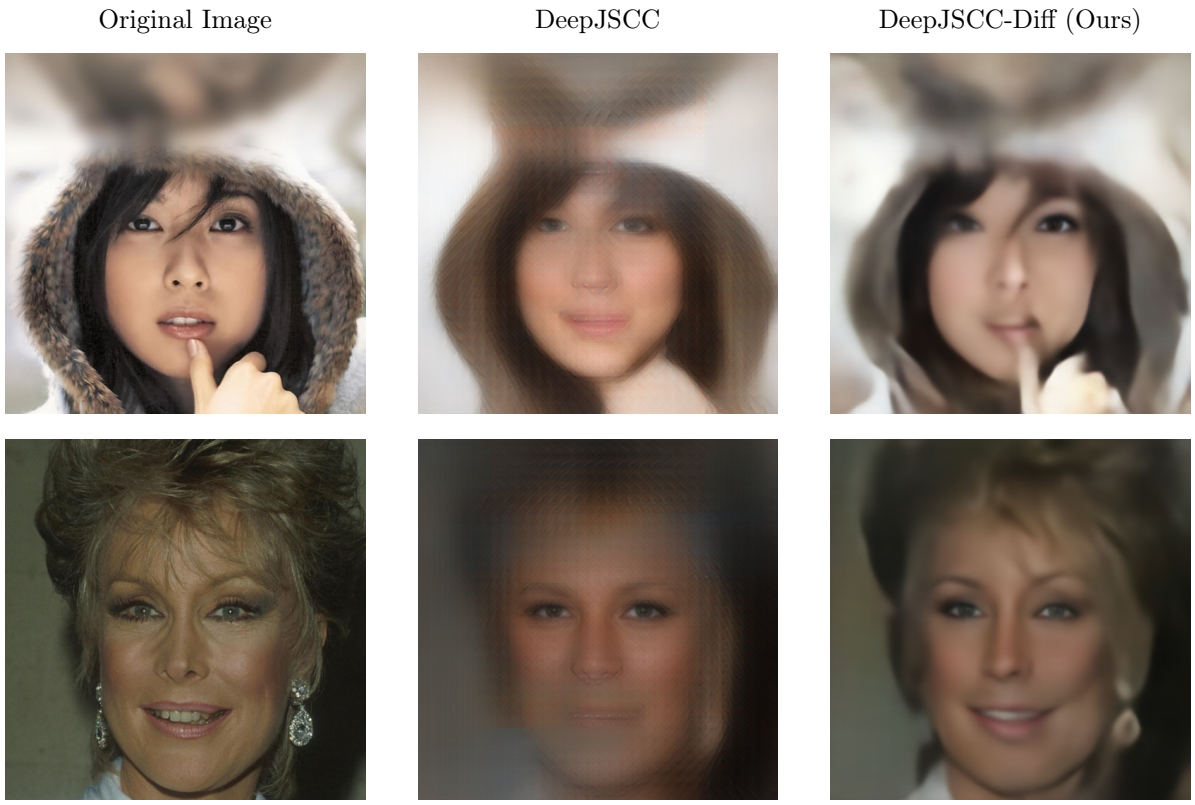


Figure 3.5: Qualitative comparison of the reconstructed images from CelebA-HQ dataset for $\rho = 0.0013$ and $\text{SNR}_{\text{test}} = 3$ dB.

3.5 Conclusion

We have introduced a novel generative communication scheme for image transmission over noisy wireless channels, which promotes realness and consistency through controlled degraded image transmission followed by restoration. The introduced method employs DDPM in addition to a modified DeepJSCC, which aims at transmitting a degraded image with the degradation modeled as a known linear transform. We have shown that the proposed scheme outperforms the standard DeepJSCC and the state-of-the-art GAN-based DeepJSCC. While we have considered a specific linear transform for the DeepJSCC encoder and decoder pair in this work, we will consider its optimization in our future work.

Chapter 4

Distributed Deep Joint Source-Channel Coding with Decoder-Only Side Information

4.1 Introduction

As outlined in Chapter 1, this thesis addresses the second research objective by developing JSCC techniques that exploit source correlations. Building upon the DeepJSCC and distributed source coding principles established in Chapter 2, this chapter extends these approaches to scenarios with decoder-only side information, enabling more efficient transmission of correlated images. While the previous chapter considered a single-user point-to-point channel, this chapter can be broadly viewed as a two-user setting where one user transmits its image and a correlated image from a second source is available as side information at the decoder.

In this chapter, we are interested in the scenario where the receiver has access to a correlated *side information* sequence (see Fig. 4.1). For example, consider a distributed network of cameras transmitting images to a joint central processing unit. In such scenarios, images are highly correlated; hence, transmitting each image independently results in significant communication

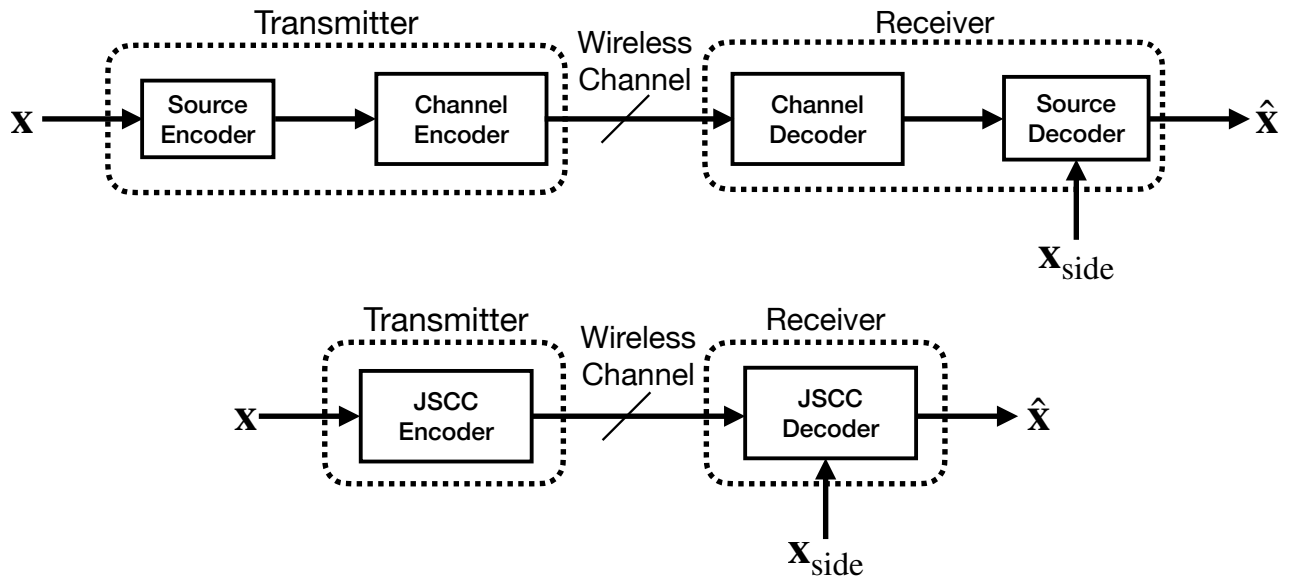


Figure 4.1: Separation-based (**top**) vs. JSCC-based (**bottom**) communication schemes, having decoder-only side information.

overhead. As a first step towards exploiting source correlations in large networks, we consider the simple case with two cameras, where one camera sends its image to another.

It is known that separate source and channel coding remains to be optimal in the case of communication with decoder-only side information [131]. This is achieved by Wyner-Ziv lossy compression of the source sequence exploiting decoder's side information through binning [7], followed by capacity-achieving channel coding. However, the price to pay for near-optimal performance is high complexity and delay since achieving the channel capacity and the Wyner-Ziv rate–distortion function necessitates large blocklengths. We instead leverage the universal function approximation capability of neural networks [132] to find constructive solutions for JSCC with decoder-only side information in the non-asymptotic regime.

Our goal is to design a practical JSCC scheme that can benefit from the side information at the receiver. Our main contributions can be summarized as follows:

- To the best of our knowledge, we introduce the first DeepJSCC-based image transmission method that explicitly exploits decoder-only side information, termed *DeepJSCC-WZ*. The proposed transmission scheme is an important building block towards fully distributed practical DeepJSCC schemes for correlated image/video signals.

- We demonstrate that our method significantly outperforms both the point-to-point DeepJSCC scheme (with no side information) and the separation-based scheme with side information, in terms of both traditional and perception-oriented fidelity metrics for all the considered channel SNR and bandwidth ratio (BR) values.

- As an upper bound, we also provide a solution for the scenario in which the side information is available at both the encoder and decoder. This allows us to quantify the performance gain by providing the side information also to the encoder.

- To facilitate further research and reproducibility, we also provide the source code of our framework and simulations on `github.com/ipc-lab/deepjscc-wz`.

4.2 System Model and Problem Formulation

We extend the point-to-point DeepJSCC system model in Section 2.1.1 to the setting where the receiver has access to correlated side information. See Fig. 4.1 for an illustration. Specifically, we focus on pairs of *stereo images* with overlapping fields of view as correlated information sources $\mathbf{x}, \mathbf{x}_{\text{side}} \in \mathbb{I}^{C_{\text{in}} \times W \times H}$. The transmitter encodes the input image into a complex-valued latent vector $\mathbf{z} = E_{\boldsymbol{\Theta}}(\mathbf{x}, \sigma^2)$ subject to the power constraint in Eq. (2.2), and transmits it over the AWGN channel as described in Eq. (2.3).

At the receiver, a nonlinear decoding function $D_{\boldsymbol{\Phi}} : \mathbb{C}^k \to \mathbb{I}^{C_{\text{in}} \times W \times H}$, parameterized by $\boldsymbol{\Phi}$, reconstructs the input image using both the channel output $\mathbf{y}$ and the side information:

$$\hat{\mathbf{x}} = D_{\Phi}(\mathbf{y}, \mathbf{x}_{\text{side}}, \sigma^2).$$

Given the bandwidth ratio ρ and the channel SNR (defined in Section 2.1.1), our learning objective is to minimize the average distortion between the original input image $\mathbf{x}$ at the

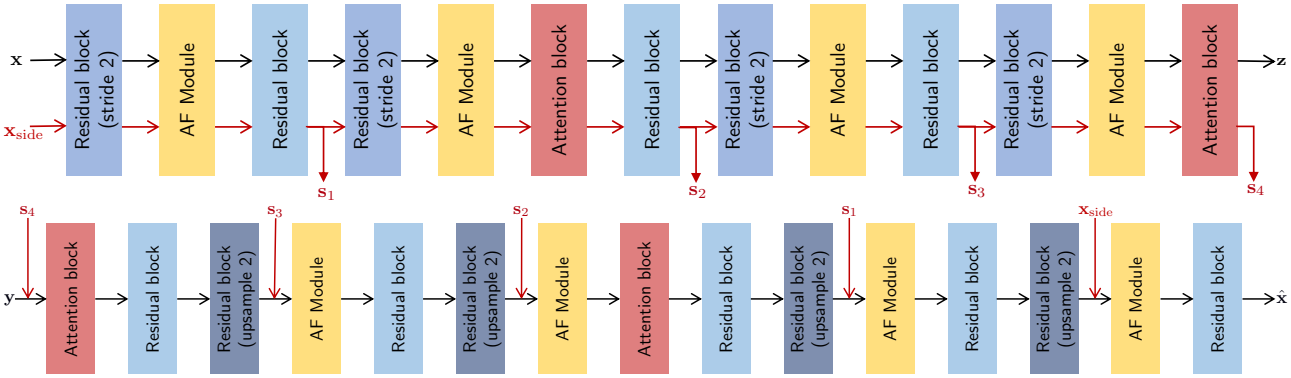


Figure 4.2: Encoder architecture of our method (**top**), which is used to encode both the input image $\mathbf{x}$ at the transmitter and the side information image $\mathbf{x}_{\text{side}}$ at the receiver side. Red arrows indicate the flow of the encoded side information. $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and $\mathbf{s}_4$ denote the encoded side information at different scales, which are to be used at the receiver side. Decoder architecture of our method (**bottom**), which is used to reconstruct the input image from the noisy channel output $\mathbf{y}$ and the side information $\mathbf{x}_{\text{side}}$.

transmitter and the reconstructed image $\hat{\mathbf{x}}$ at the receiver, i.e.,

$$\arg\min_{\boldsymbol{\Theta},\boldsymbol{\Phi}} \mathbb{E}\left[d\left(\mathbf{x},\hat{\mathbf{x}}\right)\right],$$

where the expectation is over the source and side information statistics, $(\mathbf{x},\mathbf{x}_{\text{side}}) \sim p(\mathbf{x},\mathbf{x}_{\text{side}})$, as well as the channel noise distribution. Here, $d(\cdot,\cdot)$ can be any differentiable distortion measure.

4.3 Methodology

In this section, we describe our novel architecture that we build upon the state-of-the-art DeepJSCC architecture. While the previous chapter applied DeepJSCC to single-user point-to-point transmission, this chapter addresses a distributed setting with decoder-only side information. Our DeepJSCC-WZ architecture incorporates a dedicated side-information encoder at the receiver, in addition to the transmitter's source encoder, and employs multi-scale feature fusion in the decoder, enabling more efficient transmission in correlated source settings.

Algorithm 3: Training procedure of DeepJSCC-WZ

repeat ▷ *Iterate through epochs*
  Shuffle $\mathcal{D}_{\text{train}}$ and set $l = t = 0$
  for all $(\mathbf{x}, \mathbf{x}_{\text{side}}) \in \mathcal{D}_{\text{train}}$ **do**
    ▷ *At the transmitter* ◁
    Calculate σ^2 via Eq. (2.6) for SNR $\sim$ Uniform $[-5, 5]$
    $\tilde{\mathbf{z}} = E_{\boldsymbol{\Theta}}\left(\mathbf{x}, \sigma^2\right)$ ▷ *Encoding*
    $\mathbf{z} = \sqrt{kP_{\text{avg}}/\left\|\tilde{\mathbf{z}}\right\|_2^2}\tilde{\mathbf{z}}$ ▷ *Induce power normalization*
    ▷ *AWGN channel* ◁
    Sample $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_k)$
    $\mathbf{y} = \mathbf{z} + \mathbf{n}$ ▷ *Received signal*
    ▷ *At the receiver* ◁
    $\hat{\mathbf{x}} = D_{\boldsymbol{\Phi}}(\mathbf{y}, \mathbf{x}_{\text{side}}, \sigma^2)$ ▷ *Decoding*
    $l = l + \mathcal{L}\left(\mathbf{x}, \hat{\mathbf{x}}\right)$
    t = t + 1
    if t mod *batch size* $= 0$ **then** ▷ *Standard mini-batch training*
      Update $\boldsymbol{\Theta}, \boldsymbol{\Phi}$ by backpropagating l
      Reset l and gradients to zero
until validation PSNR does not improve for e epochs

4.3.1 DeepJSCC-WZ Architecture

Our DNN architecture tackles several challenges at once: learning with side information, high-dimensional inputs, large number of parameters, spatial correlations, and SNR adaptivity. Fig. 4.2 illustrates the encoder and decoder architectures we propose for distributed variant of DeepJSCC in the Wyner-Ziv setting, i.e., having decoder-only side information.

We build upon the common DeepJSCC autoencoder backbone described in Section 2.1.2. The AF module requires SNR as an input, which is calculated via Eq. (2.6). We randomly sample SNR values during training to enable generalization (see Algorithm 3). Reliably incorporating information from multiple views or modalities for learning-based methods, such as DNNs, has been known to be challenging. Fusing multiple views can be done at the data, feature or output levels [109, 110]. In [65], the extracted side information features are concatenated with the received compressed features only before being fed to the decoder. Inspired by these works, we opt for fusing $\mathbf{x}_{\text{side}}$ to our decoder at four different stages with different scales, including the encoder's intermediate features and its output. This way, we are able to incorporate both high-level details, such as the presence of objects, from low-resolution features, and fine-grained details, such as the texture of pixels, from the higher resolution features.

In the *DeepJSCC-WZ* network, we incorporate the decoder-only side information by employing the same encoder architecture at the receiver, but feed it with $\mathbf{x}_{\text{side}}$ as input, instead of the

original source image. This encoder's intermediate outputs are given to the decoder at different scales, as shown in Fig. 4.2. As seen, $\mathbf{s}_4$ is concatenated at the filter dimension of the noisy channel output $\mathbf{y}$. Also, $\mathbf{s}_3, \mathbf{s}_2$ and $\mathbf{s}_1$ are concatenated after the consecutive upsampling layers in the decoder. Lastly, the side information $\mathbf{x}_{\text{side}}$ is concatenated with the last upsampling layer's output at the filter dimension. The encoder output is power-normalized as described in Section 2.1.2 to satisfy the constraint in Eq. (2.2).

4.3.2 Training Loss

Algorithm 3 shows the overall training procedure DeepJSCC-WZ, which is trained in an unsupervised fashion using only the raw image pairs, $(\mathbf{x}, \mathbf{x}_{\text{side}}) \in \mathcal{D}_{\text{train}}$, where $\mathcal{D}_{\text{train}}$ is the set of training data pairs. As such, it does not rely on any costly human labelling. We randomly generate and use the channel model throughout training as described in Section 4.3.1. We train the whole DNN architecture end-to-end with the task of reconstructing the input image $\mathbf{x}$ by minimizing the prescribed distortion measure $d(\cdot, \cdot)$. Note that DeepJSCC models can be directly optimized for any differentiable distortion measure, and can be easily adapted to various multi-media transmission tasks (e.g., video as in [28]). Following the reasoning in [133], we choose a composite loss function, $\mathcal{L}$, to jointly optimize the *distortion-perception trade-off* [47, 22] for the reconstructed images as:

$$\mathcal{L}\left(\mathbf{x}, \hat{\mathbf{x}}\right) = \sum_{(\mathbf{x}, \mathbf{x}_{\text{side}}) \in \mathcal{D}_{\text{train}}} \text{MSE}\left(\mathbf{x}, \hat{\mathbf{x}}\right) + \lambda \cdot \text{LPIPS}\left(\mathbf{x}, \hat{\mathbf{x}}\right), \tag{4.1}$$

where $\text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \frac{1}{m} \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|_2^2$ is the MSE loss, where m is the total number of elements in $\mathbf{x}$, i.e., $m = C_{\text{in}}WH$, and LPIPS [57] is a commonly used perceptual quality metric (see Section 2.7.4). Here, λ is the trade-off parameter, determining the relative importance of LPIPS loss w.r.t. the MSE loss. Unlike the MSE distortion metric (which is applied point-wise), perception-oriented distortion criteria, such as LPIPS and MS-SSIM [128] (see Section 2.7), measure the similarity between the original image and the reconstructed one using a proxy for the human perception.

4.4 Numerical Results

In this section, we present our experimental setup to show the performance gains of our method in different scenarios.

4.4.1 Datasets and Baselines

For the first part of the experiments, we use the *Cityscapes* dataset [134], consisting of 5000 stereo image pairs, where 2975 pairs are used for training, and 500 and 1525 pairs were used for validation and test, respectively. For the second part of experiments, following [135, 65], we compose our dataset from KITTI 2012 [136] and KITTI 2015 datasets [137, 138], termed as *KITTIStereo*, where 1576 pairs are used for training, and we validated and tested models on 790 image pairs each.

To quantify how much our scheme can benefit from the decoder-only side information, we first compare our method with conventional *DeepJSCC*, which ignores the side information. In addition, we also experiment with further reduction in the number of parameters by using the same set of parameters for encoding both the input image $\mathbf{x}$ at the transmitter, and $\mathbf{x}_{\text{side}}$ at the receiver side, called *DeepJSCC-WZ-sm*. In other words, unlike *DeepJSCC-WZ*, the encoding functions at the transmitter and at the receiver share the same parameters $\boldsymbol{\Theta}$. DeepJSCC-WZ-*sm* also receives binary indicator of encoding either $\mathbf{x}$ or $\mathbf{x}_{\text{side}}$ to encoder's AF modules. For a complete comparison, we further introduce a model named *DeepJSCC-Cond*, where the transmitter *also* has access to $\mathbf{x}_{\text{side}}$ image, which serves as an upper bound on the performance of the model of interest. We also consider separate source and channel coding designs using the neural image compression approach named *DeepNIC+Capacity*, which incorporates side information as in [65] followed by ideal capacity-achieving channel codes. For the performance of DeepNIC+Capacity, we use the results from the original paper [65] and adopt an upper bound for this scheme by equating the reported average rate values in [65] to the capacity of a complex AWGN channel multiplied by the BR value.

Figure 4.3: Visual comparison of reconstructed images from KITTIStereo (**top**) and Cityscapes (**bottom**) datasets, having BR value of $\rho = 1/32$ and $\text{SNR}_{\text{test}} = -4$ dB.

4.4.2 Implementation and Training Details

We utilize standard hyperparameters for our method that has been commonly used in the literature [2, 32]. We employ a learning rate of 1×10^{-4}, a batch size of 32, and an average power constraint of $P_{\text{avg}} = 1.0$ (see Eq. (2.2)) for all the analyzed methods. We use Adam optimizer [130] to minimize the training loss in Equation (4.1). We train the network with channel noise $\mathbf{n}$ determined by the uniformly sampled SNR between -5 and 5 dB. Following [65, 139], we centre-crop each 375×1242 image of the KITTIStereo dataset to obtain images of size 370×740, and then downsample them to 128×256. For the Cityscapes dataset, we directly downsample each image to the size of 128×256.

4.4.3 Comparison with the Baselines

Fig. 4.4 demonstrates the performance gains of our proposed method under different SNR conditions and BR values of $\rho = \{1/16; 1/32\}$, on KITTIStereo and Cityscapes datasets. We highlight that all the JSCC-based models are trained using the composite loss function in Eq. (4.1), but are evaluated across various traditional (e.g., PSNR) and perceptual (e.g., MS-SSIM and LPIPS) distortion quality metrics. For both BR values, DeepJSCC-WZ outperforms its point-to-point counterpart, that is DeepJSCC, as well as its separation-based analogue, that is

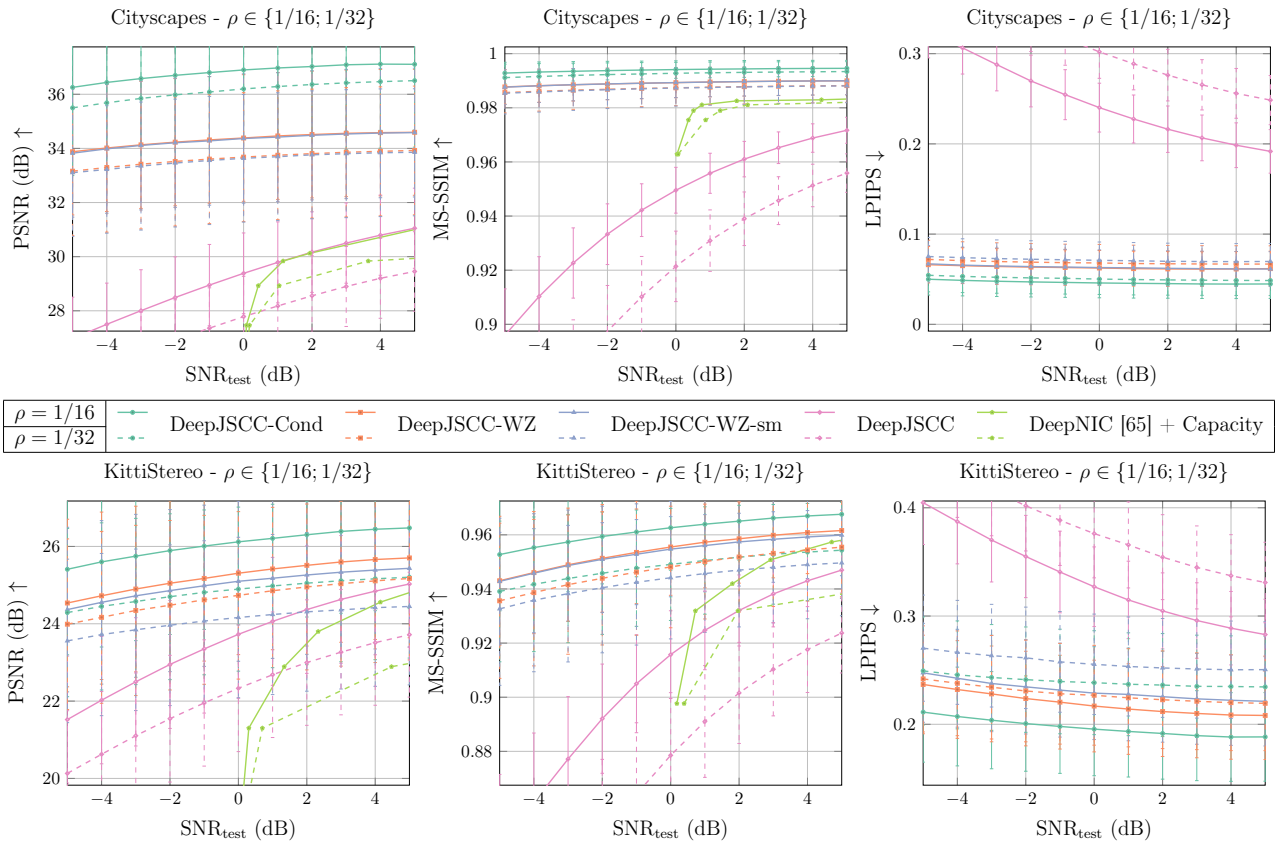


Figure 4.4: Comparison of the introduced DeepJSCC-WZ method with the baselines on the KITTIStereo and Cityscapes datasets for BRs $\rho \in \{1/16; 1/32\}$.

DeepNIC+Capacity, at all the evaluated SNRs in terms of the distortion criteria considered. Unlike the DeepNIC model, we observe that DeepJSCC-WZ does not suffer from the cliff effect and provides a graceful performance degradation as the channel SNR varies.

Notably, we observe a stark performance improvement in the LPIPS distortion metric, which is widely accepted to be more aligned with human perception of image quality (see Fig. 4.3 for some visual examples).

Looking at the performance of DeepJSCC-WZ-sm, we note that imposing the same set of parameters for the encoding of both images, $\mathbf{x}$ and $\mathbf{x}_{\text{side}}$, yields little to no effect on the performance. We also remark that DeepJSCC-WZ achieves a comparable performance with the model DeepJSCC-Cond, whose performance is expected to serve as an upper limit bound on the DeepJSCC-WZ model, considering perceptual distortion criteria such as MS-SSIM and LPIPS. This empirically proves our proposed model's capability of successfully exploiting the decoder-only side information.

Table 4.1: Number of parameters for the compared methods.

Method	$\rho = 1/16$	$\rho = 1/32$
DeepJSCC	31.6M	31.0M
DeepJSCC-WZ-sm	39.8M	38.6M
DeepJSCC-WZ	48.9M	47.4M
DeepJSCC-Cond	59.1M	57.4M

We refer to Table 4.1 for a comparison of the number of parameters of all the methods we have considered. For the two BR values, $\rho = 1/16$ and $\rho = 1/32$, the amount of increase in the number of parameters is $\approx 25\%$, $\approx 53\%$ and $\approx 85\%$ for DeepJSCC-WZ-sm, Deep-JSCC-WZ and DeepJSCC-Cond, respectively, compared to the standard point-to-point DeepJSCC. DeepJSCC-WZ-sm, DeepJSCC-WZ and DeepJSCC-Cond models have additional filter parameters at the decoder, in comparison to DeepJSCC, in order to fuse the encoded $\mathbf{x}_{\text{side}}$ features. DeepJSCC-WZ-sm has less parameters than DeepJSCC-WZ thanks to the parameter sharing of encoders that encode $\mathbf{x}$ and $\mathbf{x}_{\text{side}}$. Therefore, one can opt for the variant DeepJSCC-WZ-sm, instead of the DeepJSCC-WZ model, for a comparable performance in order to keep the number of parameters within a reasonable budget. DeepJSCC-Cond has three different encoder modules, two of which are to encode $\mathbf{x}_{\text{side}}$ at the transmitter and the receiver, while the other one is to encode $\mathbf{x}$ at the transmitter.

4.5 Conclusion

We have introduced a learning-based JSCC scheme for image transmission which is capable of exploiting a decoder-only side information that is in the form of a correlated image. We have demonstrated that the receiver is able to successfully exploit this side information, yielding superior performance over all the channel conditions and distortion criteria considered compared to ignoring the side information. Possible avenues for future work include analyzing the robustness of the proposed approach considering time-varying channel models and correlation structures for $(\mathbf{x}, \mathbf{x}_{\text{side}})$, and also considering the fully distributed communication scenarios using DeepJSCC, which has the potential to be an important ingredient towards realizing the task of

practical image/video transmission over wireless sensor networks.

Chapter 5

Distributed Deep Joint Source-Channel Coding over a Multiple Access Channel

5.1 Introduction

Conventional wireless communication systems consist of two different stages, called source coding and channel coding. As outlined in Chapter 1, this thesis addresses the third research objective by developing JSCC techniques for efficient multi-user communication. While the preceding chapters addressed single-user transmission and a distributed setting with correlated sources, this chapter moves to a fully simultaneous transmission scenario where two independent users transmit over a shared MAC. Building upon the background on JSCC and NOMA established in Chapter 2, this chapter introduces a novel deep learning-based JSCC framework that leverages NOMA for multi-user image transmission over wireless channels.

Shannon proved that such a separate design is without loss of optimality when the source and channel blocklengths are infinite [1]. However, optimality of separation fails in the practical finite blocklength regime, or in multi-user networks in general [140]. Moreover, the suboptimality gap of separation increases as the blocklength gets shorter or the SNR diminishes, which correspond to the operation of many delay-sensitive applications such as IoT or autonomous driving.

Although researchers have been investigating JSCC schemes for many decades [141, 142, 143], these schemes did not find application in practical systems due to their high complexity, and poor performance with practical sources and channel distributions. Research on JSCC has gained renewed interest recently with the adoption of DNNs to implement JSCC [2]. This data-driven approach is based on modeling the end-to-end communication system as an autoencoder architecture. Numerous follow-up studies have extended DeepJSCC in different directions. In [144], the authors demonstrate that DeepJSCC can exploit feedback to improve the end-to-end reconstruction quality. In [145], DeepJSCC architecture is modified by increasing the filter size to improve the performance. In [146], it is shown that DeepJSCC can adopt to varying channel bandwidth with almost no loss in performance. In addition to very promising results achieved for a particular channel quality, it is worth noting that DeepJSCC avoids the *cliff effect* suffered by separate source and channel coding and achieves *graceful degradation* with the channel quality; that is the image is still decodable even when the channel quality falls below the target SNR the system is designed for, albeit with lower quality. Separation-based schemes, on the other hand, suffer from the cliff effect: for a fixed number of channel uses, the channel code cannot be decoded below a certain SNR threshold, resulting in complete failure. While practical systems can partially mitigate this through rate adaptation or retransmissions, these mechanisms trade latency and bandwidth for reliability, and the underlying cliff effect persists.

In this chapter, we consider transmission of distinct images over a MAC. It is known that separation theorem applies to MACs with independent sources; however, the design of practical JSCC schemes for finite blocklengths and real information sources remain as an open challenge. A straightforward approach is to employ TDMA, where each user employs the same trained DeepJSCC network to transmit using half of the channel bandwidth. However, our goal in this chapter is to develop a DeepJSCC framework that can exploit non-orthogonal communications. A related NOMA-aided JSCC scheme is considered for collaborative inference over a MAC in [10], where the goal is to retrieve a unique identity rather than reconstructing distinct images.

Our main contributions are summarized as follows:

1. To the best of our knowledge, we introduce the first multi-user DeepJSCC-based image

transmission method, exploiting NOMA.

2. We introduce a novel Siamese network-based architecture with device embeddings that allows us to use the same set of parameters for both devices, which will be instrumental in scaling our solution to larger networks.

3. We introduce a training data subsampling methodology, and employ CL-based training to further improve the performance.

4. Through an extensive set of experiments, we show that our superposition-based method for NOMA outperforms traditional DeepJSCC with time division for all evaluated SNR conditions. We also show that our method is fair; that is, no significant difference is observed between the performances of the two users.

5. To facilitate further research and reproducibility, we provide the source code of our framework and simulations on github.com/ipc-lab/deepjscc-noma.

5.2 System Model and Problem Formulation

We extend the point-to-point DeepJSCC system model in Section 2.1.1 to a two-user uplink MAC setting. Each transmitter $i \in \{1, 2\}$ maps its input image $\mathbf{x}_i \in \mathbb{I}^{C_{\text{in}} \times W \times H}$ into a complex-valued latent vector $\mathbf{z}_i = E_{\boldsymbol{\Theta}_i,\sigma}(\mathbf{x}_i) \in \mathbb{C}^k$, subject to the average power constraint in Eq. (2.2).

The receiver observes the superposition of both transmitted signals:

$$\mathbf{y} = \mathbf{z}_1 + \mathbf{z}_2 + \mathbf{n},$$

where $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_k)$ is i.i.d. complex Gaussian noise. We assume σ is known at both the transmitters and the receiver.

A single nonlinear decoding function $D_{\boldsymbol{\Phi},\sigma} : \mathbb{C}^k \to \mathbb{I}^{2 \times C_{\text{in}} \times W \times H}$, parameterized by $\boldsymbol{\Phi}$, jointly

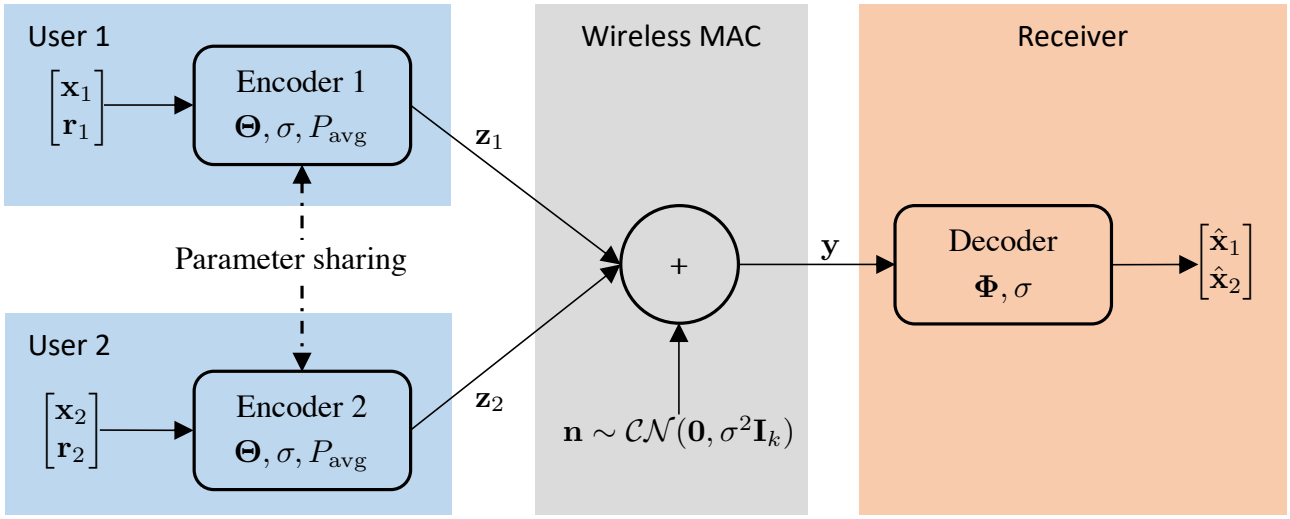


Figure 5.1: Overall architecture of our DeepJSCC-NOMA method. The device embeddings $\mathbf{r}_1$ and $\mathbf{r}_2$ are introduced in Section 5.3.2.

reconstructs both images from the superposed channel output:

$$\begin{bmatrix}\hat{\mathbf{x}}_1\\ \hat{\mathbf{x}}_2\end{bmatrix} = D_{\Phi,\sigma}(\mathbf{y}).$$

Given the bandwidth ratio ρ and channel SNR (defined in Section 2.1.1), our objective is to maximize the average PSNR (defined in Section 2.7.1) on an unseen target dataset. To achieve this goal, in the next section, we will introduce our framework to implement joint training of the neural network based decoder $D_{\boldsymbol{\Phi},\sigma}$ and encoders $E_{\boldsymbol{\Theta}_i,\sigma},\ i \in \{1,2\}$.

5.3 Methodology

In this section, we describe our novel methodology to exploit NOMA for efficient image transmission using DeepJSCC. Algorithm 4 summarizes the training methodology of the introduced method, called DeepJSCC-NOMA.

Algorithm 4: Training Procedure of DeepJSCC-NOMA

Construct $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ ▷ *See Section 5.3.3*
Initialize $\mathbf{r}_1$ and $\mathbf{r}_2$ from $\mathcal{N}(0,1)$ ▷ *Device embeddings*
repeat ▷ *Iterate through epochs*
  Shuffle $\mathcal{D}_{\text{train}}$ and set $\mathcal{L}=0$
  for all $(\mathbf{x}_1,\mathbf{x}_2)\in\mathcal{D}_{\text{train}}$ **do**
    Calculate σ via Eq. (2.6) for SNR $\sim$ Uniform$[0,20]$
    for $i=1,2$ **do** ▷ *Device with index i*
      $\tilde{\mathbf{x}}_i=\begin{bmatrix}\mathbf{x}_i & \mathbf{r}_i\end{bmatrix}^T$
      $\mathbf{z}_i=E_{\boldsymbol{\Theta},\sigma}(\tilde{\mathbf{x}}_i)$ ▷ *Encoding*
      $\mathbf{z}_i=\sqrt{kP_{\text{avg}}/\left\|\mathbf{z}_i\right\|_2^2}\mathbf{z}_i$ ▷ *Power normalization*
    ▷ *Transmit $\mathbf{z}_1$ and $\mathbf{z}_2$ via same channel* ◁
    Sample $\mathbf{n}\sim\mathcal{CN}(\mathbf{0},\sigma^2\mathbf{I}_k)$
    $\mathbf{y}=\mathbf{z}_1+\mathbf{z}_2+\mathbf{n}$ ▷ *Received signal*
    $\begin{bmatrix}\hat{\mathbf{x}}_1 & \hat{\mathbf{x}}_2\end{bmatrix}^T=D_{\Phi,\sigma}(\mathbf{y})$ ▷ *Decoding*
    $\mathcal{L}=\mathcal{L}+\text{MSE}(\mathbf{x}_1,\hat{\mathbf{x}}_1)+\text{MSE}(\mathbf{x}_2,\hat{\mathbf{x}}_2)$
    if t mod *batch size* $=0$ **then**
      ▷ *Standard mini-batch training* ◁
      Update $\mathbf{r}_1,\mathbf{r}_2,\boldsymbol{\Theta},\boldsymbol{\Phi}$ by backpropagating $\mathcal{L}$
      Reset $\mathcal{L}$ and gradients to zero
    Compute validation PSNR using $\mathcal{D}_{\text{val}}$
until validation PSNR does not improve for e epochs

5.3.1 DeepJSCC-NOMA Architecture

We build upon the common DeepJSCC autoencoder backbone described in Section 2.1.2, using the conference version [147] with two downsampling/upsampling layers instead of four. Distributed compression architectures such as [65] employ encoders with different parameters for each device. However, this makes training more demanding due to the large number of parameters. Instead, we consider a Siamese neural network architecture [148], in which the two encoders at the transmitters share their parameters, i.e., $\boldsymbol{\Theta}_1=\boldsymbol{\Theta}_2=\boldsymbol{\Theta}$. This significantly reduces the complexity thanks to our novel device embedding-based input augmentation method. Figure 5.1 summarizes our architecture for DeepJSCC-NOMA. The encoder output is power-normalized as described in Section 2.1.2 to satisfy the constraint in Eq. (2.2).

We jointly train the whole network using the training data pairs $\mathcal{D}_{\text{train}}$ using the loss function $\mathcal{L}$:

$$\mathcal{L}\left(\boldsymbol{\Theta}, \boldsymbol{\Phi}\right) = \sum_{\substack{(\mathbf{x}_1, \mathbf{x}_2) \\ \in \mathcal{D}_{\text{train}}}} \text{MSE}\left(\mathbf{x}_1, \hat{\mathbf{x}}_1\right) + \text{MSE}\left(\mathbf{x}_2, \hat{\mathbf{x}}_2\right),$$

where $\text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \frac{1}{m} \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|_2^2$ and m is the total number of elements in $\mathbf{x}$, which is given by $m = C_{\text{in}}WH$. We note that the introduced method is unsupervised as it does not rely on any costly human labeling, and raw images and the channel model are enough to train our method.

Our method takes multiple instances as input and aims to decode two images, i.e., $\mathbf{x}_1$ and $\mathbf{x}_2$, simultaneously. Therefore, our task fits into the MTL [115] framework (see Section 2.6). One common method in MTL is to share parameters to model commonalities between tasks. In our problem, the task of image transmission for both encoders is exactly equivalent, so it is natural to share parameters of the encoders. However, the receiver also needs to distinguish between the images transmitted by the two encoders during the joint decoding phase. This is why we employ input augmentation, which is explained in the next section.

5.3.2 Input Augmentation via Novel Device Embeddings

Due to the superposition nature of the MAC, even if the receiver can recover both images with low distortion, it is challenging in the case of NOMA to distinguish which image belongs to which transmitter. We also want a scalable architecture; hence, we would like to employ the same encoder architecture for both transmitters. Note that an encoder that takes both images as input is not possible since we assume no communication between the devices during the transmission phase.

Without significantly increasing the number of parameters and without changing the DeepJSCC architecture that is known to perform well, we introduce an input augmentation method to be able to improve separability of signals in the aggregate representation domain. Hence, we employ device embeddings, which are unique trainable weights for each transmitting device. We initialize the two trainable device embeddings $\mathbf{r}_i \in 1 \times W \times H$ randomly from $\mathcal{N}\left(0, 1\right)$, $i = 1, 2$,

and concatenate with the input image as follows:

$$\tilde{\mathbf{x}}_i \triangleq \begin{bmatrix} \mathbf{x}_i \\ \mathbf{r}_i \end{bmatrix}.$$

Therefore, the inputs will have $C_{\text{in}} + 1$ dimensions, e.g., 4 dimensions instead of 3 for RGB images. This simple method allows the network to differentiate between different devices. These embeddings are optimized jointly during training and remain constant during test time. For $i = 1, 2$, $\tilde{\mathbf{x}}_i$ is given to encoder i as input, instead of $\mathbf{x}_i$, to obtain $\mathbf{z}_i = E_{\boldsymbol{\Theta},\sigma}(\tilde{\mathbf{x}}_i)$ before power normalization.

Instead of separate decoders at the receiver, we only employ image specific parameters at the last layer of the decoder. This significantly reduces the number of parameters and eases the training process while better separating the transmitted signals. The decoder $D_{\boldsymbol{\Phi},\sigma}$ outputs $2C_{\text{in}} \times W \times H$ dimensional image tensor, which is then reshaped into two separate images, i.e., to the shape $2 \times C_{\text{in}} \times W \times H$. Notice that although device embeddings introduce new trainable parameters, it only depends on the image size, and does not scale with the number of devices.

Notice that, by minimizing $\mathcal{L}$, the network learns to model intra-user distributions $p(\hat{\mathbf{x}}_1|\tilde{\mathbf{x}}_1)$, $p(\hat{\mathbf{x}}_2|\tilde{\mathbf{x}}_2)$ as well as inter-user distributions $p(\hat{\mathbf{x}}_2|\tilde{\mathbf{x}}_1)$, $p(\hat{\mathbf{x}}_1|\tilde{\mathbf{x}}_2)$. With inter-user distributions, the encoders ease separability of the input instances by training device embeddings $\mathbf{r}_1$ and $\mathbf{r}_2$ since the inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ are independent and the encoders share all the parameters.

5.3.3 Training Data Subsampling Strategy

We now need to define the training pairs in $\mathcal{D}_{\text{train}}$ to compute the loss, $\mathcal{L}$. Since we are training the network on pairs $(\mathbf{x}_1, \mathbf{x}_2)$, there are N^2 possible pair combinations for a given training dataset with N samples. However, it is generally infeasible to use all of these pairs since the number of training instances grows quadratically with N. Therefore, we subsample $T \ll N^2$ pairs from all possible combinations as in DML problems [120]. To ensure uniqueness of every pair, we subsample without replacement. We use the same pairs in $\mathcal{D}_{\text{train}}$ after shuffling to

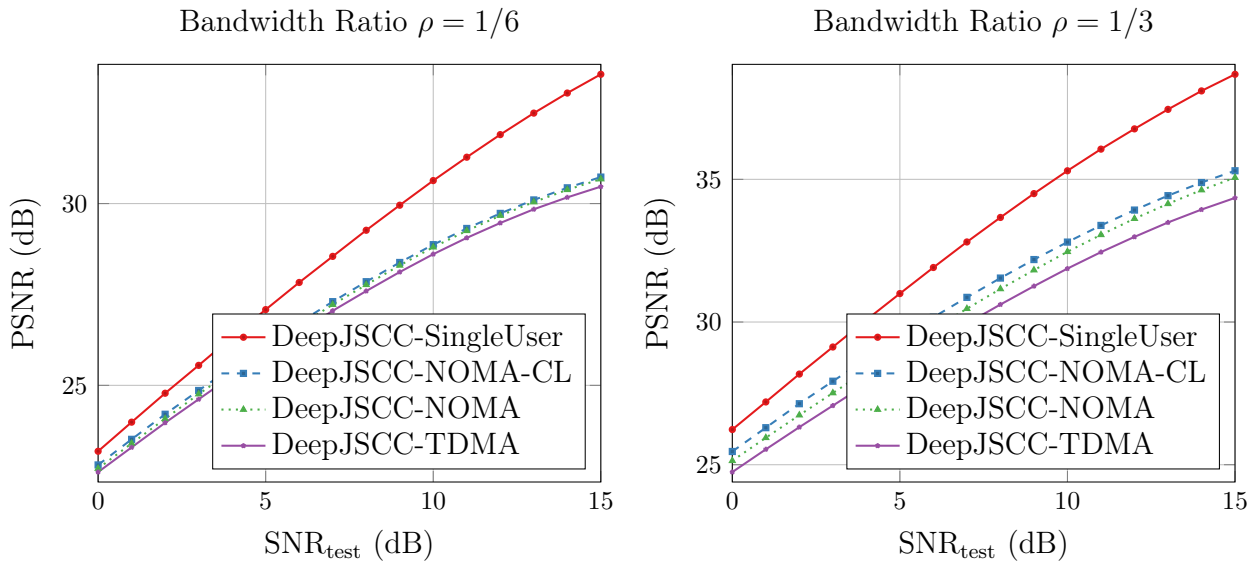


Figure 5.2: NOMA vs. TDMA-based DeepJSCC comparison for $\rho = 1/6$ and $\rho = 1/3$.

minimize $\mathcal{L}$. We also construct validation pairs $\mathcal{D}_{\text{val}}$ by splitting the initial validation data into two and use the same validation pairs after every epoch.

5.3.4 CL-Based Training Methodology

In our problem, the received signals are very noisy due to the interference from the other device. We address this problem by first training the network without superposition (which is an easier task) by only using the signal and the AWGN noise, i.e., by computing $\hat{\mathbf{x}}_1$ via $D_{\mathbf{\Phi},\sigma}(\mathbf{z}_1 + \mathbf{n})$ and $\hat{\mathbf{x}}_2$ via $D_{\mathbf{\Phi},\sigma}(\mathbf{z}_2 + \mathbf{n})$, both with the power constraint P_{avg} as in our standard training strategy. We then fine-tune the network with the standard training strategy on superposed signals as described above. This way, we enjoy benefits of the progressive training process via CL. Note that we do not change the loss function or any other component of the network in any phase of our CL training strategy.

5.4 Numerical Results

In this section, we present our experimental setup to demonstrate the performance gains of our method in different scenarios.

5.4.1 Dataset

We employ CIFAR-10 dataset for training and testing, which has training data with 50 000 instances and test data with 10 000. All the images have the shape of $3 \times 32 \times 32$ as RGB channels' dimension, width and height, respectively. We split the training data into 45 000 training instances and 5000 validation instances.

5.4.2 Baselines

We compare our method with classical DeepJSCC with time division, named DeepJSCC-TDMA. We name our method DeepJSCC-NOMA-CL. We compare it with the model without CL described in Section 5.3.4. We also compare it with the case of strong assumption of ideal SIC scenario to show the potential of our method, named DeepJSCC-SingleUser. This is also the initial model in CL-based strategy described in Section 5.3.4, which is trained and tested with non-interfering signals. We also note that previous studies already showed the superiority of DeepJSCC over classical separation-based transmission methods under power and bandwidth constraints [2], and in this chapter, we show superiority of our method over standard DeepJSCC employing time division.

5.4.3 Implementation Details

We conducted all experiments using the PyTorch framework [129]. We use the same hyperparameters and the same architecture for all the methods. We use learning rate 1×10^{-4} and batch size 64. We set the number of filters in the middle layers to 256 for both encoder and decoder. We use the power constraint $P_{\text{avg}} = 0.5$ for all the methods to make the TDMA-based DeepJSCC model comparable with previous works. We use Adam optimizer to minimize the loss [130]. We continue training until no improvement is achieved for consecutive $e = 10$ epochs. During training and validation, we run the model using different SNR values for each instance, uniformly chosen from $[0, 20]$ dB. We test and report the results for each SNR value using the same

model. For the proposed method, we use the training data with 200 000 unique pairs instead of $45\,000 \times 45\,000$ pairs using the method described in Section 5.3.3. We shuffle the training pairs or instances randomly before each epoch.

5.4.4 Comparison with TDMA-based DeepJSCC

Figure 5.2 demonstrates the performance gains of our method in different SNR conditions for $\rho = 1/6$ and $\rho = 1/3$ compression ratios. For both compression ratios, DeepJSCC-NOMA performs better than its TDMA counterpart for all evaluated SNRs. For $\rho = 1/3$, DeepJSCC-NOMA-CL achieves 0.91 dB (absolute) higher PSNR on average compared to DeepJSCC-TDMA. For $\rho = 1/6$, our method achieves 0.25 dB (absolute) higher PSNR on average compared to DeepJSCC-TDMA.

CL-based training further improves DeepJSCC-NOMA for all evaluated SNRs for both compression ratios. We observe that DeepJSCC-NOMA-CL achieves an improvement of 0.32 dB and 0.08 dB with respect to DeepJSCC-NOMA in terms of PSNR for $\rho = 1/3$ and $\rho = 1/6$, respectively. This shows the benefits of the employed CL method. DeepJSCC-SingleUser serves as an upper bound although it is a highly optimistic assumption, so we do not expect it to be tight.

5.4.5 Analysis of Fairness

Figure 5.3 illustrates the fairness of DeepJSCC-NOMA-CL for $\rho = 1/3$. There is no significant difference between the average PSNR achieved by the two devices, while both of them are outperforming TDMA-based DeepJSCC. A similar observation is also made for $\rho = 1/6$.

5.4.6 Analysis of the Number of DNN Parameters

Table 5.1 shows the comparison of our method with the standard point-to-point DeepJSCC with time-division. For both the compression rates $\rho = 1/6$ and $\rho = 1/3$, the amount of increase in the number of parameters is only $\approx 1\%$ thanks to the input augmentation and single decoder tricks.

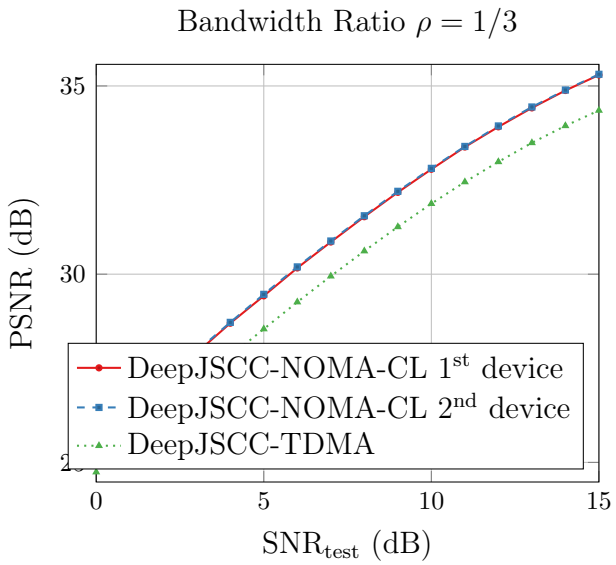


Figure 5.3: Fairness analysis between different users of our method for bandwidth ratio $\rho = 1/3$.

Table 5.1: Number of parameters for the compared methods

Method	$\rho = 1/3$	$\rho = 1/6$
DeepJSCC-TDMA	22.2M	22.1M
DeepJSCC-NOMA	22.4M	22.3M

Otherwise, we would need to use different encoder and decoders as explained in Section 5.3.2, which would cause growth in the number of parameters as the number of devices increases.

Notice that the differences between the proposed DeepJSCC-NOMA method and the classical DeepJSCC for TDMA are the followings: (1) number of parameters in the last layer of the encoder and the first layer of the decoder since the available bandwidth is twice in NOMA; (2) trainable embedding of our novel input augmentation described in Section 5.3.2; (3) the number of parameters in the last layer of decoder since we are decoding two different images using the same decoder. But, as the table shows, these introduce only a marginal increase in the number of parameters, and hence, the computational and memory complexity.

5.5 Conclusion

We have introduced a novel joint image compression and transmission scheme for the multi-user uplink scenario. The proposed approach exploits NOMA and the transmitters employ identical

DNNs. We have shown that, thanks to the novel input augmentation trick, the receiver is able to recover both images despite the analog transmission of underlying DeepJSCC approach, and is able to attribute decoded images to the correct transmitters. We have shown that the proposed DeepJSCC-NOMA scheme achieves superior performance compared to its time-division counterpart.

Chapter 6

Learning to Interfere in Non-Orthogonal Multiple-Access Joint Source-Channel Coding

6.1 Introduction

In wireless communication systems, efficiently transmitting data from multiple users to a common receiver is a fundamental challenge that has driven the development of various multiple access techniques. As outlined in Chapter 1, this thesis addresses the fourth research objective by developing scalable multi-user communication techniques. While Chapter 5 introduced a NOMA-based DeepJSCC framework for two users, scaling to many simultaneous users introduces significant additional challenges in interference management and training stability. Building upon that framework, this chapter introduces a novel progressive fine-tuning approach for scalable multi-user DeepJSCC that supports 64 users and can scale even more.

Traditional approaches rely on orthogonalization strategies such as TDMA and frequency-division multiple access (FDMA), where users are allocated dedicated time slots or frequency bands. TDMA is a communication technique, where users take turns transmitting data within specific

time slots. Although it enables the use of point-to-point schemes, TDMA is inherently limited in capacity compared to superposition-based schemes [8]. In contrast, NOMA allows multiple users to share the same time and frequency resources simultaneously [8, 9]. In the uplink setting considered in this work, multiple users transmit concurrently over a shared MAC, and the receiver must jointly decode the superposed signals. However, in NOMA, signals coming from different users inevitably interfere with each other, making the implementation significantly more complex and computationally heavy.

The quest for improved spectral efficiency has led to increased interest in JSCC, which combines source compression and channel coding into a unified framework. As discussed in Chapter 5, separation-based approaches are suboptimal in finite blocklength regimes even for independent sources. However, when it comes to practical finite blocklength regime, separation is suboptimal even in the case of independent inputs, and to the best of our knowledge, there are no practical joint coding algorithms that can surpass the performance of separation-based benchmarks with reasonable complexity for realistic source and channel distributions.

While DeepJSCC has shown remarkable success in single-user scenarios, several limitations emerge when considering multi-user networks. The 'analog' nature of DeepJSCC is critical in achieving robustness against channel variations. However, it can potentially become a limitation when it is considered in the context of a larger multi-user network. It can result in higher peak-to-average power ratio [149], prevent the encryption of the transmitted messages [63], and cause error accumulation over multi-hop networks [150, 151]. Most importantly for this work, it does not allow decoding and removal of interference in the case of multiple interfering terminals communicating with DeepJSCC.

Extending DeepJSCC to multi-user settings presents significant challenges. DeepJSCC can be trivially extended to multiple users via TDMA, as shown in Fig. 6.1. However, while TDMA mitigates the interference problem, it also results in suboptimal performance compared to superposition coding that exploits NOMA. The fundamental challenge in extending DeepJSCC to multiple users with NOMA stems from the fact that different users' signals interfere with each other, creating a complex optimization problem that traditional deep learning architectures

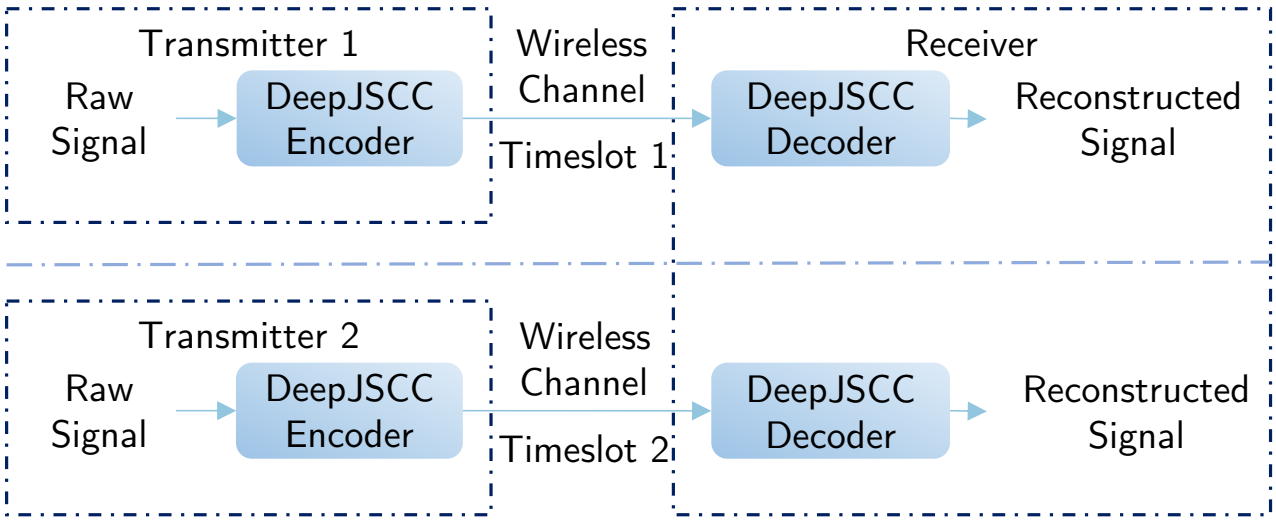


Figure 6.1: Extension of DeepJSCC to two users via TDMA.

struggle to handle efficiently. Additionally, the inherent focus of deep learning architectures on single-sample settings with i.i.d. data becomes problematic in multi-user scenarios, where the i.i.d. assumption breaks down, and Shannon's separation theorem no longer holds, complicating joint encoding and decoding. This complexity has resulted in prior work in the multi-user DeepJSCC domain being limited to only two users due to the scalability challenges posed by interference management and training complexity [19, 12, 11, 72, 73, 21].

To address these multi-user interference challenges, we propose a novel multi-user framework that scales to many users, as demonstrated with 64 users, by employing a multi-view autoencoder architecture with orthogonalized user-specific projections and progressive fine-tuning. Our primary objective is to develop a JSCC scheme capable of managing multiple users, while utilizing NOMA to achieve substantial performance improvements over traditional orthogonal approaches. The proposed coding scheme is inspired by the code division multiple access (CDMA) technique in digital communications, but we apply it to a continuous-amplitude modulation scheme in the context of JSCC, and learn the orthogonalization codebook employed by the users rather than using fixed chip sequences.

6.1.1 Contributions

Our main contributions are summarized as follows:

1. **Novel Scalable Multi-View Autoencoder Architecture**: Unlike existing multi-user JSCC methods that use separate networks per user or fixed power allocation schemes, we present a novel multi-view autoencoder architecture for multi-user semantic image transmission using NOMA. We introduce learnable user-specific projections, enabling shared encoder and decoder parameters across devices, with only 2.4% parameter overhead for 64 users compared to existing approaches that require linear scaling, crucial for scaling our solution to large wireless networks[1].

2. **Novel Progressive Fine-Tuning Strategy**: Addressing the training instability issues that limit existing multi-user JSCC methods to 2-3 users, we introduce a novel progressive fine-tuning strategy that doubles the number of users at each stage, building on any point-to-point DeepJSCC scheme. It maintains performance at the start of each stage thanks to orthogonalized user-specific projections.

3. **Comprehensive Performance Evaluation**: Extensive experiments show our method outperforms TDMA-based DeepJSCC, NOMA alternatives, and separation-based methods with BPG, neural codecs, and LDPC across all SNR conditions for both independent and correlated source samples, ensuring fair performance among users. Our method consistently outperforms existing multi-user approaches across all SNR conditions and successfully scales to 64 users, while existing methods are limited to 2-3 users. With 64 users, performance improves with only 2.4% additional trainable parameters compared to a single-user architecture.

4. **Performance Analysis**: We perform comprehensive ablation studies clearly showing gains from progressive-fine-tuning, user-specific projections, and orthogonal initialization. These studies demonstrate that each component addresses specific limitations of existing multi-user JSCC approaches: progressive training solves convergence issues, learnable projections outperform fixed schemes, and orthogonal initialization ensures training stability.

[1]We will publish the source code and model checkpoints on GitHub at `https://github.com/ipc-lab/deepjscc-pnoma` under the CC-BY 4.0 license.

6.2 System Model and Problem Definition

We extend the point-to-point DeepJSCC system model in Section 2.1.1 to an n-user uplink MAC setting with a single receiver. Let $\mathbf{x}_i \in \mathbb{I}^{C_{\text{in}} \times W \times H}$ denote the image of user i. The channel output is

$$\mathbf{y} = \sum_{i=1}^{n} h_i \mathbf{z}_i + \mathbf{n} \in \mathbb{C}^k, \tag{6.1}$$

where $\mathbf{z}_i \in \mathbb{C}^k$ is the transmitted signal vector of user i, $h_i \in \mathbb{C}$ is the channel gain, and $\mathbf{n} \sim \mathbb{C}\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_k)$ is i.i.d. complex Gaussian noise with variance σ^2. We set $h_i = 1, \forall i$, for the AWGN channel and $h_i \sim \mathbb{C}\mathcal{N}(0, 1)$ for the Rayleigh fading channel, $\forall i \in [n]$. Each user is subject to the average power constraint in Eq. (2.2). The bandwidth ratio ρ and SNR are defined in Section 2.1.1.

Throughout this work, we assume that σ and $\mathbf{h} = \begin{bmatrix} h_1 & \dots & h_n \end{bmatrix}^T$ are known to both the users and the receiver. The i^{th} user employs a nonlinear encoding function $E_{\boldsymbol{\Theta}_i}$, parameterised by $\boldsymbol{\Theta}_i$, to map its image into a complex-valued latent vector $\mathbf{z}_i = E_{\boldsymbol{\Theta}_i}(\mathbf{x}_i, \mathbf{h}, \sigma) \in \mathbb{C}^k$. A single nonlinear decoding function $D_{\boldsymbol{\Phi}}$, parameterised by $\boldsymbol{\Phi}$, jointly decodes all user images from the superposed channel output:

$$\begin{bmatrix} \hat{\mathbf{x}}_1 & \dots & \hat{\mathbf{x}}_n \end{bmatrix}^T = D_{\boldsymbol{\Phi}}(\mathbf{y}, \mathbf{h}, \sigma). \tag{6.2}$$

Joint decoding enables the decoder to exploit the joint distribution of all user signals and can, in principle, achieve the full MAC capacity region [70]. Together with superposition coding with SIC and time-sharing, and rate-splitting [79], joint decoding is one of the capacity-achieving strategies for the MAC. Our neural network implementation provides a practical approximation to this theoretically optimal scheme.

For fair comparison among methods with different numbers of users, we define the per-user bandwidth ratio as $\bar{\rho} \triangleq \rho/n$ and the per-user average power constraint as $\bar{P}_{\text{avg}} \triangleq P_{\text{avg}}/n$. Throughout this work, we use $\bar{P}_{\text{avg}} = 1$ for all methods to make the TDMA-based DeepJSCC baseline comparable with prior work [2, 32].

The goal is to maximise the average PSNR (defined in Section 2.7.1) on an unseen target dataset.

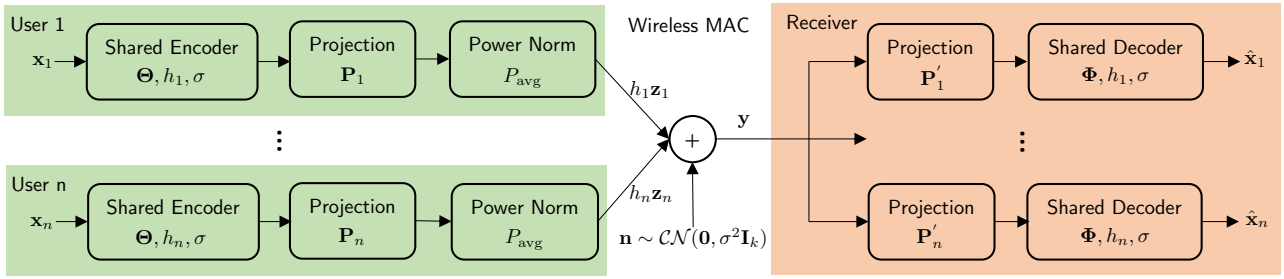


Figure 6.2: Overall architecture of our DeepJSCC-PNOMA method.

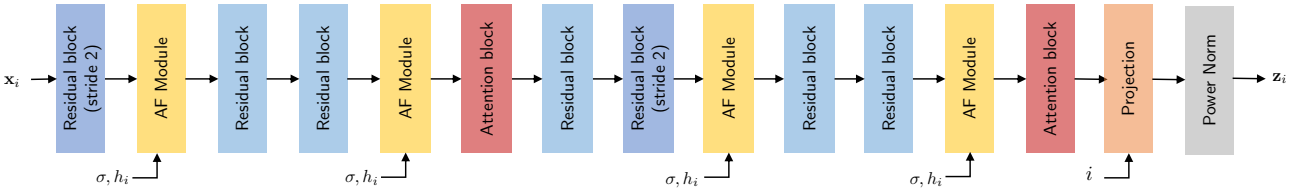


Figure 6.3: Encoder architecture, user-specific projection layer and power normalization layer of the introduced method.

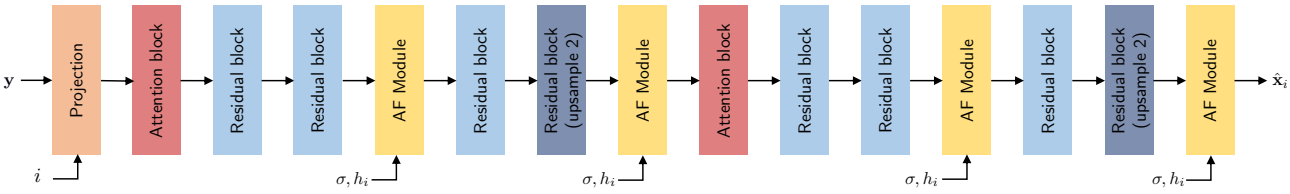


Figure 6.4: User-specific projection layer and decoder architecture of the introduced method.

We additionally evaluate using SSIM, MS-SSIM, and LPIPS, all defined in Section 2.7.

6.3 Methodology

Designing a multi-user JSCC system involves several interrelated architectural choices. First, joint decoding, where a single decoder processes the superposed channel output, can exploit inter-user correlations and, in principle, achieve the full MAC capacity region, but at a higher computational cost than sequential approaches such as SIC. Second, full parameter sharing across users is efficient but makes it difficult for the decoder to distinguish individual transmissions; conversely, separate per-user networks scale poorly with the number of users. Third, interference must be managed in a structured way so that the network can learn to exploit, rather than merely suppress, the superposition of signals. Finally, the system should support a variable number of users without retraining from scratch.

These considerations motivate our *multi-view autoencoder* design, DeepJSCC-PNOMA, which combines a shared encoder–decoder backbone with lightweight, learnable user-specific projection matrices for structured user differentiation, and a progressive fine-tuning procedure that doubles the user count at each stage while preserving training stability. The remainder of this section details each component.

6.3.1 Point-to-Point DeepJSCC Architecture

We build upon the common DeepJSCC autoencoder backbone described in Section 2.1.2, based on the encoder and decoder designs from [32]. In point-to-point DeepJSCC, the transmitter encodes the input as $\tilde{\mathbf{z}} = E_{\boldsymbol{\Theta}}(\mathbf{x}, h, \sigma)$, then flattens and power-normalizes it as described in Section 2.1.2 to satisfy the constraint in Eq. (2.2). The receiver decodes the noisy channel output $\mathbf{y}$ using $\hat{\mathbf{x}} = D_{\boldsymbol{\Phi}}(\mathbf{y}, h, \sigma)$.

Remark 1 (Extensibility)**.** *The encoder and decoder architectures are interchangeable with other autoencoders. Our components directly utilize the encoder and decoder without modifications or constraints, allowing for flexible adoption of new architectures and performance improvements.*

6.3.2 Parameter Sharing Between Users via User-Specific Projections

We extend the point-to-point architecture to a multi-user setting over a MAC by introducing a novel superposition coding technique that improves scalability and performance of multi-user neural networks. Figure 6.2 summarizes our method, named DeepJSCC-PNOMA. We assume that no communication occurs between users during the transmission phase. Figures 6.3 and 6.4 illustrate the encoder and decoder architectures for our multi-user DeepJSCC, respectively. This architecture is flexible and can be substituted with any other encoder and decoder architecture, as mentioned in Remark 1.

Multi-user JSCC requires managing interference while maintaining scalability. Traditional approaches use either separate networks per user (leading to 6300% parameter overhead for 64 users) or simple parameter sharing (poor user differentiation). Our multi-view autoencoder

treats each user as a distinct "view," enabling structured interference management through learnable projections with shared backbones, requiring only 2.4% parameter overhead for 64 users at $\bar{\rho} = 1/12$ and 9.4% at $\bar{\rho} = 1/6$.

Many distributed compression and multi-user DeepJSCC systems, such as those described in [65, 21, 12, 11, 72, 73, 39], use distinct encoders and decoders for each user. A practical system, however, should employ parameter sharing to enhance efficiency and scalability. However, challenges arise when all users employ the same mapping function, as users create direct interference and the model struggles to differentiate between users' data.

Our multi-view framework treats each user as a distinct "view": shared encoder and decoder parameters capture common patterns, learnable user-specific projections $\mathbf{P}_i$ create structured separation, joint decoding processes combined signals, and orthogonal initialization ensures training stability.

We introduce a novel superposition coding method with user-specific projections, which allows the network to scale to multiple users without significantly increasing the number of parameters and without changing the DeepJSCC architecture that is known to perform well. The encoders at the users share all their parameters, excluding the user-specific projection layer, i.e., $\mathbf{\Theta}_1 = \cdots = \mathbf{\Theta}_n = \mathbf{\Theta}$. At every user $i \in [n]$, we have a complex-valued projection matrix $\mathbf{P} \in \mathbb{C}^{m \times nm}$, where m is the number of filters of the last convolutional layer of the encoder network. Similarly, at the receiver, we have a complex-valued projection matrix $\mathbf{P}' \in \mathbb{C}^{nm \times m}$ for every image to be decoded, i.e., $i \in [n]$. We initialize $\mathbf{P}_i$ and $\mathbf{P}'_i$, $\forall i \in [n]$, in a way that all of their rows are orthogonal, via QR factorization as detailed in Algorithm 8. We jointly optimize projection matrices along with the other parameters of the network throughout the training.

Remark 2 (CDMA Analogy)**.** *These matrices can be considered as the spreading codes in a CDMA system. The list of possible projection matrices can be agreed upon a priori, and these matrices can be shared across all the terminals. Then, at each instance, the active users can be assigned one of these matrices by the receiver, e.g., base station.*

Unlike traditional CDMA spreading codes (<1KB per user) [9], our approach requires projection matrices $\mathbf{P}_i \in \mathbb{C}^{m \times nm}$ *and* $\mathbf{P}'_i \in \mathbb{C}^{nm \times m}$ *(32MB per user for* $m = 256$*,* $n = 64$*). This overhead*

enables end-to-end optimization and superior interference management, with matrices distributed once during deployment.

Data processing begins with the encoder DNN, which converts an image into a real-valued tensor of shape $\mathbb{R}^{W' \times H' \times 2m}$, where W' and H' are the downsampled dimensions. These tensors are then mapped to complex numbers by pairing consecutive real values, resulting in $\mathbb{C}^{W' \times H' \times m}$. Each user subsequently applies a user-specific projection by multiplying with $\mathbf{P}_i$, resulting in $\tilde{\mathbf{z}}_i \mathbf{P}_i$, where $\tilde{\mathbf{z}}_i$ is the encoder output. The output is flattened and undergoes power normalization as described in Section 2.1.2 to satisfy the constraint in Eq. (2.2). Finally, each user $i \in [n]$ sends its signal over the channel.

The receiver obtains superposed signals along with multiplicative and additive noise. The receiver first reshapes the matrix to image form of shape $\mathbb{C}^{W^{'} \times H^{'} \times m}$, and then multiplies the reshaped tensor with $\mathbf{P}_i^{'}$, i.e., $\mathbf{y}\mathbf{P}_i^{'}$, which is then given as an input to the decoder DNN to obtain $D_{\Phi}(\mathbf{y}\mathbf{P}_i^{'})$. This projection and decoding step is repeated for every user $i \in [n]$. This enables parameter sharing while distinguishing user inputs/outputs, improving training and scalability. The multi-view design provides computational efficiency, progressive training compatibility, variable user support, and joint optimization benefits.

Remark 3 (Scalability)**.** *The overall number of network parameters is bounded by*

$$\mathcal{O}\Big(|\mathbf{\Theta}| + |\mathbf{\Phi}| + n^2 \bar{\rho}^2\Big),$$

where $|\mathbf{\Theta}|$ and $|\mathbf{\Phi}|$ denote the sizes of the shared encoder and decoder, respectively, and the term $n^2\bar{\rho}^2$ accounts for the user-specific projection matrices. Since the dominant contribution comes from the shared parameters $|\mathbf{\Theta}| + |\mathbf{\Phi}|$, the architecture remains scalable even as the number of users n grows large.

Remark 4 (Shape Adaptivity)**.** *The network can process input images with arbitrary width W and height H since only the filter dimension is affected by the projection.*

Remark 5 (Parallelization)**.** *The receiver can decode images from different users in parallel because the decoders are identical, and the projections are independent, requiring no communication.*

The receiver only needs to assign a particular index to each user so that they know which projection matrix to use. This can be done during the user admission process.

6.3.3 Training Procedure and Construction of Training Samples

We jointly train the whole network with parameters $\boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathbf{P}_i, \mathbf{P}_i', \forall i \in [n]$, using the training data samples $\mathcal{D}_{\text{train}}$ via the loss function $\mathcal{L} = \sum_{\substack{(\mathbf{x}_1,\ldots,\mathbf{x}_n) \\ \in \mathcal{D}_{\text{train}}}} \sum_{i=1}^{n} \text{MSE}\left(\mathbf{x}_i, \hat{\mathbf{x}}_i\right)$, where $\text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \frac{1}{m} \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|_2^2$ and m is the total number of elements in $\mathbf{x}$, which is given by $m = C_{\text{in}}WH$. Notice that minimizing $\mathcal{L}$ also maximizes PSNR over the training set. We note that the introduced method is unsupervised as it does not rely on any costly human labeling, and raw images and the channel model are enough to train our method. Algorithm 5 shows the training and fine-tuning procedures of DeepJSCC-PNOMA.

During training, our neural network takes multiple image instances $\mathbf{x}_1, \ldots, \mathbf{x}_n$ as input and decodes these images as $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n$. Therefore, our task aligns well with the MTL framework [115] (see Section 2.6), as reconstructing the image of each user can be viewed as a distinct yet closely related task. One common method in MTL is to share parameters to model commonalities between tasks. In our problem, since the task of image transmission for all encoders is exactly equivalent, we naturally share parameters among the encoders. However, the receiver also needs to distinguish between the images transmitted by different encoders and associate each decoded image to its transmitter during the decoding phase. This is why we employ a user-specific projection layer, which is described in Section 6.3.2. The projection matrices $\mathbf{P}_i$ and $\mathbf{P}_i'$, $\forall i \in [n]$, are optimized jointly with the rest of the network during training and remain constant during test time. This simple method allows the use of a standard single-user DeepJSCC architecture, spreading its output over the available bandwidth while obeying the average power constraint and enabling the decoder to differentiate between the transmitters of different images.

We now define the training tuples in $\mathcal{D}_{\text{train}}$ to compute the loss function $\mathcal{L}$. Since we are training the network on $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ tuples, there are N^n possible combinations for a given training dataset with N samples, i.e., $|\mathcal{D}_{\text{train}}| = N$. However, it is generally infeasible to use all of these combinations since the number of training instances grows exponentially with N. Moreover, our

Algorithm 5: Training and Fine-tuning of DeepJSCC-PNOMA

Construct $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ for n users ▷ *See Section 6.3.3*
Initialize $\mathbf{\Theta}$ and $\mathbf{\Phi}$ randomly for $n = 1$ or from previous values for $n > 1$ ▷ *See Section 6.3.4*
Initialize $\mathbf{P}_i$ and $\mathbf{P}_i^{'}$, $\forall i \in [n]$, as I for $n = 1$ or orthogonally for $n > 1$ ▷ *See Section 6.3.2*
repeat ▷ *Iterate through epochs*
  Shuffle $\mathcal{D}_{\text{train}}$ and set $\mathcal{L} = t = 0$
  for all $(\mathbf{x}_1, \ldots) \in \mathcal{D}_{\text{train}}$ **do**
    for $i \in [n]$ **do** ▷ *Device with index i*
      $\tilde{\mathbf{z}}_i = E_{\mathbf{\Theta}}(\mathbf{x}_i, h_i, \sigma)$ ▷ *Encoding*
      $\mathbf{z}_i = \text{Flatten}(\tilde{\mathbf{z}}_i \mathbf{P}_i)$ ▷ *User-specific projection*
      $\mathbf{z}_i = \sqrt{kP_{\text{avg}}/(\mathbf{z}_i \mathbf{z}_i^H)}\mathbf{z}_i$ ▷ *Power normalization*
    ▷ *Transmission of latents over a MAC* ◁
    Calculate σ via Eq. (2.6) for SNR $\sim$ Uniform $[0, 20]$
    Sample $\mathbf{n} \sim \mathbb{C}\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_k)$
    if channel is Gaussian **then**
      $h_i = 1$, $\forall i \in [n]$
    else if channel is Rayleigh **then**
      Sample $h_i \sim \mathbb{C}\mathcal{N}(0, 1)$, $\forall i \in [n]$
    $\mathbf{y} = \sum_{i=1}^{n} h_i \mathbf{z}_i + \mathbf{n}$ ▷ *Received signal*
    for $i \in [n]$ **do**
      $\mathbf{y}_i = \text{Reshape}(\mathbf{P}_i^{'} \mathbf{y})$ ▷ *User-specific projection*
      $\hat{\mathbf{x}}_i = D_{\Phi}(\mathbf{y}_i, h_i, \sigma)$ ▷ *Decoding*
    $\mathcal{L} = \mathcal{L} + \sum_{i=1}^{n} \text{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i)$
    t = t + 1
    if t mod *batch size* $= 0$ **then**
      ▷ *Standard mini-batch training* ◁
      Update $\mathbf{\Theta}, \mathbf{\Phi}, \mathbf{P}_i, \mathbf{P}_i^{'}$, $\forall i \in [n]$ by backpropagation
      Reset $\mathcal{L}$ and gradients to zero
  Compute validation PSNR using $\mathcal{D}_{\text{val}}$
until validation PSNR improves less than Δ for e epochs

experimental results suggest that distributing random samples from a shuffled training batch to different users may cause over-regularization and lead to underfitting. Consistently using the same sample combinations from a finite list of tuples across different epochs improved the outcomes.

Therefore, we subsample $T \ll N^n$ tuples from all possible combinations [120]. To generate T tuples, we sample nT integers from $[N]$ and group every n consecutive samples. We use the same tuples in $\mathcal{D}_{\text{train}}$ after shuffling to minimize $\mathcal{L}$. We also construct validation tuples $\mathcal{D}_{\text{val}}$ by splitting the initial validation data among n users and use the same validation tuples after every epoch. Algorithms 6 and 7 outlines the method used to construct the training and evaluation

samples, respectively.

Algorithm 6: Construction of training samples

▷ *Inputs: number of tuples T, size of training data $N = |\mathcal{D}_{\text{train}}|$, number of users n* ◁

▷ *Generate a vector of random permutation of nT indices* ◁

$\mathbf{t} = \text{RandomPermutation}\left([1, 2, \ldots, nT]^{\text{T}}\right)$

▷ *Match them to the training data size* ◁

$\mathbf{t} = \mathbf{t} \bmod n$

▷ *Assign indices to users by reshaping* ◁

$\mathbf{T} = \text{Reshape}\left(\mathbf{t}, (T, n)\right)$

▷ $\mathbf{T}$ *can now be used as indices of the samples for training, where first dimension is shuffled at every epoch* ◁

Algorithm 7: Construction of evaluation (validation and test) samples

▷ *Inputs: number of users n, size of evaluation data M (either the cardinality of validation data $|\mathcal{D}_{\text{val}}|$ or the cardinality of test data $|\mathcal{D}_{\text{test}}|$)* ◁

▷ *Generate a vector of random permutation of M indices* ◁

$\mathbf{t} = \text{RandomPermutation}\left([1, 2, \ldots, M]^{\text{T}}\right)$

▷ *Assign indices to users by reshaping* ◁

$\mathbf{T} = \text{Reshape}\left(\mathbf{t}, \left(\frac{M}{n}, n\right)\right)$

▷ $\mathbf{T}$ *can now be used as indices of the samples for evaluation* ◁

6.3.4 Progressive Fine-Tuning Method for Doubling the Number of Users

It is known that high noise in the data, gradients or weights adversely affect the training of DNNs. In a MAC, we not only deal with additive and multiplicative noise effects but also interference from other users since signals are superposed.

When $\mathbf{P}_i$ and $\mathbf{P}_i'$, $\forall i \in [n]$, are initialized randomly, signals of different devices interfere with each other, and the amount of interference increases with the number of users. We address this problem via a novel progressive fine-tuning method by gradually doubling the number of users while sustaining the performance of fine-tuned network in the beginning of training via orthogonal initialization of projection matrices.

Figure 6.5 illustrates our progressive fine-tuning methodology, which enables stable scaling from

Algorithm 8: Progressive fine-tuning for doubling the number of users in DeepJSCC-PNOMA

Initialize $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ randomly for a given $\bar{\rho}$
$k = 3WH\bar{\rho} = 3WH\rho$ ▷ *k: available bandwidth for TDMA case ($n = 1$) given $\bar{\rho}$.*
$m = \frac{k}{WH/4^c} = \frac{3WH\rho}{WH/4^c} = 3\rho 4^c$ ▷ *m: # of transmitted filters*
$\mathbf{P}_1 = \mathbf{P}_1^{'} = \mathbf{I}_m$
▷ *Initial Training* ◁
Train $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ using Algorithm 5
$\mathbf{P}_{1,\text{old}} = \mathbf{P}_1$
$\mathbf{P}_{1,\text{old}}^{'} = \mathbf{P}_1^{'}$
for all $n \in (2^1, 2^2, 2^3, \ldots)$ **do**
  ▷ *Generate orthogonal matrix* $\mathbf{Q}$ ◁
  Sample $\mathbf{M} \sim \mathcal{N}\left(\mathbf{0}_{nm}, \frac{1}{nm}\mathbf{I}_{nm}\right)$
  $\mathbf{Q}, \mathbf{R} = \text{QRFactorization}\left(\mathbf{M}\right)$
  ▷ *Make* $\mathbf{Q}$ *uniform according to the procedure in [152]* ◁
  $\mathbf{d} = \text{Sign}\left(\text{Diagonal}\left(\mathbf{R}\right)\right)$
  $\begin{bmatrix}\mathbf{K}_1 \\ \mathbf{K}_2\end{bmatrix} = \mathbf{d} \cdot \mathbf{Q}$ ▷ $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{C}^{\frac{n}{2}m \times nm}$
  for all $i \in (1, 2, \ldots, {}^{n}/_{2})$ **do**
    ▷ *Copy and orthogonalize projection matrices of first half of devices* ◁
    $\mathbf{P}_i = \mathbf{P}_{i,old}\mathbf{K}_1$
    $\mathbf{P}_i^{'} = \mathbf{K}_1^H \mathbf{P}_{i,old}^{'}$
    ▷ *Copy and orthogonalize projection matrices of second half of devices* ◁
    $\mathbf{P}_{i+{}^{n}/_{2}} = \mathbf{P}_{i,old}\mathbf{K}_2$
    $\mathbf{P}_{i+{}^{n}/_{2}}^{'} = \mathbf{K}_2^H \mathbf{P}_{i,old}^{'}$
  ▷ *Fine-tuning step* ◁
  Fine-tune $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, $\mathbf{P}$ and $\mathbf{P}_i^{'}$, $\forall i \in [n]$ using Algorithm 5
  ▷ *Copy projection matrices for the next step* ◁
  for all $i \in [n]$ **do**
    $\mathbf{P}_{i,\text{old}} = \mathbf{P}_i$
    $\mathbf{P}_{i,\text{old}}^{'} = \mathbf{P}_i^{'}$
  Export DeepJSCC-PNOMA with n users: $E_{\boldsymbol{\Theta}}$, $D_{\boldsymbol{\Phi}}$, $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, $\mathbf{P}_i$ and $\mathbf{P}_i^{'}$, $\forall i \in [n]$

single-user to multi-user configurations. The detailed algorithmic procedure is presented in Algorithm 8.

We first start with training a standard point-to-point DeepJSCC (or DeepJSCC-TDMA), i.e., DeepJSCC-PNOMA when $n = 1$ and $\mathbf{P}_1 = \mathbf{I}_m$, where m is the number of filters in the transmitted signal that is processed by CNNs. Given a trained or fine-tuned network for ${}^{n}/_{2}$ users, i.e. $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$ and $\mathbf{P}_i$, $\mathbf{P}_i^{'}$, $\forall i \in [{}^{n}/_{2}]$, we now describe how we extend it to n users as follows. We first create a new encoder E and decoder D for each user using the previous parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ with parameter sharing, respectively. We then copy the previous projections to $\mathbf{P}_{i,\text{old}} = \mathbf{P}_i$ and $\mathbf{P}_{i,\text{old}}^{'} = \mathbf{P}_i^{'}$,

$\forall i \in [n/2]$. Then, we duplicate all these projection matrices so that $\mathbf{P}_i = \mathbf{P}_{n/2+i} = \mathbf{P}_{i,\text{old}}$ and $\mathbf{P}'_i = \mathbf{P}'_{n/2+i} = \mathbf{P}'_{i,\text{old}}$. We then generate a uniformly distributed orthonormal matrix by sampling an $nm \times nm$ Gaussian matrix $\mathbf{M} \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{nm}\mathbf{I}_{nm}\right)$ and computing its QR factorization, $\mathbf{M} = \mathbf{QR}$. To remove the sign (or phase) ambiguity in the QR decomposition, we post-multiply $\mathbf{Q}$ by a diagonal matrix with entries [152]

$$\mathbf{d} = \text{Sign}\big(\text{Diagonal}(\mathbf{R})\big),$$

and partition the adjusted matrix as

$$\begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix} = \mathbf{d} \cdot \mathbf{Q},$$

where $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{C}^{\frac{n}{2}m \times nm}$.

We multiply $\mathbf{P}_i$, $\forall i \in [n/2]$, with $\mathbf{K}_1$ and multiply $\mathbf{P}_i$, $\forall i \in [n/2 + 1, n]$, with $\mathbf{K}_2$. We also multiply $\mathbf{P}'_i$, $\forall i \in [n/2]$, with $\mathbf{K}_1^H$ and multiply $\mathbf{P}'_i$, $\forall i \in [n/2 + 1, n]$, with $\mathbf{K}_2^H$. Notice that it is enough to optimize $\mathbf{P}_i$ and $\mathbf{P}'_i$ instead of their factors due to the linearity property of matrix multiplication, allowing efficiency in optimization.

At this stage, it is crucial to note that the power of the encoded signal remains unchanged after applying the projection matrices. In other words, for each $i \in \{1, \ldots, n\}$, the encoded signal $\tilde{\mathbf{z}}_i$ satisfies

$$\|\tilde{\mathbf{z}}_i\|_2^2 = \|\tilde{\mathbf{z}}_i \mathbf{P}_i\|_2^2.$$

This property follows from the fact that each projection matrix $\mathbf{P}_i$ is unitary (i.e., $\mathbf{P}_i^H \mathbf{P}_i = I$); specifically, since $\|\tilde{\mathbf{z}}_i\|_2^2 = \tilde{\mathbf{z}}_i^H \tilde{\mathbf{z}}_i$ and

$$\|\tilde{\mathbf{z}}_i \mathbf{P}_i\|_2^2 = (\tilde{\mathbf{z}}_i \mathbf{P}_i)^H (\tilde{\mathbf{z}}_i \mathbf{P}_i) = \tilde{\mathbf{z}}_i^H (\mathbf{P}_i^H \mathbf{P}_i) \tilde{\mathbf{z}}_i = \tilde{\mathbf{z}}_i^H \tilde{\mathbf{z}}_i,$$

the power is preserved.

Moreover, due to the orthogonality of the new projection matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ (i.e., $\mathbf{K}_1 \mathbf{K}_2^H = 0$),

the normalized signals corresponding to different user groups are guaranteed to be orthogonal at this stage before any optimization. In particular, for a user i (using $\mathbf{K}_1$) and a user j (using $\mathbf{K}_2$), the inner product of their normalized signals is

$$\mathbf{z}_i \cdot \mathbf{z}_j^H = \frac{\tilde{\mathbf{z}}_i \mathbf{P}_i \mathbf{K}_1}{\sqrt{\frac{1}{k}\|\tilde{\mathbf{z}}_i \mathbf{P}_i \mathbf{K}_1\|^2}} \cdot \left(\frac{\tilde{\mathbf{z}}_j \mathbf{P}_j \mathbf{K}_2}{\sqrt{\frac{1}{k}\|\tilde{\mathbf{z}}_j \mathbf{P}_j \mathbf{K}_2\|^2}} \right)^H = 0,$$

which follows immediately from $\mathbf{K}_1 \mathbf{K}_2^H = 0$.

Due to the design of the user-specific projection matrices in DeepJSCC-PNOMA, which preserve the power of each encoded signal and enforce mutual orthogonality among signals, the system with n users initially behaves as if it were time-sharing among $\frac{n}{2}$ users under fixed per-user average power $\bar{P}_{\text{avg}}$ and per-user bandwidth ratio $\bar{\rho}$. In this configuration, each user's signal retains its original power and remains free from interference, enabling independent decoding. Consequently, the network can begin fine-tuning without any performance degradation from inter-user interference.

By gradually increasing the system complexity while maintaining stability and efficiency, this progressive training approach serves as an effective curriculum for multi-user learning. The encoder–decoder pair is guaranteed to learn a robust feature representation without any interference from other users when they begin with a single-user DeepJSCC baseline. In order to guarantee that each user initially occupies a unique and interference-free subspace in the transmission domain, previously learned projections are replicated and orthogonalized using QR-based transformations when the number of users doubles. The power of every encoded signal stays constant throughout all stages due to the unitary nature of the projection matrices. By fine-tuning at every stage of expansion, the network can seamlessly adjust to the new user configuration and gradually learn to handle lingering interference and user correlation. As a result, the model converges faster, avoids instability associated with joint multi-user initialization, and achieves improved generalization through structured and incremental adaptation.

The next step is to fine-tune the model using the same training procedure described in Section 6.3.3.

6.4 Numerical Results

6.4.1 Datasets

Dataset Overview: Our experimental evaluation employs four diverse datasets to comprehensively assess the performance and generalizability of DeepJSCC-PNOMA across different image characteristics, resolutions, and application domains. This multi-dataset approach ensures robust validation of our method's effectiveness across varied visual content types and complexity levels.

We use the CIFAR10 dataset [156] for training and testing. CIFAR10 consists of 50 000 training images and 10 000 test images, each of dimensions $3 \times 32 \times 32$. We further split the training set into 45 000 training instances and 5000 validation instances. In addition, we employ the TinyImagenet dataset, which contains 100 000 training instances, 10 000 validation instances, and 10 000 test instances, all with the shape $3 \times 64 \times 64$. For evaluating correlated sources, we use the Cityscapes dataset [134], which comprises 5000 stereo image pairs. Specifically, 2975 pairs are allocated for training, while 500 and 1525 pairs are used for validation and testing, respectively. Following [21], each image in the Cityscapes dataset is downsampled to 128×256. Finally, to assess generalization performance and enable qualitative comparisons, we employ the Kodak dataset, which consists of 24 images of dimensions $3 \times 512 \times 768$. Given its limited size, the Kodak dataset is reserved exclusively for testing.

Dataset Licenses: All these datasets are publicly accessible under permissible licenses. We use CIFAR10 dataset by following the MIT License; TinyImagenet by following the MIT license since it is a subset of Imagenet; Cityscapes by following specific license agreement on its website (`https://www.cityscapes-dataset.com/license/`) that is permissive and publicly available for academic purposes; and Kodak by following GNU GPLv3 license.

Dataset Sources: We use CIFAR10 downloaded by Torchvision; TinyImagenet dataset downloaded from `https://github.com/rmccorm4/Tiny-Imagenet-200`; Cityscapes dataset downloaded from `https://www.cityscapes-dataset.com/`; Kodak dataset downloaded from

Table 6.1: Employed hyperparameters for fair comparison between scenarios with different number of users

$\bar{\rho}$	$\bar{P}_{\text{avg}}$	σ^2	Method	n	ρ	P_{avg}
1/6	1	1	DeepJSCC-TDMA	1	1/6	1
			DeepJSCC-PNOMA	1	1/6	1
				2	2/6	1/2
				4	4/6	1/4
				8	8/6	1/8
				16	16/6	1/16
			Perfect SIC (2 users)	1	2/6	1/2
			Perfect SIC (16 users)	1	16/6	1/16
1/12	1	1	DeepJSCC-TDMA	1	1/12	1
			DeepJSCC-PNOMA	1	1/12	1
				2	2/12	1/2
				4	4/12	1/4
				8	8/12	1/8
				16	16/12	1/16
			Perfect SIC (2 users)	1	2/12	1/2
			Perfect SIC (16 users)	1	16/12	1/16

`https://huggingface.co/datasets/Freed-Wu/kodak`.

Data Preprocessing: All datasets undergo consistent preprocessing: normalization to [0,1] range, random horizontal flips during training (except Kodak), and appropriate resizing when necessary. No additional augmentation techniques are applied to maintain fair comparison with baseline methods and preserve the original image characteristics essential for transmission quality evaluation.

Train/Validation/Test Splits: For CIFAR-10 and TinyImageNet, we use standard splits provided by the datasets. For Cityscapes, we utilize the official train/validation split for correlated experiments. For Kodak, given its small size, we use all 24 images for evaluation purposes in fairness analysis, consistent with established image compression evaluation protocols.

6.4.2 Implementation Details

We conducted all experiments using the PyTorch framework [129]. We use the same hyperparameters and the same architecture for all the methods. Following previous works for point-to-point DeepJSCC [2, 32], we use learning rate 1×10^{-4}, set the number of filters in the middle CNN layers to 256 and batch size to 32. We use Adam optimizer to minimize the loss [130]. We continue training until no more than $\Delta = 1e-3$ improvement is achieved for consecutive $e = 10$ epochs. During training and validation, we run the model using different SNR values for each instance, uniformly chosen from $[0, 20]$ dB. We test and report the results for each SNR value using the same model. For the proposed method, we use the training data with $3N$ randomly sampled pairs, where elements are chosen among the same training set with N instances, instead of N^n pairs for both CIFAR10 and TinyImagenet datasets, using the method described in Section 6.3.3. We shuffle the training pairs or instances randomly before each epoch. We conduct all our deep learning experiments by training models with the distributed data parallel method on an internal cluster setup, featuring 2 x NVIDIA RTX A6000 GPUs, each with 48GB of GPU memory, and an Intel(R) Core(TM) i9-10980XE CPU. The same Intel i9-10980XE CPU is also utilized for data compression with the BPG method.

Remark 6 (Comparing Methods with Different Number of Users)**.** *We want to compare scenarios with different number of users fairly. Hence, we fix the per-user bandwidth ratio, $\bar{\rho}$. This would mean that if we increase the number of users and apply TDMA, we would still have the same bandwidth ratio for each image. In order to normalize also the available power, we fix the per-user average power constraint, defined as $\bar{P}_{\text{avg}} = {}^{P_{\text{avg}}}/_{n}$.*

Table 6.1 shows the model hyperparameters used in the experiments for fairness when comparing methods with different numbers of users. DeepJSCC-PNOMA method corresponds to different settings under different hyperparameters as shown in Table 6.1, e.g., it can simulate Perfect SIC and DeepJSCC-TDMA under specific hyperparameters.

6.4.3 Comparison with Point-to-Point and NOMA-Based Schemes

We employ CIFAR10 and TinyImagenet datasets for comparison, which are described in Section 6.4.1. As benchmark digital coding schemes for comparison, we employ BPG [153] and a variety of neural image compression codecs [35, 154, 155] in conjunction with 5G LDPC codes for channel encoding. After experimenting with different coding rates and QAM schemes using 5G LDPC codes with a blocklength of 6144 bits, we chose the optimal configuration. A natural extension of DeepJSCC to multi-user case is via time-division, named DeepJSCC-TDMA, where all the users share the same DeepJSCC encoder and decoder, and each user is only active during its own time slot. We report mean PSNRs and standard deviations wherever possible.

Figure 6.6 shows the superiority of our method for $n = 16$ compared to the separation-based methods (consisting of BPG or neural codecs combined with LDPC) and to DeepJSCC-TDMA. Our method achieves significantly better reconstruction performance in terms of PSNR for all the SNR values. Moreover, experiments over a Rayleigh fading channel yield results that are in line with those obtained under the AWGN channel, further demonstrating the robustness of our approach under different channel conditions. This observation also holds true for the MS-SSIM and LPIPS metrics (defined in Section 2.7), as demonstrated in Section 6.4.14. Section 6.4.13 further examines the fairness of DeepJSCC-PNOMA among users by evaluating the consistency of their reconstruction quality.

Figure 6.7 shows the comparison with DeepJSCC-NOMA [19], its curriculum learning extension DeepJSCC-NOMA-CL [19], and a genie-aided model based on Perfect SIC [19]. DeepJSCC-NOMA uses NOMA with a shared Siamese encoder and device embeddings for JSCC, enabling simultaneous image transmission and reconstruction. Its variant, DeepJSCC-NOMA-CL, first trains on non-interfering signals then fine-tunes on superposed transmissions to boost robustness and quality. Perfect SIC assumption-based model assumes no interference between users and is decoded only with channel noise and fading. DeepJSCC-PNOMA outperforms DeepJSCC-NOMA and DeepJSCC-NOMA-CL over all SNR values. DeepJSCC-PNOMA surprisingly outperforms even the Perfect SIC method when SNR < 5 dB for $\bar{\rho} = 1/6$ and SNR < 3 dB for $\bar{\rho} = 1/12$; implying that orthogonal initialization, training sample subsampling

and progressive fine-tuning-based training components are highly effective. We also note that Perfect SIC is an unrealistic assumption since signals naturally interfere with each other in real-world, and particularly in the case of DeepJSCC, it is impossible to decode and cancel interference completely.

6.4.4 Correlated Inputs

Figure 6.8 presents a comparative analysis between DeepJSCC-PNOMA and DeepJSCC-TDMA on the Cityscapes dataset in terms of the average PSNR. The results demonstrate that DeepJSCC-PNOMA consistently outperforms DeepJSCC-TDMA across all evaluated metrics, SNRs, and bandwidth ratios. This superior performance indicates the efficacy of our method in leveraging common information between correlated images. By incorporating this correlated information, DeepJSCC-PNOMA is able to more efficiently utilize the available bandwidth, leading to enhanced image reconstruction quality under varying SNRs. This observation also holds true for the MS-SSIM [128] and LPIPS [57] metrics (defined in Section 2.7), as demonstrated in Section 6.4.15.

6.4.5 Number of Users

Next, we evaluate our method on CIFAR10 and TinyImagenet datasets for varying numbers of users, n. This comparison remains fair by maintaining equal total power and total bandwidth, as described in Remark 6. Figure 6.9 illustrates the effect of increasing the number of users on DeepJSCC-PNOMA. As the number of users grows, the system's performance generally improves due to the potential for more concurrent transmissions. However, this improvement shows diminishing returns, where the incremental gain per additional user decreases. This phenomenon is attributed to increased interference with more users. Consequently, while adding users can initially boost the system performance, the rate of improvement gradually declines as the network nears its operational limits, consistent with the information-theoretic capacity of the MAC with respect to n.

Table 6.2: Number of trainable model parameters

Method	# Users	$\bar{\rho} = 1/6$	$\bar{\rho} = 1/12$
DeepJSCC-TDMA	All	22 200 211	22 149 011
DeepJSCC-PNOMA (Ours)	1	22 200 211	22 149 011
	2	22 202 259	22 149 523
	4	22 208 403	22 151 059
	8	22 232 979	22 157 203
	16	22 331 283	22 181 779
	32	22 724 499	22 280 083
	64	24 297 363	22 673 299
DeepJSCC-NOMA (-CL) [19]	2	22 382 846	22 260 478
Separate encoders & decoders	1	22 200 211	22 149 011
	2	44 402 470	44 298 534
	4	88 809 036	88 598 092
	8	177 634 456	177 200 280
	16	355 334 448	354 416 944
Perfect SIC	2	22 322 579	22 200 211
	16	26 699 667	23 615 891

6.4.6 Model Complexity

Table 6.2 compares the number of trainable parameters for DeepJSCC-TDMA, DeepJSCC-PNOMA, DeepJSCC-NOMA (-CL) [19], DeepJSCC-PNOMA without parameter sharing, and the Perfect SIC model over an AWGN MAC.

DeepJSCC-PNOMA with $n = 1$ matches DeepJSCC-TDMA in parameters. For $n = 2$, DeepJSCC-PNOMA has fewer parameters than DeepJSCC-NOMA and DeepJSCC-NOMA-CL while performing better. Note that DeepJSCC-NOMA (-CL) values are based on CIFAR10, and the parameter counts increase with the resolution of the inputs. DeepJSCC-PNOMA scales to $n = 16$ users with only 0.6% increase in the number of trainable parameters when $\bar{\rho} = 1/6$ and 0.1% increase when $\bar{\rho} = 1/12$, and further scales to $n = 64$ users with 9.4% increase when $\bar{\rho} = 1/6$ and 2.4% increase when $\bar{\rho} = 1/12$. As shown in Remark 3, the number of extra parameters compared to the $n = 1$ is $\mathcal{O}(n^2\bar{\rho}^2)$.

Practical implementation challenges such as real-time processing, channel estimation, and mobility are common to all machine learning-based communication systems and are beyond the

scope of this work, which focuses on the core algorithmic contributions of DeepJSCC-PNOMA. Nonetheless, our experiments demonstrate efficient inference with millisecond-level latency, scalable memory usage, and robust performance across diverse channel conditions. Technical specifications and computational efficiency metrics are detailed in the appendix.

Additionally, the network maintains high reconstruction quality while lowering training and inference costs thanks to the progressive expansion and fine-tuning strategy, which reuses previously trained components. The framework is appropriate for deployment in sizable, dynamic communication systems because it demonstrates a well-managed trade-off where incremental complexity permits scalable multi-user performance without sacrificing computational efficiency.

6.4.7 Ablation Study of Introduced Components

Figure 6.10 demonstrates the improvements resulting from various components on the validation split of the CIFAR10 dataset to prevent leakage from the test split. The first two plots in Fig. 6.10 clearly show that increasing the number of training pairs enhances the model's performance when fine-tuning from $n = 1$ to $n = 2$, albeit with longer training times. The last two plots illustrate the benefits of progressive fine-tuning-based training, orthogonal initialization of user-specific projections, and parameter sharing. Parameter sharing not only boosts performance but also reduces training time. Without progressive fine-tuning, we observe only marginal gains between $n = 2$ and $n = 16$. However, progressive fine-tuning provides a more stable method for increasing the number of users. Orthogonal initialization is the most effective component of our method, as it ensures that performance does not degrade due to randomness at the start of each fine-tuning step.

6.4.8 Orthogonality Analysis

We examine the orthogonality of encoding filters, which reflect geometric signal separation in DeepJSCC-PNOMA systems. For each user $u \in [n]$, we define m encoding filters $\{\mathbf{e}_i^{(u)}\}_{i=1}^{m}$,

reshaped into vectors in $\mathbb{R}^d$. The angle between any two filters is calculated as:

$$\theta_{ij}^{uv} = \arccos\left(\frac{\langle \mathbf{e}_i^{(u)}, \mathbf{e}_j^{(v)} \rangle}{\|\mathbf{e}_i^{(u)}\| \|\mathbf{e}_j^{(v)}\|}\right).$$

Figure 6.11 shows filter orthogonality for models trained on independent (CIFAR-10) and correlated (Cityscapes) images using two Kodak images transmitted over an AWGN channel at 3 dB. Kernel density estimations (KDEs) indicate angles mostly around 90°, with the model trained on Cityscapes dataset displaying greater angular variability, reflecting reduced orthogonality due to shared information between correlated sources. Heatmaps in the third and fourth panels highlight a clear block-diagonal structure distinguishing intra-user from inter-user relationships, resulting from progressive fine-tuning that initially enforces orthogonality. Thus, source correlation critically influences filter orthogonality: independent sources foster near-orthogonality, although perfect orthogonality is not achieved, while correlated sources encourage information sharing by further reducing orthogonal separation.

6.4.9 Qualitative Comparison of Reconstructed Outputs and Superposed Signals

Figure 6.12 compares the reconstructed images using different coding approaches, where DeepJSCC-PNOMA demonstrates notable visual improvement over other methods, validating the quantitative results presented in Section 6.4.3. Figure 6.13 visualizes the superposed filters while transmitting the images in Fig. 6.12 for the two-user case in both DeepJSCC-PNOMA and DeepJSCC-TDMA. In DeepJSCC-TDMA, one user transmits while the other remains silent; DeepJSCC-PNOMA, in contrast, allows both users to transmit simultaneously, making both images discernible from the superposed filters.

Table 6.3: Scalability analysis: Performance comparison on CIFAR-10 under AWGN channel for different bandwidth ratios. Values shown as PSNR (dB). **Bold values** indicate best performance within each bandwidth ratio group.

		SNR (dB)			
Bandwidth Ratio	**Users**	**0**	**3**	**7**	**10**
$\bar{\rho} = 1/6$	$n = 16$	25.96	28.37	31.28	33.27
	$n = 32$	26.03	28.42	31.33	**33.32**
	$n = 64$	**26.05**	**28.46**	**31.34**	33.29
$\bar{\rho} = 1/12$	$n = 16$	23.04	25.07	27.49	29.03
	$n = 32$	23.18	25.18	**27.57**	**29.08**
	$n = 64$	**23.20**	**25.19**	27.53	28.98

6.4.10 Analysis of Losses

Figure 6.14 illustrates the validation losses throughout the training process for bandwidth ratios $\bar{\rho} = 1/6$ and $\bar{\rho} = 1/12$. In both plots, the DeepJSCC-PNOMA model employing progressive fine-tuning exhibits a notably higher initial loss and converges to a higher final loss for both $n = 2$ and $n = 16$. This discrepancy is particularly pronounced at $\bar{\rho} = 1/6$. Similarly, the absence of orthogonal initialization results in elevated initial and final losses, potentially due to the model being trapped in a local optimum, likely caused by the interference from other users. These differences can probably be attributed to the random weights leading to increased interference from other users during the fine-tuning process. As observed with DeepJSCC-TDMA (equivalent to DeepJSCC-PNOMA with $n = 1$), DeepJSCC-PNOMA with $n = 2$, and DeepJSCC-PNOMA with $n = 16$, the final loss value decreases as the number of users n increases, aligning with the expected capacity gains of MACs with more users.

6.4.11 Scalability to Higher User Counts

To further validate the scalability and generalizability of our approach, we extend our experiments to systems with 32 and 64 users. Table 6.3 presents a comprehensive performance comparison across different user counts (16, 32, and 64) under AWGN channel conditions for both $\bar{\rho} = 1/6$ and $\bar{\rho} = 1/12$ power constraints.

The results demonstrate that DeepJSCC-PNOMA maintains consistent performance as the number of users scales from 16 to 64, with marginal improvements in PSNR and LPIPS metrics. Notably, the system shows stable performance across all SNR levels tested (0, 3, 7, and 10 dB), indicating robust scalability. For $\bar{\rho} = 1/6$, PSNR improvements range from 0.07-0.09 dB when scaling from 16 to 64 users, while for $\bar{\rho} = 1/12$, improvements range from 0.16-0.19 dB. The LPIPS metric shows similar stability with minimal variations across different user counts.

These results confirm that our progressive fine-tuning approach and orthogonal initialization strategy effectively handle the increased interference and complexity associated with larger user populations, making the proposed method viable for large-scale deployments.

Beyond image transmission, a variety of network architectures and dynamic scenarios can benefit from the core concepts of our work, including mobile and vehicular communications in 5G/6G systems, IoT sensor networks with time-varying device populations, and drone swarm coordination with rapidly changing topologies. The progressive user expansion and orthogonal projection design allow adaptive scaling as users join or leave the network, while the fine-tuning mechanisms could enable lightweight online updates under time-varying channels or user mobility. Although exploring such real-time adaptations lies beyond the current scope, the proposed framework provides a strong foundation for scalable and low-latency deep joint source–channel coding in next-generation communication networks.

6.4.12 Technical Specifications and Computational Efficiency

During evaluation, the model processes inference efficiently with millisecond-level latency for reconstruction across all user counts tested. The progressive fine-tuning strategy contributes to training stability while maintaining reasonable computational overhead compared to training from scratch for each user configuration.

Memory usage remains manageable even for larger user counts due to the shared encoder–decoder architecture and the parameter sharing mechanism described in Section 6.3. The system's ability to scale from 1 to 64 users while maintaining consistent performance demonstrates the practical

Table 6.4: Training durations and number of epochs passed before early stopping. Duration format is hours:minutes:seconds.

	Training Duration		Number of Epochs	
Method	$\bar{\rho} = 1/6$	$\bar{\rho} = 1/12$	$\bar{\rho} = 1/6$	$\bar{\rho} = 1/12$
DeepJSCC-PNOMA ($n = 1$)	1:04:38	1:50:16	54	80
DeepJSCC-PNOMA ($n = 2$)	4:42:27	2:09:58	86	40
DeepJSCC-PNOMA ($n = 4$)	3:49:57	2:34:43	42	28
DeepJSCC-PNOMA ($n = 8$)	10:41:58	7:14:31	31	21
DeepJSCC-PNOMA ($n = 16$)	20:24:06	3:40:45	33	11
DeepJSCC-PNOMA ($n = 2$)	4:42:27	2:09:58	86	40
DeepJSCC-PNOMA w/o progressive fine-tuning ($n = 2$)	8:12:00	7:22:08	86	56
DeepJSCC-PNOMA w/o orthogonal initialization ($n = 2$)	8:09:30	5:55:50	86	40
DeepJSCC-PNOMA w/o parameter sharing ($n = 2$)	5:43:02	7:58:39	86	59

viability of the proposed approach for real-world multi-user communication scenarios.

Runtime and Memory Discussion

We present the training durations, conducted in a shared, non-optimal environment with multiple processes and an uneven workload.

Training Durations: Table 6.4 presents the training durations and the number of epochs completed before early stopping for a subset of the trained models. CIFAR10 training durations range from 2 to 20 hours. TinyImageNet training durations range from 5 to 80 hours. Cityscapes training durations range from 45 to 160 minutes. Training durations mainly depend on the number of tuples T, per-user bandwidth $\bar{\rho}$, training dataset size $\mathcal{D}_{\text{train}}$, and the number of users n. The table highlights the effectiveness of techniques such as progressive fine-tuning, orthogonal initialization, and parameter sharing in accelerating the training process. As the number of users increases, these techniques become even more critical, facilitating faster model convergence in terms of epochs and reducing the computational burden associated with training. This not only enhances the performance of the DeepJSCC-PNOMA model but also ensures its efficiency and scalability across diverse multi-user communication environments. However, the increase in training duration with a higher number of users underscores the need for further computational optimizations, such as quantization and model pruning. Additionally, it is important to note that some training durations vary significantly due to the uneven workload on the servers during

the training process.

Evaluation Durations: All evaluations at a given SNR are completed in under one minute, with minimal variation, primarily influenced by the size of the validation and test datasets.

Memory: For evaluations, all experiments require less than 3 GB of GPU memory. For CIFAR10 training, approximately $1.5n$ GB of GPU memory is utilized per GPU. For TinyImageNet training, around 2.4 GB of GPU memory is used per GPU. For Cityscapes training, 12 GB of GPU memory is used for $n = 1$, and 32 GB of GPU memory is used for $n = 2$.

Software Requirements

The code to reproduce experiments requires the following software dependencies: Python 3.9.0 or higher, PyTorch (torch) version 2.1.0 or higher, Torchvision version 0.16.0 or higher, Lightning version 2.0.6, and Torchmetrics version 1.3.1. PyTorch provides a platform for building and training neural networks, Torchvision offers datasets and model architectures for computer vision, Lightning standardizes high-performance deep learning research, and Torchmetrics supplies performance metrics for model evaluation. Once Python and PyTorch are installed manually, all the necessary dependencies can be installed by running pip install −r requirements.txt in the main directory.

6.4.13 Analysis of Fairness

In this subsection, we analyze the fairness of DeepJSCC-PNOMA, ensuring that all users obtain comparable image reconstruction quality even under varying channel conditions—a critical attribute in multi-user communication systems. We formally define the fairness objective as follows:

Definition 1 (Fairness Objective)**.** *Let $PSNR_i$ denote the average PSNR for user i. The system is considered fair if the differences $|PSNR_i - PSNR_j|$ are negligible for all $i, j \in [n]$.*

To evaluate this metric, we conducted experiments on the Kodak dataset using a DeepJSCC-PNOMA model with $n = 4$ users for the first plot and $n = 2$ users for the second plot, both trained on CIFAR10 dataset on AWGN channel. All plots in Fig. 6.15 are derived from experiments on the Kodak dataset. Figure 6.15 shows the fairness analysis for the DeepJSCC-PNOMA method. The first plot presents the PSNR values for two users transmitting the same image under a fixed channel condition (SNR = 0 dB); the nearly identical PSNR values confirm that the system maintains fairness even under challenging noise conditions. The second plot, which displays PSNR performance across a range of SNR values, further demonstrates that the system consistently delivers uniform image reconstruction quality regardless of channel variations. Together, these results robustly validate our fairness criterion and underscore the capability of DeepJSCC-PNOMA to provide an equitable quality of service across different users.

6.4.14 Comparison on SSIM, MS-SSIM and LPIPS Metrics

In addition to PSNR, we evaluate using SSIM, MS-SSIM, and LPIPS. All metrics are defined in Section 2.7. For our evaluations, we use SSIM for CIFAR10 due to its resolution, MS-SSIM for TinyImagenet, and LPIPS for both datasets.

Figures 6.16 and 6.17 show the comparison of our method with separation-based alternatives combined with LDPC codes. These results align with our discussion in Section 6.4.3, despite being evaluated using different metrics. This consistency supports the generalization of our method to achieve high perceptual quality.

Figure 6.18 compares our method on the AWGN MAC with separation-based alternatives and capacity on CIFAR10 and TinyImageNet datasets. Achieving capacity is highly optimistic and generally not feasible in the real world. We do not use any specific practical channel coding or modulation schemes to determine this bound. Compressing the source at the maximum possible rate and assuming error-free transmission requires a capacity-achieving combination of channel coding and modulation for reliable transmission. Therefore, the performance of any separation-based transmission scheme using an actual channel coding scheme and modulation

with BPG compression will likely fall short of this upper bound. Despite this, our method performs significantly better in all evaluated SNRs. We also note that in low-SNR regime separation-based alternatives often fail to transmit as seen in the figure.

6.4.15 Additional Comparison on Correlated Inputs

Figure 6.19 presents a comparative analysis between DeepJSCC-PNOMA and DeepJSCC-TDMA on the Cityscapes dataset for MS-SSIM and LPIPS metrics. Similar to PSNR, for MS-SSIM and LPIPS, DeepJSCC-PNOMA obtains the best performance for all evaluated bandwidth ratios and SNRs.

6.4.16 Comparison for Varying Number of Actively Participating Users

Definition 2 (Comprehensiveness Objective)**.** *Let $PSNR_{\mathcal{P}}$ be the average PSNR over the whole test set when only a subset of users $\mathcal{P} \subset [n]$ transmit, i.e., $\mathbf{z}_i = \mathbf{0}, \forall i \notin \mathcal{P}$. The system is said to be comprehensive if $\forall \mathcal{P} \subset [n]$, $PSNR_{\mathcal{P}} \geq PSNR_{[n]}$.*

Figure 6.20 compares the performance of DeepJSCC-PNOMA with $n = 16$ users against scenarios with fewer actively transmitting users, specifically for $\mathcal{P} \in \{1, 2, 4, 8, 16\}$. This comparison enables us to evaluate the comprehensiveness objective outlined in Definition 2.

Notably, even though the DeepJSCC network for $n = 16$ is not explicitly trained for scenarios with fewer actively transmitting users, the performance remains remarkably consistent. This consistency is likely attributable to the near-orthogonality of the user signals, as discussed in Section 6.4.8.

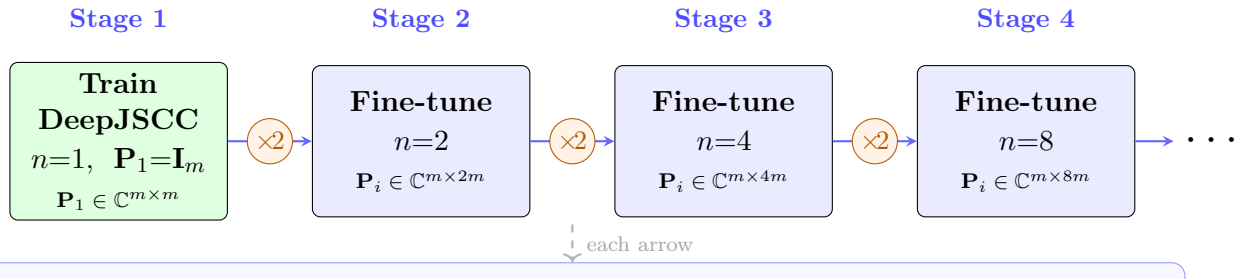
These results clearly demonstrate that the method can be effectively used with a variable number of actively transmitting users, aligning with the comprehensiveness objective across most channel conditions. We also posit that there is a potential performance gain resulting from

the reduced interference when fewer users are transmitting. This phenomenon can be explained by the increase in information-theoretic capacity under such conditions. In our experiments, we did not observe this effect because non-transmitting users were not included during the training phase. We leave the inclusion of non-transmitting users for future work.

6.5 Conclusion

We have introduced a novel joint image compression and transmission scheme for multi-user uplink scenarios, leveraging NOMA with identical DNN-based encoders and decoders for all users. Utilizing a user-specific projection trick, inspired by the CDMA scheme, the receiver can recover images from multiple users despite the analog transmission inherent in DeepJSCC, correctly attributing each image to its respective user. Our DeepJSCC-PNOMA scheme outperforms digital and DeepJSCC-based point-to-point alternatives. Furthermore, it scales up to 64 users with only an extra 9.4% of trainable parameters at $\bar{\rho} = 1/6$ and 2.4% at $\bar{\rho} = 1/12$, demonstrating consistent performance gains.

Figure 6.5: Progressive fine-tuning methodology for scaling DeepJSCC-PNOMA. *Top:* Starting from a single-user baseline (n=1, $\mathbf{P}_1$=$\mathbf{I}_m \in \mathbb{C}^{m\times m}$), the number of users is iteratively doubled; each stage shows the resulting encoder projection dimensions. *Bottom:* Detail of a single expansion step from $n/2$ to n users showing all matrix dimensions. Orthogonal matrices $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{C}^{\frac{n}{2}m\times nm}$ (obtained via QR factorization) multiply the existing projections to double their column dimension. Orthogonality ($\mathbf{K}_1\mathbf{K}_2^H = \mathbf{0}$) ensures zero initial inter-group interference; unitarity preserves signal power.



Expansion Step Detail: $n/2$ users $\longrightarrow$ n users

1. *Generate orthogonal sub-matrices:* Sample $\mathbf{M}\sim\mathcal{N}\big(\mathbf{0}, \frac{1}{nm}\mathbf{I}_{nm}\big)$, compute $\mathbf{Q}, \mathbf{R} = \mathrm{QR}(\mathbf{M})$, set $\mathbf{d} = \mathrm{Sign}\big(\mathrm{Diagonal}(\mathbf{R})\big)$, then partition:

$$\underbrace{\begin{bmatrix}\mathbf{K}_1\\ \mathbf{K}_2\end{bmatrix}}_{nm\times nm} = \mathbf{d}\cdot\mathbf{Q}, \qquad \mathbf{K}_1,\ \mathbf{K}_2\ \in\ \mathbb{C}^{\frac{n}{2}m\times nm}$$

2. *Update encoder projections* (users $1,\dots,n/2$ use $\mathbf{K}_1$; users $n/2{+}1,\dots,n$ use $\mathbf{K}_2$):

$$\mathbf{P}_i \leftarrow \underbrace{\mathbf{P}_{i,\mathrm{old}}}_{\in\,\mathbb{C}^{m\times\frac{n}{2}m}} \times \underbrace{\mathbf{K}_j}_{\in\,\mathbb{C}^{\frac{n}{2}m\times nm}} \in \mathbb{C}^{m\times nm}$$

3. *Update decoder projections:*

$$\mathbf{P}_i' \leftarrow \underbrace{\mathbf{K}_j^H}_{\in\,\mathbb{C}^{nm\times\frac{n}{2}m}} \times \underbrace{\mathbf{P}_{i,\mathrm{old}}'}_{\in\,\mathbb{C}^{\frac{n}{2}m\times m}} \in \mathbb{C}^{nm\times m}$$

Properties: $\mathbf{K}_1\mathbf{K}_2^H = \mathbf{0}$ (orthogonality $\Rightarrow$ zero initial inter-group interference); $\|\tilde{\mathbf{z}}_i\mathbf{P}_i\|^2 = \|\tilde{\mathbf{z}}_i\|^2$ (unitarity $\Rightarrow$ preserved signal power).

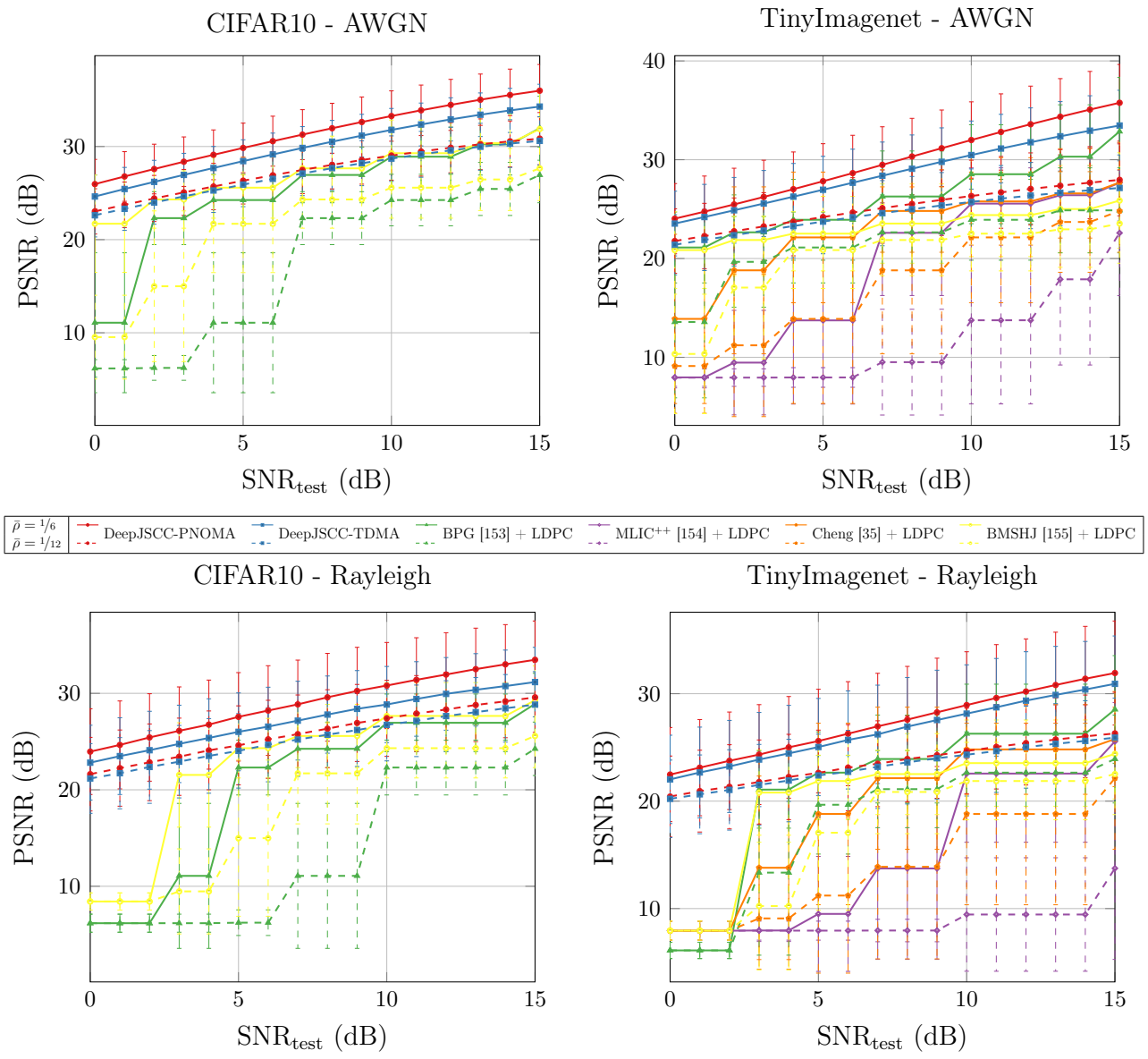


Figure 6.6: Comparison with point-to-point DeepJSCC and separation-based methods.

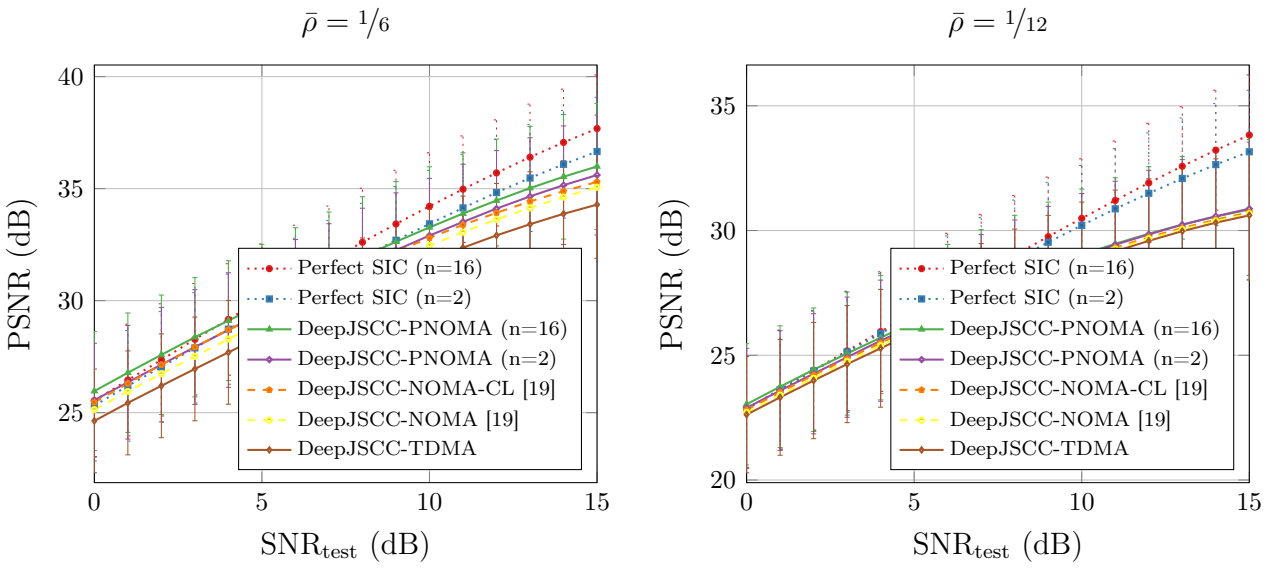


Figure 6.7: Comparison with NOMA-based methods for different bandwidth ratios on CIFAR-10.

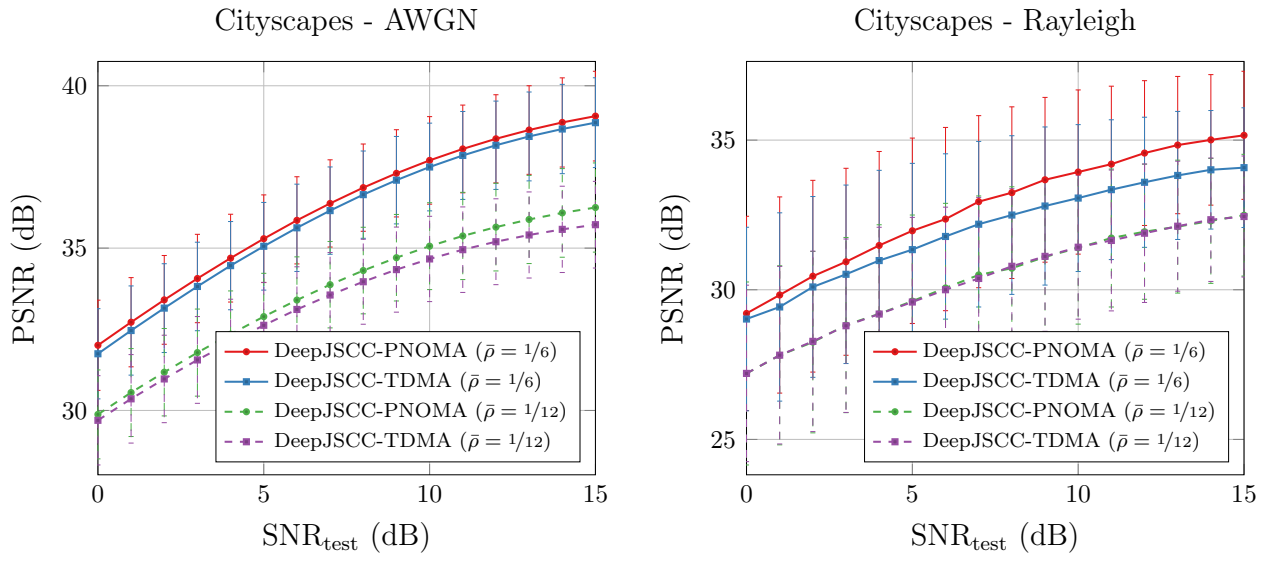


Figure 6.8: Performance comparison in terms of PSNR on AWGN and Rayleigh channels for correlated source images from Cityscapes dataset.

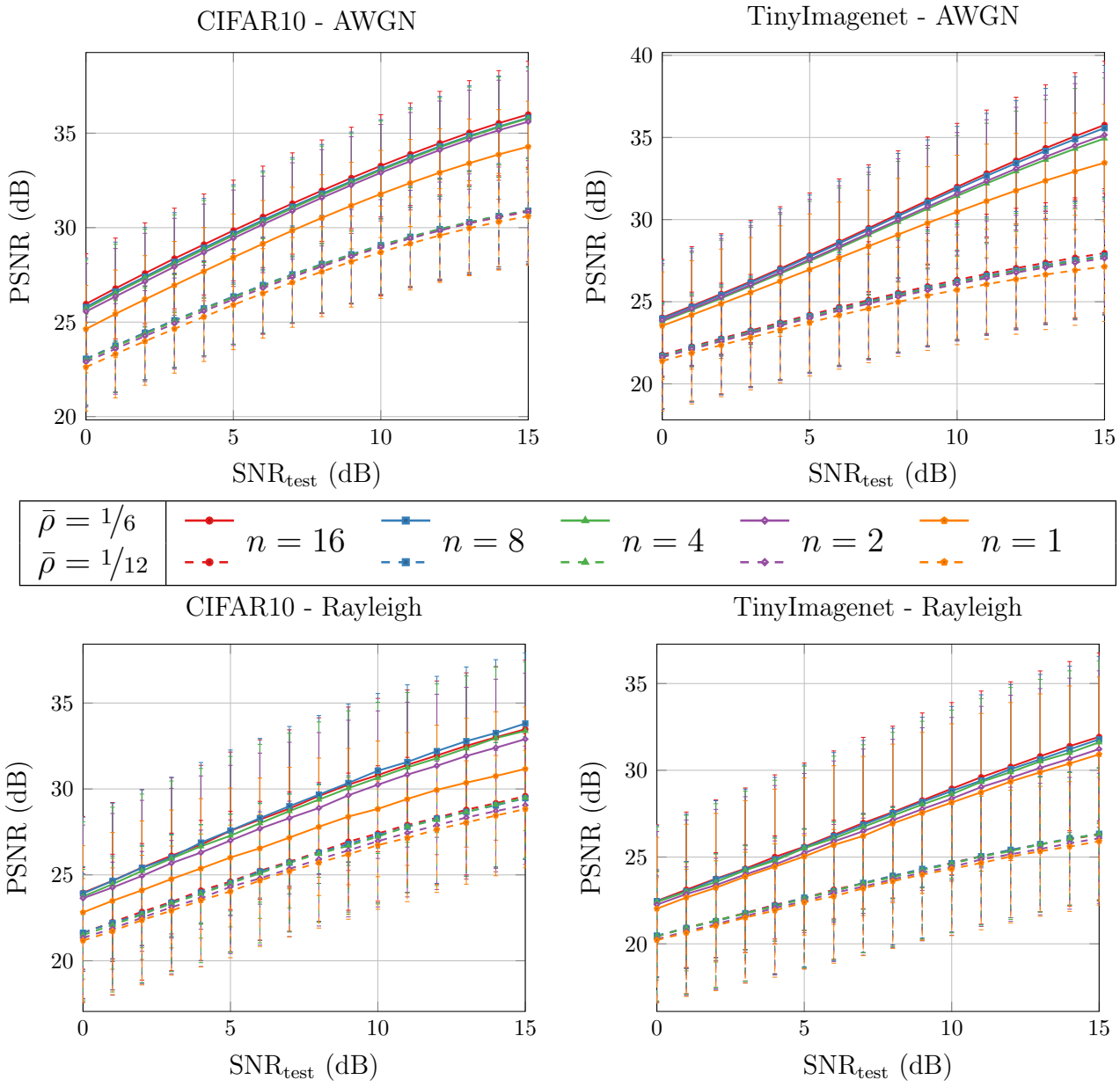


Figure 6.9: Comparison of DeepJSCC-PNOMA with number of users $n \in \{1, 2, 4, 8, 16\}$.

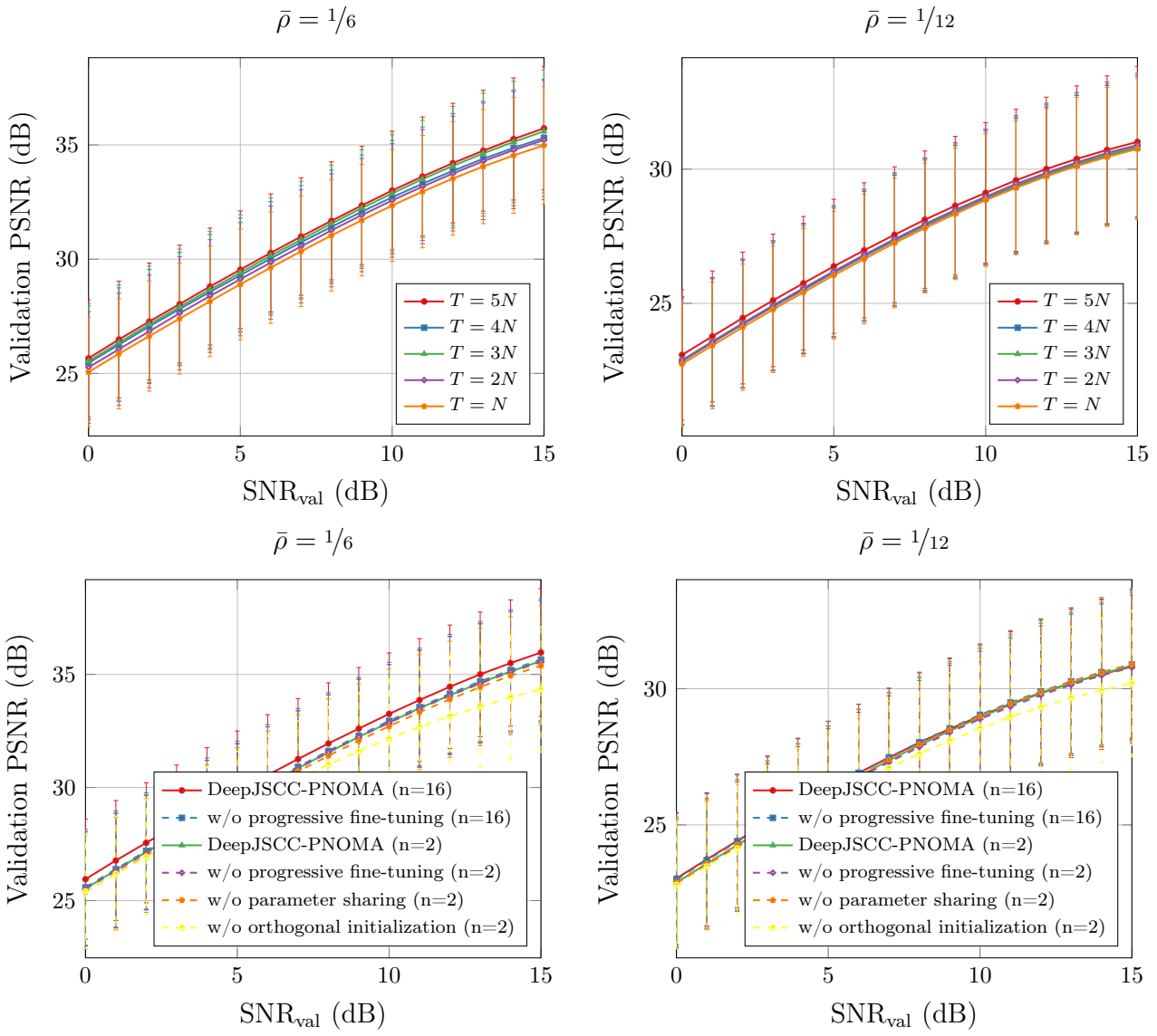


Figure 6.10: Ablation study of the number of the sampled training pairs and the introduced components.

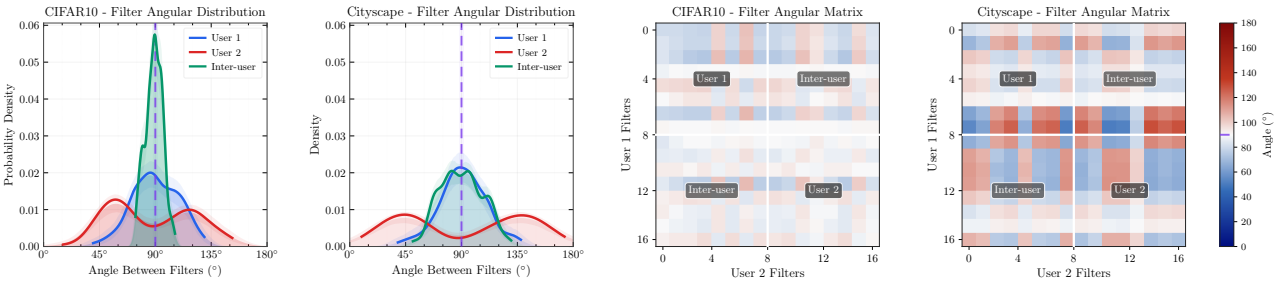


Figure 6.11: Analysis of orthogonality between transmitted filters on Kodak.

Figure 6.12: Qualitative comparison of reconstructed images.

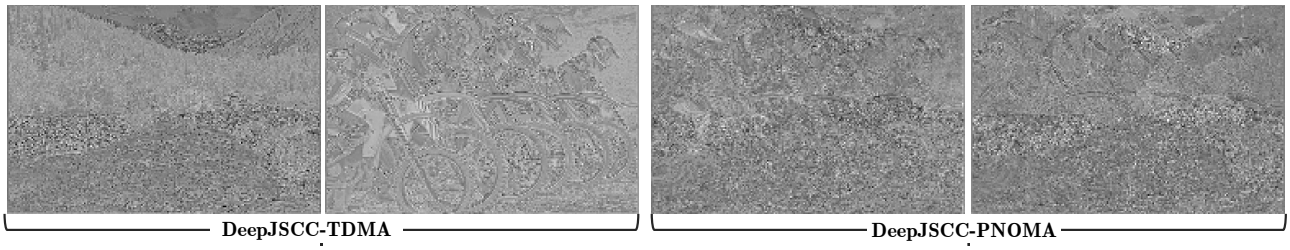


Figure 6.13: Visualization of transmitted filters for DeepJSCC-TDMA and DeepJSCC-PNOMA.

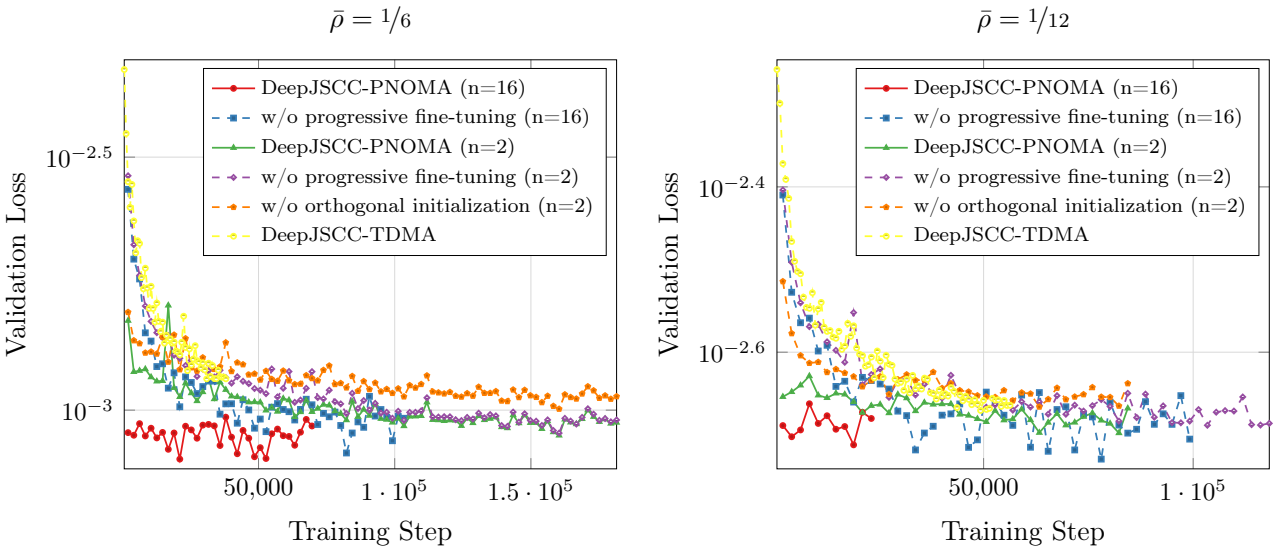


Figure 6.14: Visualization of losses throughout the training or fine-tuning steps.

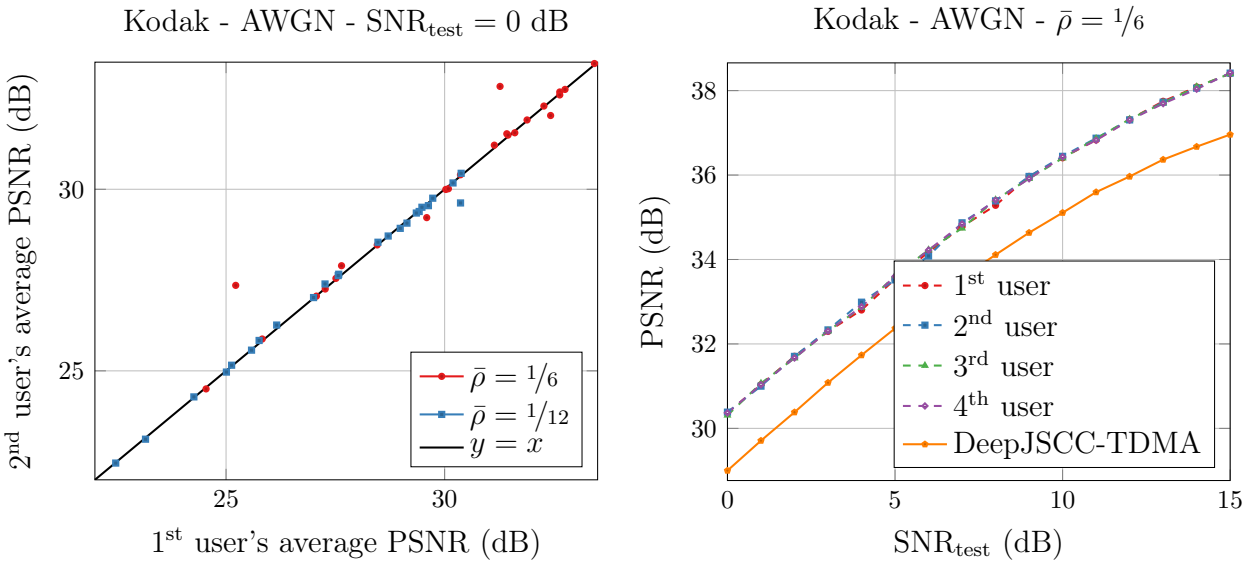


Figure 6.15: Analysis of fairness between users on Kodak dataset.

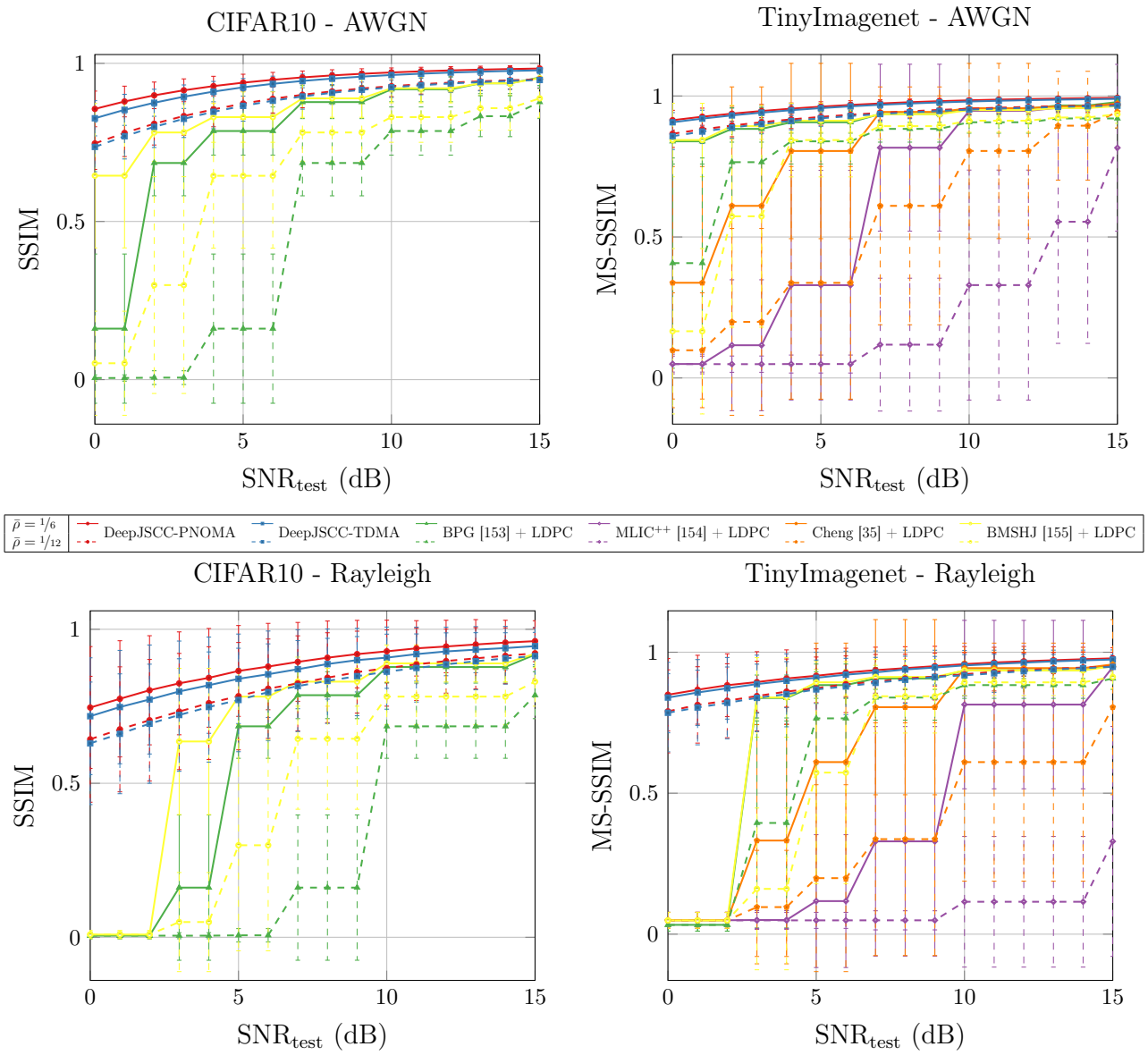


Figure 6.16: Comparison with point-to-point DeepJSCC and separation-based methods over SSIM and MS-SSIM metrics.

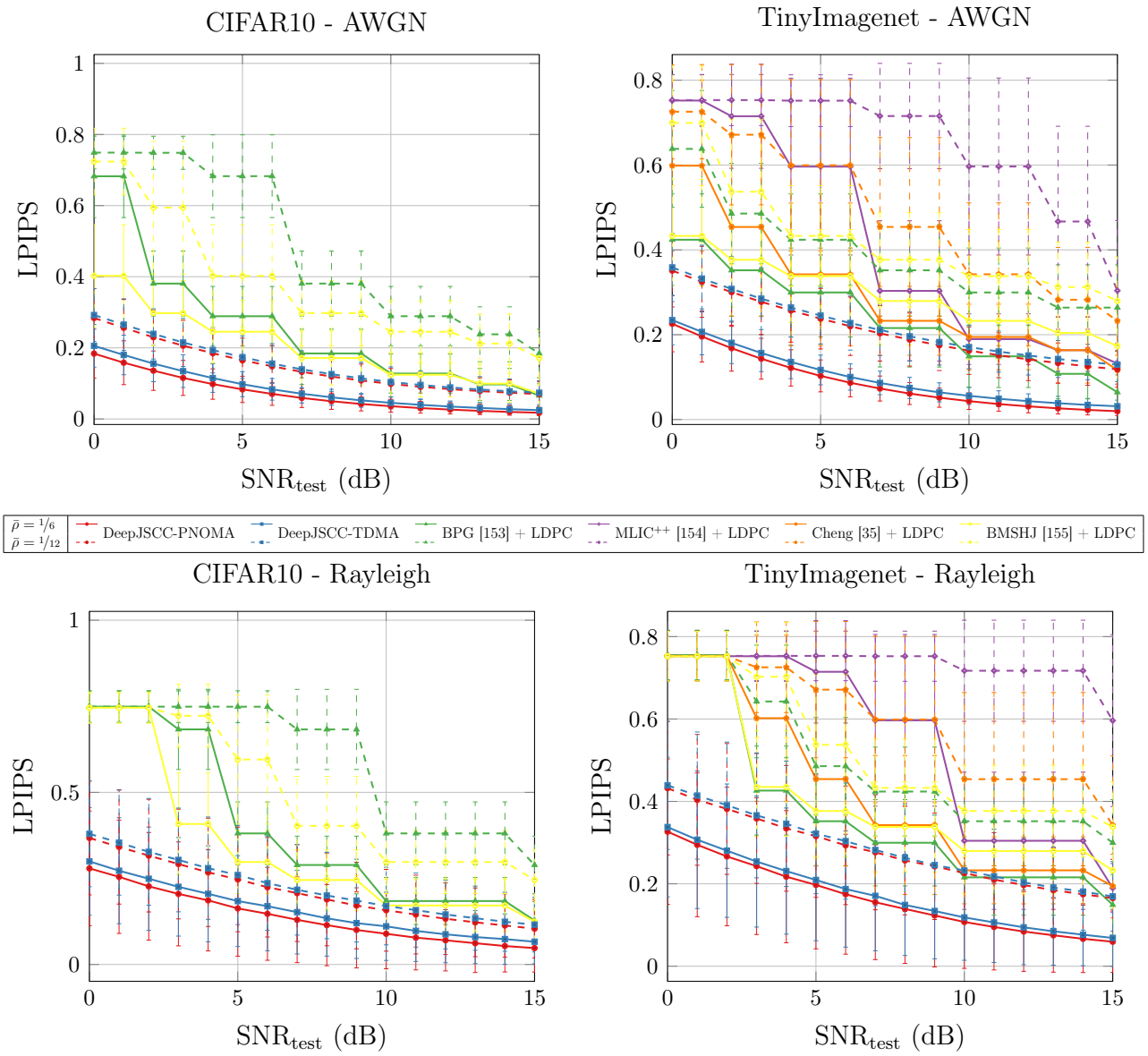


Figure 6.17: Comparison with point-to-point DeepJSCC and separation-based methods over SSIM, MS-SSIM and LPIPS metrics.

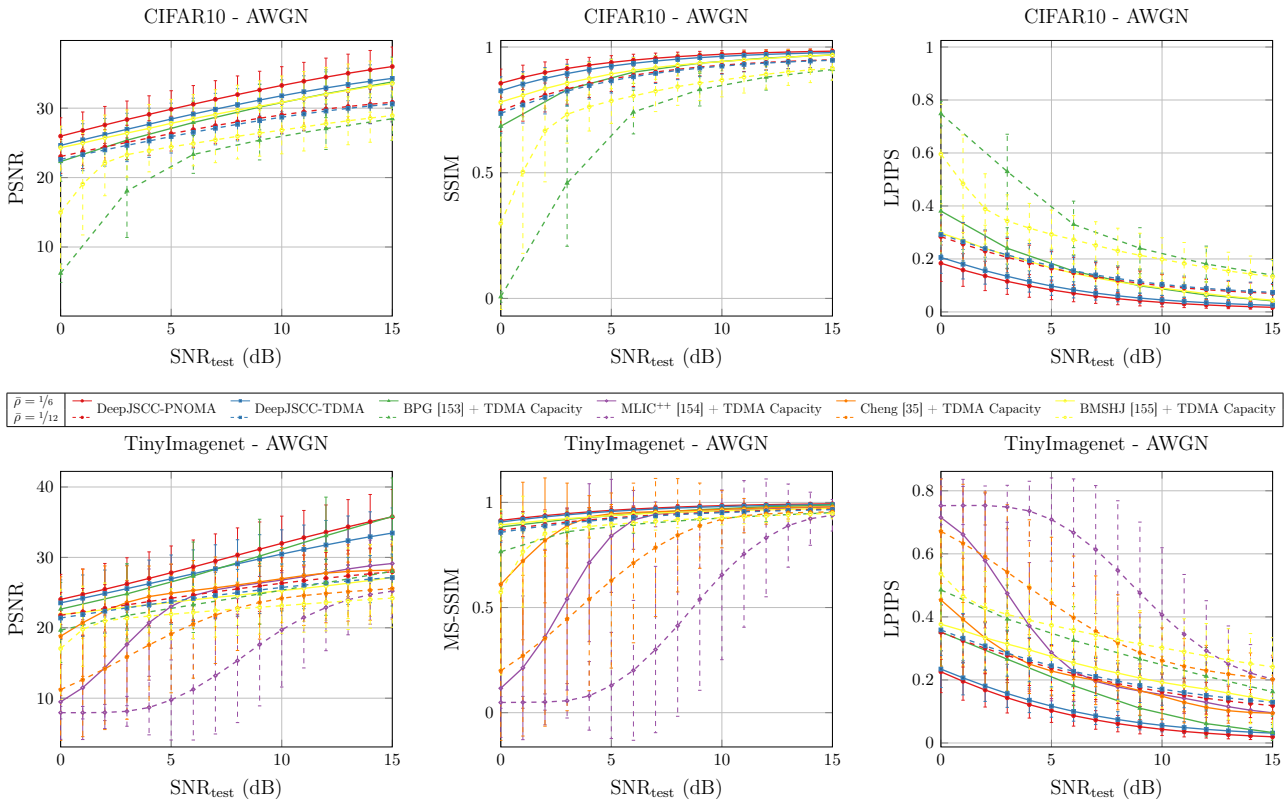


Figure 6.18: Comparison with point-to-point DeepJSCC and separation-based methods with capacity over PSNR, SSIM, MS-SSIM and LPIPS metrics.

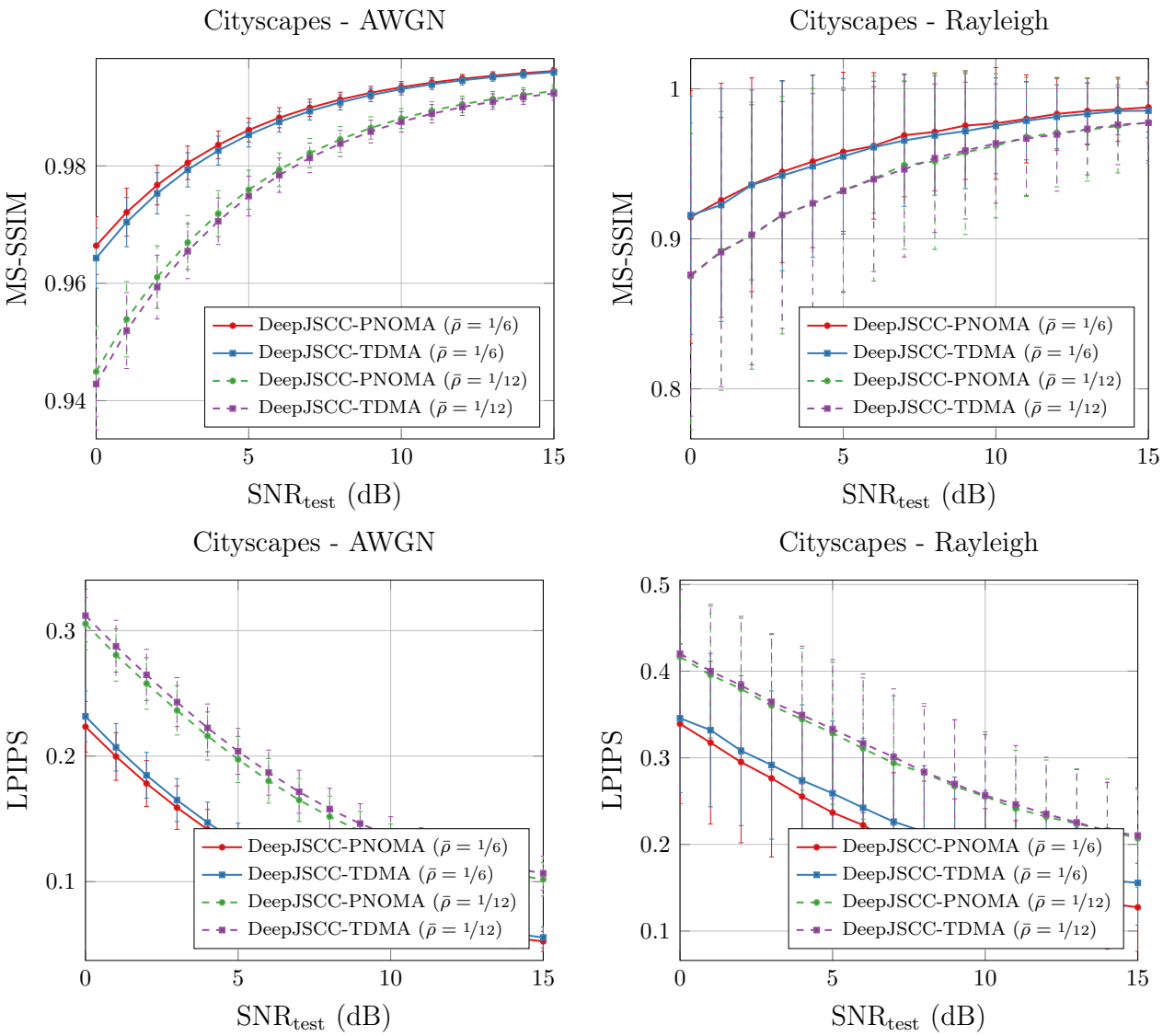


Figure 6.19: Performance comparison in terms of MS-SSIM and LPIPS on AWGN and Rayleigh channels for correlated source images from Cityscapes dataset.

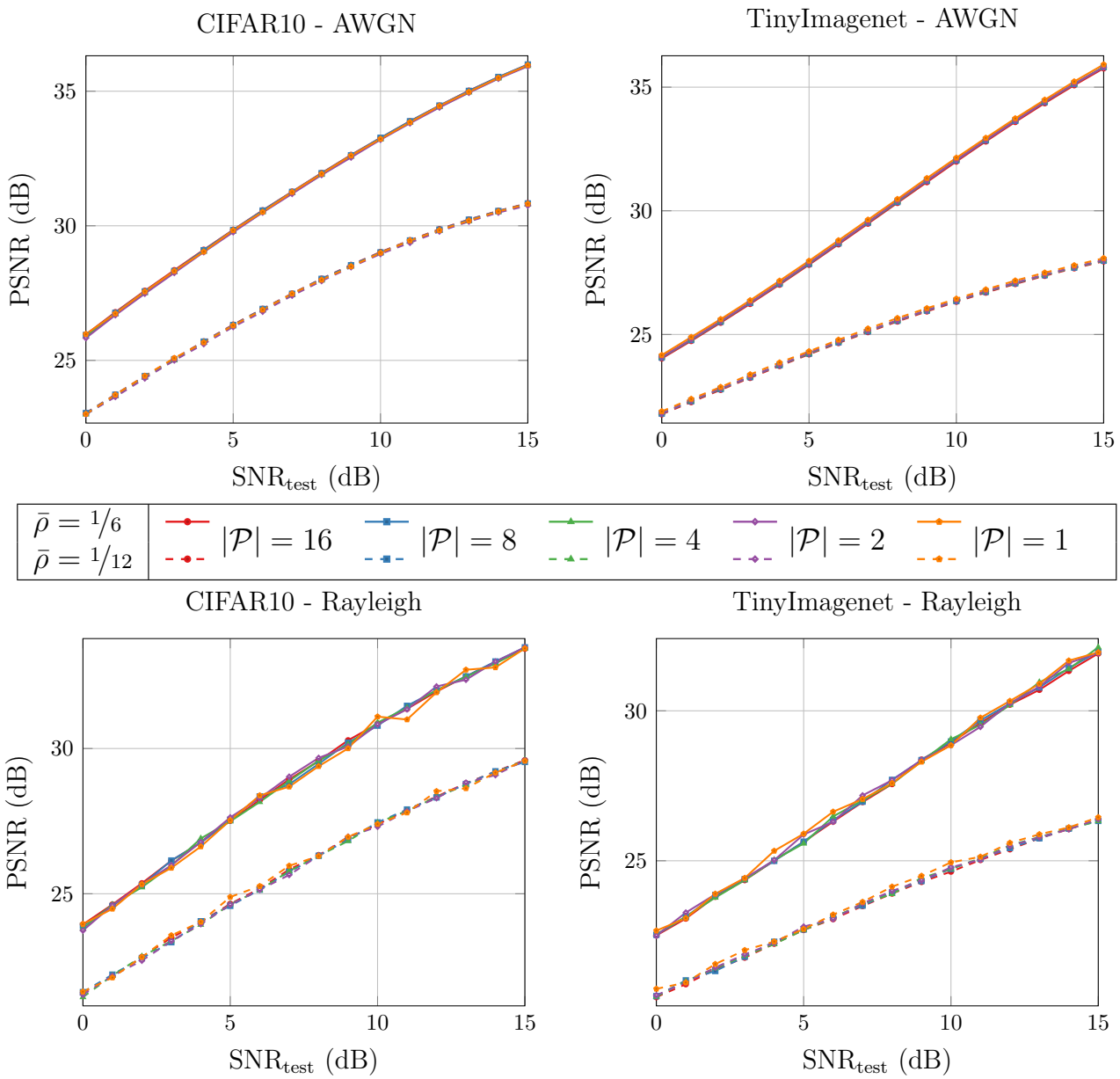


Figure 6.20: Comparison of DeepJSCC-PNOMA for number of active users $|\mathcal{P}| \in \{1, 2, 4, 8, 16\}$ to demonstrate the comprehensiveness of our method according to the Definition 2.

Chapter 7

Private Collaborative Edge Inference via Over-the-Air Computation

7.1 Introduction

The increasing adoption of IoT devices results in the collection and processing of massive amounts of mobile data at the wireless edge. As outlined in Chapter 1, conventional centralized machine learning methods are impractical for edge applications due to privacy concerns and limited communication resources. While previous chapters progressively extended DeepJSCC from single-user to multi-user image transmission, this chapter addresses a complementary large-scale multi-user challenge: collaborative edge inference across many distributed devices. Building upon the distributed inference and privacy considerations established in Chapter 2, this chapter introduces a novel privacy-preserving collaborative inference framework that leverages OAC to address the fifth research objective of this thesis.

Implementing decentralized ML models at the edge solves this issue, and thus, *edge learning* and *edge inference* have attracted significant attention over the recent years [85, 157, 88, 158, 159]. Edge learning aims to train large ML models in a distributed setting, whereas edge inference aims to make inferences in a distributed manner at the edge.

Although collaborative training (e.g., FL) at the edge can bring significant advantages, it requires coordination and communication across nodes. Moreover, limited wireless resources are a major bottleneck, and noise, interference, and lack of accurate channel state information can prevent or slow down the convergence of learning algorithms, or result in reduced accuracy [160]. Therefore, in this chapter, we consider collaborative inference using independently trained models at the edge nodes. While a growing body of work studies distributed training over wireless networks, the literature on distributed wireless inference, particularly using deep learning techniques, is relatively limited [30, 161, 162]. Moreover, beyond a few exceptions, such physical factors affecting edge inference have not been explored in the literature [85].

We treat the resultant problem as a collaborative edge inference problem, where the individual hypotheses of the clients need to be conveyed to the central inference server (CIS), and combined for the most accurate decision. Moving sensing to the edge reduces latency and improves privacy since it removes the requirement to communicate the data or the models. Often the data is not accumulated at a single client or cannot be shared between clients due to privacy or latency constraints. When data offloading or collaborative training are not possible, locally trained models are suboptimal and a collaborative inference solution is needed that incorporates the decisions at different clients. We, therefore, introduce a distributed inference solution that incorporates all the data and models while limiting privacy leakage during inference time through DP guarantees.

Privacy is an important concern in all ML applications since the data used for training can reveal sensitive information about its owner. In the case of ensemble inference, when the models are queried, their outputs may reveal sensitive information about their training sets. For instance, even when an adversary has black-box access to a model, whether or not a data sample is used during training can be inferred via membership inference attacks [163], or even the whole model can be reconstructed via model inversion attacks [164]. Hence, even if adversaries can only observe the inference results, we need to introduce some additional mechanisms to protect the sensitive information. Figure 7.1 summarizes the architecture for collaborative inference with DP guarantees.

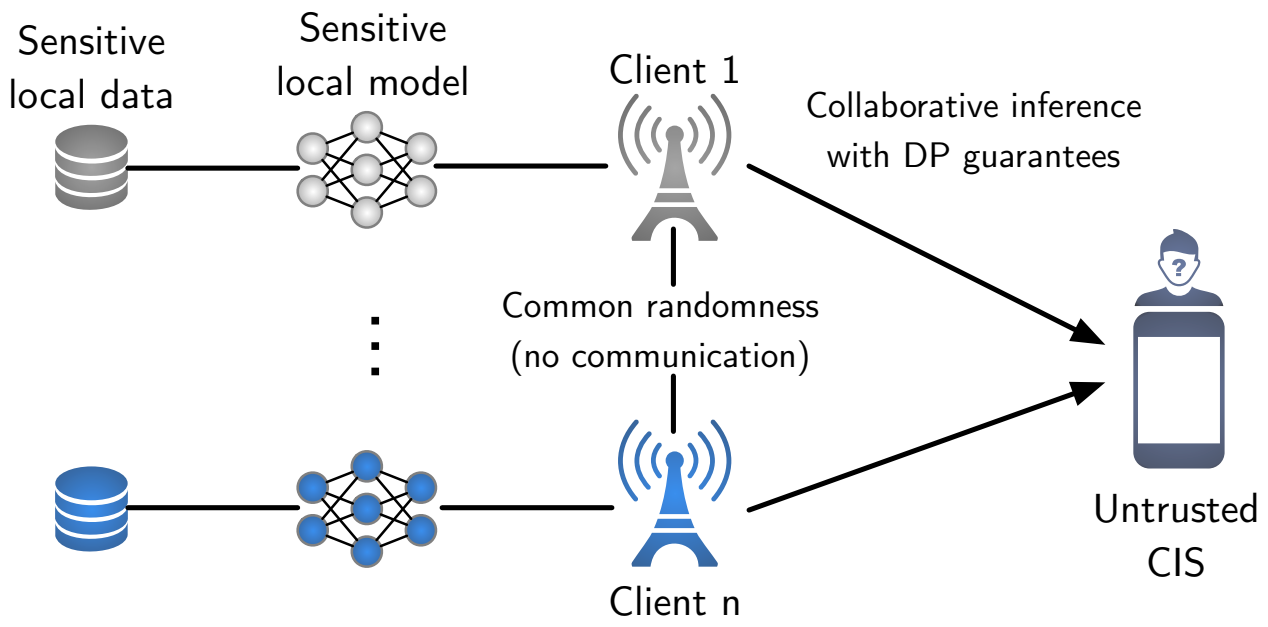


Figure 7.1: Overview of collaborative inference with DP guarantees.

DP guarantees can be obtained by introducing additional randomness to the output, such as adding noise, at the expense of some accuracy loss. Since DP bounds the amount of information leaked about the individuals, DP mechanisms make black-box attacks less effective. One approach to provide DP guarantees to ML is differentially private training [13]. Typically, Gaussian noise is added to the gradients during training, where the noise variance is determined according to the desired privacy level. This approach is extended to a federated setting in [165, 166]. However, DP training provides only a fixed DP guarantee, and we cannot operate at different privacy-utility trade-offs during inference, which may be beneficial when serving users with different levels of trustworthiness. Moreover, DP training does not prevent the model stealing attacks since the model can be still reconstructed via black-box access to it. Hence, in this chapter, we focus on embedding privacy-preserving mechanisms into the inference phase. We simply lift DP training assumption on the models and provide a systematic method for controllable privacy-utility trade-off for distributed inference.

In a recent line of work [90, 91, 96, 19], it has been shown that, in distributed training tasks, superposition property of the MAC can be exploited to use communication resources much more efficiently and to significantly improve the learning and transmission performance. Instead of conventional digital communication, in OAC, clients transmit their updates simultaneously

in an uncoded manner such that the receiver automatically gets the aggregated signal via the MAC (please refer to [167] for a comprehensive overview of OAC techniques and applications). Hence, besides communication efficiency, OAC also helps preserve the privacy of the clients since individual signals transmitted by the clients are not observed by the receiver [94]. Moreover, the noise and channel fading acting on the aggregated signal at the receiver can be effective at preserving the privacy of all the signals transmitted by the clients [89, 97, 168, 93, 94]. In this chapter, we extend the use of OAC beyond distributed training and exploit it for efficient and private distributed edge inference.

7.1.1 Contributions

The main contributions of our work are summarized as follows:

1. We introduce a novel distributed inference framework at the wireless edge that exploits OAC while providing bandwidth efficiency and variability. To the best of our knowledge, the conference version of this chapter [18] was the first in the literature to exploit OAC for distributed inference.

2. We provide flexible privacy guarantees depending on the scenario without imposing any restrictions on the training phase.

3. We systematically compare and discuss the privacy of the introduced collaborative classification methods, and show that the proposed framework with OAC performs significantly better than its orthogonal and digital counterparts while using fewer wireless resources under privacy constraints. We also provide statistical significance tests to show the reliability of the obtained results.

4. To facilitate further research and reproducibility, we publicly share the source code of our framework on github.com/ipc-lab/collaborative-inference-oac.

7.2 System Model and Problem Definition

7.2.1 System Model

We consider privacy-preserving multi-user classification at the wireless edge for both ensemble and multi-view cases. In our model, we consider n clients each with a separate model $f_i : \mathbb{R}^l \to \mathbb{R}^k, i \in [n]$, trained on dataset $\mathcal{D}_i \subset \mathcal{D}$ for either a multi-class or a binary classification task, where $\mathcal{D} = \bigcup_{i\in[n]} \mathcal{D}_i$, and $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for clients $i \neq j$. Each client also has access to a common validation dataset $\mathcal{D}_{\text{val}}$.

A new query to be classified arrives at each time step t, and all clients receive a view of the query, $\mathbf{x}_{i,t} \in \mathbb{R}^l$. Then, each participating client makes an inference with its local model denoted by $f_i(\mathbf{x}_{i,t}) \in \mathbb{R}^k$ and then encodes it into d channel symbols, which is the total available bandwidth per query.

Remark 7. *The target task becomes ensemble classification when all devices receive the same view $\mathbf{x}_t$ as query, i.e., $\mathbf{x}_{1,t} = \cdots = \mathbf{x}_{n,t} = \mathbf{x}_t$.*

The clients are connected to a CIS via a wireless medium, and, at time t, we assume each client i knows its channel gain to the CIS, $h_{i,t} \in \mathbb{R}$. In our setting, we assume that $h_{i,t}$'s are normally distributed with a zero mean and a variance σ_h^2. We further assume that the channel gains change across clients and time steps, but they stay the same per inference round; that is, the channel gain of a client is the same for all d channel uses. To reduce the total power consumption and to amplify the privacy guarantees, we consider random participation of the clients in each inference round such that each client i independently participates with probability p. To limit the power consumption, only the clients whose channel gains are larger than a certain threshold participate in the inference process. This is one of the sources of randomness determining p. Hence, p is a tunable parameter via such a transmission threshold. If necessary, via additional randomness, p can be made even smaller.

Let $\mathbf{y}_{i,t} \in \mathbb{R}^d$ denote the signal transmitted by client i. The received signal at CIS is

$$\mathbf{z}_t = \sum_{i \in \mathcal{P}_t} h_{i,t} \mathbf{y}_{i,t} + \mathbf{n}_t, \tag{7.1}$$

where $\mathbf{n}_t \in \mathbb{R}^d$ is the i.i.d. AWGN with zero mean and variance σ_n^2, i.e., $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_d)$. We enforce an average power constraint P, which is denoted by the following:

$$\mathbb{E}\left[\|\mathbf{y}_{i,t}\|_2^2\right] \leq P. \tag{7.2}$$

After receiving $\mathbf{z}_t \in \mathbb{R}^d$, CIS processes it via function $s : \mathbb{R}^d \to [k]$ and outputs the most probable class. The system model is illustrated in Fig. 7.2.

7.2.2 Threat Model and Target Problem

In our problem, the purpose is to limit the privacy leakage of clients' local models, which is equivalent to limiting the leakage about the individual datasets $\mathcal{D}_i, \forall i \in [n]$ since local models are trained on disjoint datasets. In our threat model, we assume all the clients follow the protocol, and the CIS is honest but curious, i.e., it does not deviate from the protocol, but by exploiting the signals received from the clients, it may try to infer sensitive information about the clients' models and datasets. Hence, our goal is to limit the leakage to CIS about the local models from $\mathbf{z}_t$ while trying to maximize the inference accuracy.

Remark 8. *Although our channel model is defined in real numbers $\mathbb{R}$ for simplicity, it can be extended to complex channels via mapping half of the vectors or function outputs to the real part and the rest to the imaginary part of a complex number.*

7.3 Methodology

In this section, we gradually introduce the modules of our framework.

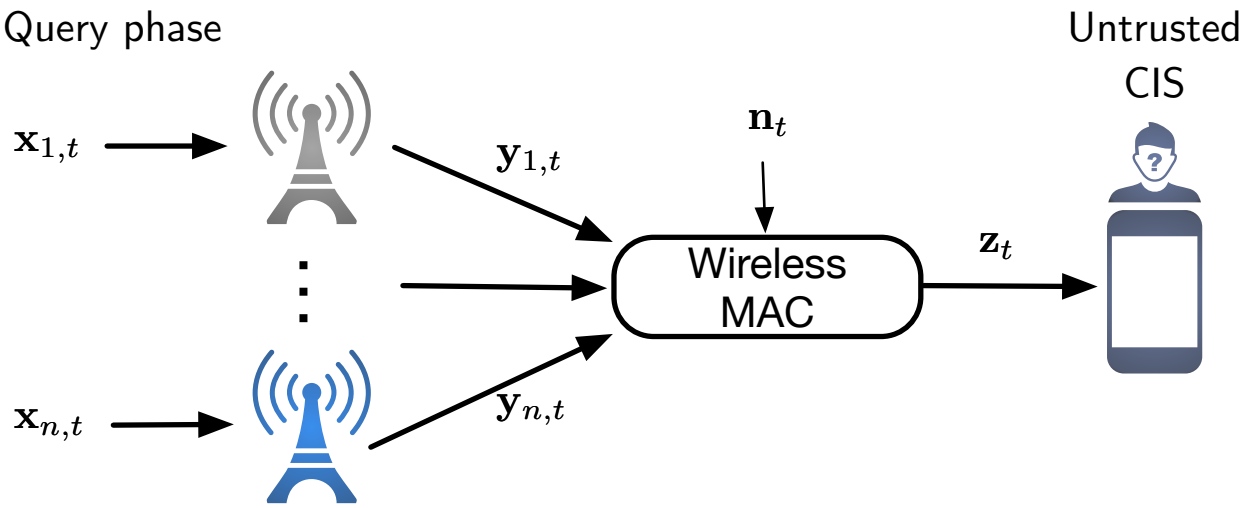


Figure 7.2: Overview of our framework for collaborative private inference at the wireless edge.

7.3.1 Fusion Methods

In the following, we present alternative methods to perform local prediction for participating clients, having received the query $\mathbf{x}_{i,t}$.

Let $\mathbf{r}_{i,t} \in \mathbb{R}^k$ be a vector containing classifier scores (beliefs) for each class at client i for the sample received at time slot t, where k is the number of classes, and the j^{th} element of $\mathbf{r}_{i,t}$, denoted by $(\mathbf{r}_{i,t})_j$, contains the score for class j. We normalize the sum of the scores in $\mathbf{r}_{i,t}$ to 1, i.e., $\|\mathbf{r}_{i,t}\|_1 = 1$, and hence, the maximum possible score of a class is 1.

We consider three fusion strategies, denoted *belief averaging with OAC (BA-OAC)*, *weighted belief averaging with OAC (WBA-OAC)*, and *majority voting with OAC (MV-OAC)*. For each strategy, we first define a local pre-processing function $\tilde{f}_i : \mathbb{R}^l \to \mathbb{R}^k$ that maps the received query $\mathbf{x}_{i,t}$ to a k-dimensional score vector. This score vector is subsequently mean-centered to yield the transmitted prediction $f_i(\mathbf{x}_{i,t})$, as detailed in Eq. (7.6). Before presenting the strategies, we introduce the following notation.

Definition 1. OneHot(j, l) *outputs an* l*-dimensional one-hot vector for* $j \leq l$*, where only the* j^{th} *element is* 1 *and the rest are* 0*.*

BA-OAC. Each client directly transmits its belief vector, and the CIS selects the class with

the highest aggregated score. The local pre-processing function is simply

$$\tilde{f}_i(\mathbf{x}_{i,t}) = \mathbf{r}_{i,t}. \tag{7.3}$$

WBA-OAC. Each client scales its belief vector by its normalized class-wise accuracy before transmission. The CIS then selects the class with the highest weighted aggregated score. The local pre-processing function is

$$\tilde{f}_i(\mathbf{x}_{i,t}) = \mathbf{w}_i \odot \mathbf{r}_{i,t}, \tag{7.4}$$

where $\mathbf{w}_i \in \mathbb{R}^k$ is a vector of normalized per-class accuracies on $\mathcal{D}_{\text{val}}$ for client i satisfying $\|\mathbf{w}_i\|_1 = 1$, and $\odot$ denotes element-wise multiplication. If real-time validation or test data are available without violating privacy constraints, they can be used to update $\mathbf{w}_i$ in place of $\mathcal{D}_{\text{val}}$.

MV-OAC. Each client casts a vote for its predicted class, and the CIS selects the class with the most votes. The local pre-processing function is

$$\tilde{f}_i(\mathbf{x}_{i,t}) = \text{OneHot}\left(\underset{j\in[k]}{\arg\max}(\mathbf{r}_{i,t})_j, k\right). \tag{7.5}$$

Note that *BA-OAC* and *WBA-OAC* fuse soft discriminative scores, whereas *MV-OAC* fuses hard predicted labels. After computing $\tilde{f}_i(\mathbf{x}_{i,t})$, we apply mean centering by subtracting its element-wise mean $\hat{f}_i(\mathbf{x}_{i,t})$, where $\left(\hat{f}_i(\mathbf{x}_{i,t})\right)_j = \frac{1}{k}$, $\forall j \in [k]$:

$$f_i\left(\mathbf{x}_{i,t}\right) = \tilde{f}_i(\mathbf{x}_{i,t}) - \hat{f}_i(\mathbf{x}_{i,t}). \tag{7.6}$$

This centering reduces transmission power since $\|f_i\left(\mathbf{x}_{i,t}\right)\|_2^2 < \left\|\tilde{f}_i(\mathbf{x}_{i,t})\right\|_2^2$. The reduction is particularly significant for *MV-OAC*, where the one-hot structure of $\tilde{f}_i(\mathbf{x}_{i,t})$ concentrates most of the energy in a single element that is largely removed by the mean subtraction.

7.3.2 Ensuring Privacy

Next, we explain how we make our inference procedure privacy-preserving by introducing some randomness. First, we formally define DP for our ensemble inference task.

Let $\mathcal{L}, \mathcal{L}' \in \mathbb{L}$ be the sets of local models of the clients, which differ at most in one of the clients, i.e., $\mathcal{L} = \{f_j\} \cup \{f_i : i \in [n] \setminus j\}$ and $\mathcal{L}' = \{f'_j\} \cup \{f_i : i \in [n] \setminus j\}$ such that $f_j \neq f'_j$ and $\mathbb{L}$ is the set of all possible local models of n clients. Such $\mathcal{L}$ and $\mathcal{L}'$ are called *neighbouring* sets. In our case, since we aim to protect the local models' privacy against CIS, we require that the CIS cannot distinguish between $\mathcal{L}$ and $\mathcal{L}'$ by observing $\mathbf{z}_t$. Hence, the DP guarantees considered in the chapter are all local-model-level DP, i.e., the replacement of one client's local model with another model does not generate a distinguishable effect on the observation of the CIS.

Local-model-level privacy guarantees might seem too conservative, but besides protecting against inferring sensitive features of the local datasets, local-model-level privacy provides protection against model stealing attacks, i.e., the CIS will not be able to accurately reconstruct the local model parameters.

Next, we formally define the notion of DP which characterizes the indistinguishability of $\mathbf{z}_t$ against the local model sets $\mathcal{L}$ and $\mathcal{L}'$. For this, for a clearer notation, let us consider the signal observed by the CIS, $\mathbf{z}_t$, as the output of a randomized function $M(\boldsymbol{X}_t, \mathcal{L})$, which represents all the local prediction, randomization and transmission phases, where $\boldsymbol{X}_t \in \mathbb{R}^{l \times n}$ is the matrix whose i^{th} column represents the query received by client i.

Definition 2. *Consider the signal observed by the CIS, $\mathbf{z}_t$, as the output of a randomized function $M : \mathbb{R}^{l \times n} \times \mathbb{L} \to \mathbb{R}^d$ and let $\mathcal{L}$ and $\mathcal{L}'$ be two possible neighbouring model sets. For $\varepsilon > 0$ and $\delta \in [0, 1)$, M is called (ε, δ)-DP if*

$$\Pr(M(\boldsymbol{X}_t, \mathcal{L}) \in \mathcal{R}) \leq e^{\varepsilon} \Pr(M(\boldsymbol{X}_t, \mathcal{L}') \in \mathcal{R}) + \delta, \tag{7.7}$$

for all neighbouring pairs $(\mathcal{L}, \mathcal{L}')$, $\forall \boldsymbol{X}_t \in \mathbb{R}^{l \times n}$ and $\forall\ \mathcal{R} \subset \mathbb{R}^d$.

In the above definition, ε indicates the amount of privacy loss and hence, smaller ε implies a

stronger privacy guarantee. On the other hand, δ characterizes the failure probability of this guarantee, i.e., bad events in which $\Pr(M(\mathcal{L}) \in \mathcal{R}) \leq e^{\varepsilon} \Pr(M(\mathcal{L}') \in \mathcal{R})$ does not hold. Hence, δ should be close to zero for a strong privacy guarantee. Choosing appropriate values for ε and δ is crucial for balancing between privacy and utility, and is still an active area of research. In principle, to have a strong privacy guarantee, choosing $\varepsilon \leq 1$ is preferable. It means that when a private sample is included, which is the private model of a device in our use case, the probability of any output value does not exceed more than ~ 2.72 times the probability of the same output value when the sample is not used. However, in practice, depending on the use case, considering privacy vs. utility trade-off, much larger ε values would be enough. For example, in [169], it has been demonstrated that using even a relatively large value of $\varepsilon = 8$ can be sufficient to deteriorate the performance of membership inference attacks. Moreover, in [170], it has been shown that using $\varepsilon < 100$ would be enough to prevent data reconstruction attacks in MNIST and CIFAR-10 datasets. For further discussions on how ε and δ should be chosen in practice, we refer the readers to [14, 171, 169, 170, 172, 173].

To achieve DP guarantees, the output released to an adversary should be randomized. In this chapter, we consider releasing a noisy version of model outputs, $f_i(\mathbf{x}_{i,t})$, for each client by adding Gaussian noise [102]. Each client can generate a noisy version of its model prediction as follows:

$$g_i(\mathbf{x}_{i,t}) = f_i(\mathbf{x}_{i,t}) + \mathbf{m}_{i,t}, \tag{7.8}$$

where $\mathbf{m}_{i,t} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\text{client}} \mathbf{I}_k)$. One of the main advantages of OAC is that the noise terms added by different clients are automatically aggregated at CIS. Thus, it has a further privacy amplification effect. We provide the analysis of the privacy guarantees achieved by our framework in Section 7.4. This analysis reveals that DP guarantees are directly dependent on the variance of the aggregated noise at the CIS. Hence, to obtain DP guarantees, each client should add a Gaussian noise with $\sigma^2_{\text{client}} = \sigma^2/|\mathcal{P}_t|$, where σ^2 is a constant depending on the desired DP guarantees and $\mathcal{P}_t$ is the set of participating clients. Since this noise variance depends on $|\mathcal{P}_t|$, each client must know $|\mathcal{P}_t|$; however, $\mathcal{P}_t$ must remain unknown to the CIS, as otherwise it could use this knowledge to better disaggregate the received superposed signal and infer individual

client contributions. We describe how the clients can jointly determine $|\mathcal{P}_t|$ without revealing $\mathcal{P}_t$ to the CIS via common randomness in Section 7.3.3.

Remark 9. *Note that in OAC, $\mathbf{z}_t$ already has channel noise that provides some degree of privacy guarantees, which can even be amplified by fading [97]. However, to achieve the desired level of DP, channel noise power may not be sufficient. First of all, channel noise may not follow a Gaussian distribution in practice (which is often used as worst-case assumption for channel capacity). More importantly, the clients cannot control or reliably know the noise variance or channel gain, as they rely on the CIS for this information. Thus, it is not a reliable source of randomness [94], and we ignore the channel noise and fading while analysing privacy guarantees. Instead, we have each client add some additional Gaussian noise before releasing their contributions. Hence, in reality, the privacy guarantees are slightly better than the ones we obtain in this chapter due to the presence of additional channel noise. If we were not ignoring the channel noise, each client would need to add less noise to the decision vectors, i.e., noise variance of $(\sigma^2 - \sigma_n^2)/|\mathcal{P}_t|$ would suffice instead of $\sigma^2/|\mathcal{P}_t|$.*

We discuss the DP guarantees of our method in Section 7.4. In the next section, we describe the transmission method of the clients.

7.3.3 Transmission

Before transmission, the clients determine the participation set $\mathcal{P}_t$ via common randomness. During an initial setup phase, all n clients agree on a shared random seed. At each time slot t, every client deterministically evaluates a pseudo-random number generator, seeded with the shared seed and the time index t, to draw n independent Bernoulli (p) outcomes, one per client. Since all clients use the identical seed and generator, they arrive at the same $\mathcal{P}_t$ (and hence $|\mathcal{P}_t|$) without any communication. Since the CIS does not know the seed, it cannot determine $\mathcal{P}_t$. If $|\mathcal{P}_t| = 0$, which occurs with very low probability, the clients repeat the process until at least one participates.

Furthermore, we use linear projection before transmission to adapt to the available bandwidth d.

When the number of classes k is smaller than d, it is reasonable to employ the whole bandwidth to improve the transmission quality against channel noise. However, the number of classes k can be much larger than d, e.g., the number of labels ranges from thousands to billions in extreme classification problems [174, 175]. In this case, to enable latency critical applications [176], it is better to project the decision vector $g_i(\mathbf{x}_{i,t}) \in \mathbb{R}^k$ to a d-dimensional space to reduce the bandwidth, albeit with lower reliability. Therefore, for more generality, we adopt a linear projection of the decision vector $g_i(\mathbf{x}_{i,t}) \in \mathbb{R}^k$ to dimension d.

Our objective is to recover the superposition of projected signals at the CIS with the least error possible, while we can expand or collapse the bandwidth. In particular, we employ the same projection matrix $\mathbf{P} \in \mathbb{R}^{d \times k}$ at all the clients to produce d-dimensional transmit vectors. To generate $\mathbf{P}$, we first sample $\mathbf{M} \sim \mathcal{N}(\mathbf{0}_c, \mathbf{I}_c)$, where $c = \{d, k\}$. Then, we apply QR factorization to $\mathbf{M}$ using the procedure in [152] to generate an orthogonal matrix $\mathbf{Q}$ as:

$$\begin{aligned} \mathbf{Q}^{'}, \mathbf{R} &= \text{QRFactorization}\left(\mathbf{M}\right), \\ \mathbf{d} &= \text{Sign}\left(\text{Diagonal}\left(\mathbf{R}\right)\right), \\ \mathbf{Q} &= \mathbf{d} \cdot \mathbf{Q}^{'}, \end{aligned}$$

where Diagonal extracts the diagonal elements of the input matrix as a vector, and Sign replaces negative elements with -1 and positive elements with $+1$. Then, we select the first d rows and k columns as $\mathbf{P} = \mathbf{Q}_{1:d,1:k}$. Since the CIS is only interested in the final decision, it only needs to pinpoint the maximum value of the superposed decision vector rather than estimating the entire decision vector. In addition, most of the clients may make the same decision; hence, it is feasible to identify the maximum value even when $d < k$.

Remark 10. *Any orthogonal matrix can be used instead of* $\mathbf{Q}$*. When* $d < k$*,* $\mathbf{P}$ *can also be designed as a matrix, e.g., a Gaussian or Bernoulli random matrix, that satisfies the restricted isometry property of compressed sensing [177].*

We need to make sure that each client's noisy decision vector $g_i(\mathbf{x}_{i,t})$ is received at the CIS at the same power level. Recall that the channel gain for each client is assumed to be perfectly

known by that client, which then employs channel inversion to cancel its effect. Thus, each client scales its signal by $1/h_{i,t}$ prior to transmission. Note that, since a client does not participate in the inference process if its channel has a low gain, this scaling does not result in excessive power usage. The CIS may require a specific power level for the reception of the signals depending on the available power of the clients or the channel's power constraint. Typically, an average power constraint is imposed. We satisfy the average power constraint in Eq. (7.2) by scaling via γ, whose calculation is provided in Section 7.3.5. Lastly, we normalize by $|\mathcal{P}_t|$ so that the receiver receives the average of all the users.

After these steps, the signal transmitted by the i^{th} client is given by

$$\mathbf{y}_{i,t} = \begin{cases} \gamma \mathbf{P} \dfrac{g_i(\mathbf{x}_{i,t})}{|\mathcal{P}_t| h_{i,t}}, & \text{if } i \in \mathcal{P}_t, \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{7.9}$$

where $\mathcal{P}_t$ is the set of participating clients.

Now, all participating clients transmit their signals synchronously over a MAC. Following the standard assumption in OAC literature [91, 96], we assume that all clients are symbol-level synchronized, which can be achieved in practice via standard mechanisms such as timing advance and pilot signals from the CIS. Under this assumption, their signals are superposed at the CIS. Nevertheless, this assumption may not hold in large-scale networks, where achieving synchronization is challenging due to hardware variations, channel conditions, and geographical distances. In asynchronous settings, synchronization errors can lead to significant interference, reducing overall system accuracy. For approaches to mitigate these issues, we refer the reader to [178, 167].

7.3.4 Final Decision by the CIS

The signal received by the CIS at time t after passing through the channel defined in Eq. (7.1) is given as follows:

$$\begin{aligned}\mathbf{z}_t &= \sum_{i\in\mathcal{P}_t} h_{i,t}\mathbf{y}_{i,t} + \mathbf{n}_t \\ &\overset{(a)}{=} \sum_{i\in\mathcal{P}_t} h_{i,t}\gamma\mathbf{P}\frac{f_i(\mathbf{x}_{i,t}) + \mathbf{m}_{i,t}}{|\mathcal{P}_t|h_{i,t}} + \mathbf{n}_t \\ &\overset{(b)}{=} \frac{\gamma\mathbf{P}}{|\mathcal{P}_t|}\sum_{i\in\mathcal{P}_t}\left(f_i(\mathbf{x}_{i,t}) + \mathbf{m}_{i,t}\right) + \mathbf{n}_t \\ &= \frac{\gamma\mathbf{P}}{|\mathcal{P}_t|}\sum_{i\in\mathcal{P}_t} f_i(\mathbf{x}_{i,t}) + \frac{\gamma\mathbf{P}}{|\mathcal{P}_t|}\sum_{i\in\mathcal{P}_t}\mathbf{m}_{i,t} + \mathbf{n}_t, \end{aligned} \tag{7.10}$$

where (a) follows from Eqs. (7.8) and (7.9) and (b) follows from the linearity properties of the projection and scalars. After receiving $\mathbf{z}_t$, CIS decision function $s(\cdot)$ multiplies the received signal by $\frac{1}{\gamma}\mathbf{P}^T$ to recover the desired signal, which is the average of votes, local beliefs or weighted local beliefs:

$$\begin{aligned}\tilde{s}\left(\mathbf{z}_t\right) &= \frac{1}{\gamma}\mathbf{P}^T\mathbf{z}_t \\ &= \frac{\mathbf{P}^T\mathbf{P}}{|\mathcal{P}_t|}\sum_{i\in\mathcal{P}_t} f_i(\mathbf{x}_{i,t}) + \mathbf{b}_t, \end{aligned} \tag{7.11}$$

where $\mathbf{b}_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}^T\sigma_n^2 + \mathbf{P}^T\mathbf{P}\gamma^2\sigma_{\text{client}}^2/|\mathcal{P}_t|\right)$ is the accumulated noise due to privacy and wireless channel. Note that when $d \geq k$, we have $\mathbf{P}^T\mathbf{P} = \mathbf{I}_k$ due to the orthogonality property; and therefore, the CIS reconstructs the average of participating clients' score vectors along with some noise.

Then, the CIS applies the arg max function to decide the most probable class, i.e.,

$$s(\mathbf{z}_t) = \underset{j\in[k]}{\arg\max}\, \tilde{s}\left(\mathbf{z}_t\right)_j . \tag{7.12}$$

Algorithm 9 summarizes all the steps introduced in this section. In the next section, we derive the scaling factor γ.

Algorithm 9: OAC-Based Private Collaborative Inference

Require: Trained client model $f_i(\cdot)$ for every client i, CIS model $s(\cdot)$, new samples $\mathbf{x}_{i,t}$ $\forall i \in [n]$ at time step t

Ensure: Index of the decided class

function PRIVATE_COLLABORATIVE_INFERENCE

Let $\mathcal{P}_t$ contain each client i with probability p, determined via a common seed among clients.

for all client $i \in \mathcal{P}_t$ in *parallel* **do**

Client i receives $\mathbf{x}_{i,t}$

Calculate $f_i(\mathbf{x}_{i,t})$ ▷ *Client Model*

$g_i(\mathbf{x}_{i,t}) = f_i(\mathbf{x}_{i,t}) + \mathcal{N}(\mathbf{0}, \sigma^2/|\mathcal{P}_t|\mathbf{I}_k)$ ▷ *Add noise*

Transmit $\gamma \mathbf{P} g_i(\mathbf{x}_t)/\left(|\mathcal{P}_t| h_{i,t}\right)$

$\mathbf{z}_t = \mathbf{n}_t + \gamma \sum_{i \in \mathcal{P}_t} g_i(\mathbf{x}_{i,t})$ ▷ *Air Sum*

CIS receives $\mathbf{z}_t$

return $s(\mathbf{z}_t)$ ▷ *CIS Model*

7.3.5 Determining the Scaling Factor

Here, we derive the scaling factor γ to normalize the prediction vectors with additional noise introduced due to privacy according to the average power constraint defined in Eq. (7.2). In our setting, each client i wants to transmit $\mathbf{g}_{\mathbf{p}_i} \triangleq \mathbf{P} g_i(\mathbf{x}_{i,t})$ to the CIS. For a clearer notation, we drop the time index for the rest of this subsection. To account for channel gains and to satisfy the average power constraint in Eq. (7.2), before transmitting $\mathbf{g}_{\mathbf{p}i}$, we scale it by $\frac{\gamma}{|\mathcal{P}_t| h_i}$, where γ is a scalar chosen to satisfy the average power constraint. In the following, we derive it in detail. For $i \in [n]$, we have,

$$\mathbb{E}\left[\|\mathbf{y}_i\|_2^2\right] = \mathbb{E}\left[\left\|\frac{\gamma}{|\mathcal{P}_t| h_i} \mathbf{g}_{\mathbf{p}_i}\right\|_2^2\right] \leq P. \tag{7.13}$$

We have

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{y}_i\|_2^2\right] &= \frac{\gamma^2}{|\mathcal{P}_t|^2} \mathbb{E}\left[\left\|\frac{1}{h_i} \mathbf{g}_{\mathbf{p}_i}\right\|_2^2\right] \\
&= \frac{\gamma^2}{|\mathcal{P}_t|^2} \sum_{j=1}^{d} \mathbb{E}\left[\frac{(\mathbf{g}_{\mathbf{p}_i})_j^2}{h_i^2}\right] \\
&\overset{(a)}{=} \frac{\gamma^2}{|\mathcal{P}_t|^2} \sum_{j=1}^{d} \mathbb{E}\left[(\mathbf{g}_{\mathbf{p}_i})_j^2\right] \mathbb{E}\left[\frac{1}{h_i^2}\right],
\end{aligned} \tag{7.14}$$

where (a) follows from independence between the elements of $\mathbf{g}_{\mathbf{p}_i}$ and h_i. Note that each client transmits only if $h_i^2 \geq h_{\min}$. Since h_i is normally distributed with zero mean and variance σ_h^2, we have

$$\begin{aligned}\mu_{1/h} \triangleq \mathbb{E}\left[\frac{1}{h_i^2}\right] &= \frac{2}{\sqrt{2\pi}\sigma_h}\int_{\sqrt{h_{\min}}}^{\infty}\frac{1}{x^2}e^{-\frac{x^2}{2\sigma_h^2}}dx \\ &= \frac{1}{\sqrt{2\pi}\sigma_h}\int_{h_{\min}}^{\infty}\frac{1}{h}e^{-\frac{h}{2\sigma_h^2}}dh \\ &= \frac{1}{\sqrt{2\pi}\sigma_h}\int_{h_{\min}/(2\sigma_h^2)}^{\infty}\frac{1}{h}e^{-h}dh \\ &= \frac{1}{\sqrt{2\pi}\sigma_h}E_1\left(\frac{h_{\min}}{2\sigma_h^2}\right), \end{aligned} \tag{7.15}$$

where $E_1(x) \triangleq \int_x^{\infty}\frac{e^{-t}}{t}dt$. Hence Eq. (7.14) can be written as

$$\frac{\gamma^2}{|\mathcal{P}_t|^2}\mu_{1/h}\sum_{j=1}^{d}\mathbb{E}\left[(\mathbf{g}_{\mathbf{p}_i})_j^2\right] = \frac{\gamma^2}{|\mathcal{P}_t|^2}\mu_{1/h}\left\|\mathbf{g}_{\mathbf{p}_i}\right\|_2^2. \tag{7.16}$$

Recall that $\mathbf{g}_{\mathbf{p}_i} = \mathbf{P}g_i(\mathbf{x}_{i,t}) = \mathbf{P}\left(f_i(\mathbf{x}_{i,t}) + \mathbf{m}_{i,t}\right)$. Since $\mathbf{m}_{i,t}$ consists of independent Gaussian random variables with variance σ_{client}^2, we have

$$\frac{\gamma^2}{|\mathcal{P}_t|^2}\mu_{1/h}\left\|\mathbf{g}_{\mathbf{p}_i}\right\|_2^2 = \frac{\gamma^2}{|\mathcal{P}_t|^2}\mu_{1/h}\left(\left\|\mathbf{P}f_i(\mathbf{x}_{i,t})\right\|_2^2 + d\sigma_{\text{client}}^2\right). \tag{7.17}$$

To bound $\|\mathbf{P}f_i(\mathbf{x}_{i,t})\|$, first, let us consider $\mathbf{P} \in \mathbb{R}^{d\times k}$, where $d \geq k$. In this case, $\mathbf{P}$ consists of k orthogonal columns. Hence, we have

$$\begin{aligned}\left\|\mathbf{P}f_i(\mathbf{x}_{i,t})\right\|_2^2 &= f_i^T(\mathbf{x}_{i,t})\mathbf{P}^T\mathbf{P}f_i(\mathbf{x}_{i,t}) \\ &= f_i^T(\mathbf{x}_{i,t})\mathbf{I}_k f_i(\mathbf{x}_{i,t}) \\ &= \left\|f_i(\mathbf{x}_{i,t})\right\|_2^2. \end{aligned} \tag{7.18}$$

Next, we consider the case $d < k$, in which $\mathbf{P}$ consists of d orthogonal rows. Let us write

$f_i(\mathbf{x}_{i,t}) = |f|\mathbf{f}_u$, where $\mathbf{f}_u$ is the unit vector pointing to the same direction as $f_i(\mathbf{x}_{i,t})$. Hence

$$\|\mathbf{P}f_i(\mathbf{x}_{i,t})\|_2^2 = |f|^2 \sum_{i=1}^{d} \langle \mathbf{P}_i, f_u \rangle^2 \leq |f|^2 = \|f_i(\mathbf{x}_{i,t})\|_2^2 . \tag{7.19}$$

where $\mathbf{P}_i$ is the i^{th} row of $\mathbf{P}$.

Finally, by Eq. (7.6), the maximum value of $\|f_i(\mathbf{x}_{i,t})\|_2^2$ is attained when $\tilde{f}_i(\mathbf{x}_{i,t})$ is exactly 1 at one coordinate and all zeros in the other coordinates. Hence,

$$\|f_i(\mathbf{x}_{i,t})\|_2^2 \leq \frac{1}{k^2}(k-1) + \left(1 - \frac{1}{k}\right)^2 = 1 - \frac{1}{k}. \tag{7.20}$$

Hence, for each client, we must satisfy

$$\frac{\gamma^2}{|\mathcal{P}_t|^2}\mu_{1/h}\left(1 - \frac{1}{k} + d\sigma_{\text{client}}^2\right) \leq P, \tag{7.21}$$

which leads to

$$\gamma = |\mathcal{P}_t|\sqrt{\frac{P}{\mu_{1/h}\left(1 - \frac{1}{k} + d\sigma_{\text{client}}^2\right)}}. \tag{7.22}$$

7.4 Privacy Analysis

In this section, we provide the privacy analysis of the proposed over-the-air ensemble scheme. Note that, as discussed in Section 7.3.2, while calculating the DP guarantees, the channel noise $\mathbf{n}_t$ is ignored. Thus, our guarantees are upper bounds and the actual privacy guarantees can be stronger due to the channel noise.

We first analyze the case in which all the clients participate.

Theorem 1. *If all the clients participate in the inference, i.e., $p = 1$, then, Algorithm 9 is (ε, δ)-DP such that for any $\varepsilon > 0$,*

$$\delta = \Phi(1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}) - e^{\varepsilon}\Phi(-1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}), \tag{7.23}$$

where Φ is the cumulative distribution function of standard normal distribution.

Proof. Our theorem is a special case of the following lemma.

Lemma 1 (Theorem 8 in [179]). *Let $f : \mathbb{L} \to \mathbb{R}^k$ be a function with $||f(\mathcal{L}) - f(\mathcal{L}')||_2 \leq C$, where $\mathcal{L}$ and $\mathcal{L}'$ are neighbouring inputs and $||\cdot||_2$ is L_2 norm. A mechanism $M(\mathcal{L}) = f(\mathcal{L}) + \mathcal{N}(0, \sigma^2 \mathbf{I}_k)$ is (ε, δ)-DP if and only if*

$$\Phi\left(C/(2\sigma) - \varepsilon\sigma/C\right) - e^{\varepsilon}\Phi\left(-C/(2\sigma) - \varepsilon\sigma/C\right) \leq \delta. \tag{7.24}$$

To apply Lemma 1 directly in our case, first observe that γ and $\mathbf{P}$ are common to all the clients. Hence, the effective noise is directly added to $\tilde{\mathbf{z}}_t \triangleq \sum_{i \in \mathcal{P}_t} f_i(\mathbf{x}_{i,t})$, and we need to calculate the L_2 sensitivity, C, of $\tilde{\mathbf{z}}_t$. Note that we can also ignore the operation in Eq. (7.6) since it is only a shift operation that does not change the sensitivity calculations. Consider neighbouring sets $\mathcal{L}$ and $\mathcal{L}'$. We denote the noiseless vector received by the CIS by $\tilde{\mathbf{z}}_t$ when the set of local models is $\mathcal{L}$, and by $\tilde{\mathbf{z}}'_t$ when it is $\mathcal{L}'$. Then,

$$C = \max_{\tilde{\mathbf{z}}_t, \tilde{\mathbf{z}}'_t} \|\tilde{\mathbf{z}}_t - \tilde{\mathbf{z}}'_t\|_2 = \max_{\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}'_t} \left(\sum_{j=1}^{k} (\tilde{\mathbf{z}}_{t,j} - \tilde{\mathbf{z}}'_{t,j})^2 \right)^{1/2}. \tag{7.25}$$

We know that $\left(f_i(\mathbf{x}_{i,t})\right)_j \in [0, 1], \forall j \in [k]$. Hence, $\|\tilde{\mathbf{z}}_t - \tilde{\mathbf{z}}'_t\|_2$ is maximized when $\tilde{\mathbf{z}}_t - \tilde{\mathbf{z}}'_t$ has only two non-zero elements: one is 1 and the other is -1. Then, $C = \max_{\tilde{\mathbf{z}}_t, \tilde{\mathbf{z}}'_t} \|\tilde{\mathbf{z}}_t - \tilde{\mathbf{z}}'_t\|_2 = \sqrt{2}$.

Finally, by substituting $C = \sqrt{2}$ into Eq. (7.24), we obtain Eq. (7.23). □

Next, we present the amplification effect of client sampling on the privacy guarantees.

Theorem 2. *If each client independently participate in inference with probability $p < 1$, then Algorithm 9 is (ε', δ')-DP, where, for any $\varepsilon' > 0$,*

$$\delta' = \frac{p}{1 - (1-p)^n} \Big(\Phi(1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}) - e^{\varepsilon}\Phi(-1/(\sqrt{2}\sigma) - \varepsilon\sigma/\sqrt{2}) \Big), \tag{7.26}$$

where $\varepsilon = \log\left(1 + ((1 - (1-p)^n)/p)(e^{\varepsilon'} - 1)\right)$.

Proof. Without loss of generality, let $\mathcal{L}$ and $\mathcal{L}'$ be two neighbouring sets of models differing only in the first client's model, i.e., it is either f_1 or f_1'. Let us write the output distribution of Algorithm 9 as a mixture distribution. When the model set is $\mathcal{L}$, we have $\mu = (1-\eta)\mu_0 + \eta\mu_1$ and when the model set is $\mathcal{L}'$, we have $\mu' = (1-\eta)\mu_0 + \eta\mu_1'$. In these expressions, η is the probability that client 1 is sampled, μ_0 is the output distribution when client 1 is not sampled, μ_1 is the output distribution when client 1 is sampled and the model set is $\mathcal{L}$, while μ_1' is the output distribution when client 1 is sampled and the model set is $\mathcal{L}'$. Recall that we sample client models each with probability p from $\mathcal{L}$ or $\mathcal{L}'$, and the CIS receives non-zero vectors only when $|\mathcal{P}_t| > 0$. Hence, $\eta = \Pr\{\text{Client 1 is sampled} \mid |\mathcal{P}_t| > 0\}$, resulting in $\eta = p/(1-(1-p)^n)$ via Bayes' rule.

Lemma 2 (Theorem 1 in [180])**.** *A mechanism $\mathcal{M}$ is (ε', δ')-DP if and only if*

$$\sup_{\mathcal{L},\mathcal{L}'} D_\alpha(\mathcal{M}(\mathcal{L})||\mathcal{M}(\mathcal{L}')) \leq \delta', \tag{7.27}$$

where $\alpha = e^{\varepsilon'}$ and $D_\alpha(\mu||\mu') \triangleq \int_Z \max\{0, d\mu(z) - \alpha d\mu'(z)\}d(z)$.

Lemma 2 implies that it is enough to bound $D_\alpha(\mu||\mu')$ to provide DP guarantees. For this, we use the relation in Lemma 3, which is called *advanced joint convexity* of D_α.

Lemma 3 (Theorem 2 in [180])**.** *For $\alpha \geq 1$, we have*

$$D_{\alpha'}(\mu||\mu') = \eta D_\alpha(\mu_1||(1-\beta)\mu_0 + \beta\mu_1') \tag{7.28}$$

where $\alpha' = 1 + \eta(\alpha - 1)$ and $\beta = \alpha'/\alpha$.

We further upper bound Eq. (7.28) via convexity:

$$D_{\alpha'}(\mu||\mu') \leq \eta(1-\beta)D_\alpha(\mu_1||\mu_0) + \eta\beta D_\alpha(\mu_1||\mu_1'). \tag{7.29}$$

To bound $D_\alpha(\mu_1||\mu_0)$, observe that there exists a coupling between μ_1 and μ_0 as follows. For μ_0, to guarantee $|\mathcal{P}_t| > 0$, let us first sample exactly one client c other than client 1 since we

know that client 1 is not sampled. Then, we apply Poisson sampling on the remaining set, i.e., $[n] \setminus \{c, 1\}$, to determine the other participating clients. For μ_1, assume that we have the same realization of Poisson sampling on $[n] \setminus \{c, 1\}$ as in μ_0. Further, by definition of μ_1, client 1 is also sampled. Hence, μ_1 and μ_0 can be seen as output distributions of Algorithm 9 such that the input client sets differ in only one element. Hence, $D_\alpha(\mu_1||\mu_0) \leq \delta$ due to Theorem 1. Similarly, to bound $D_\alpha(\mu_1, \mu_1')$, a coupling exists between μ_1 and μ_1' such that user 1 is sampled and f_1 and f_1' are the models in user 1, for μ_1 and μ_1', respectively. To determine the other participating clients, the same realization of Poisson sampling on $[n] \setminus \{1\}$ is applied in both μ_1 and μ_1'. Since the input client sets also differ in one element, in this case, due to Theorem 1, we have $D_\alpha(\mu_1, \mu_1') \leq \delta$. If we insert the bounds on $D_\alpha(\mu_1, \mu_0)$ and $D_\alpha(\mu_1, \mu_1')$ into Eq. (7.29), we obtain $D_{\alpha'}(\mu||\mu') \leq \eta\delta$, from which Eq. (7.26) follows. The expression for ε can be directly derived from the expression $\alpha' = 1 + \eta(\alpha - 1)$. □

7.5 Numerical Results

In this section, we present our experimental setup, implementation details and simulation results.

7.5.1 Datasets

We employ eight different multi-class classification datasets to demonstrate the effectiveness of our framework: CIFAR-10, CIFAR-100, FashionMNIST, Food101, OxfordPets, Emotion, Imdb, and MultiviewPets. CIFAR-10 is a multi-class image classification dataset containing 50 000 training images, 10 000 test images, and 10 target classes [156]. CIFAR-100 is the same dataset with finer granularity, i.e., it is partitioned into 100 target classes [156]. FashionMNIST is a multi-class fashion image classification dataset with 60 000 training images, 10 000 test images, and 10 target classes [181]. Food101 is a dish classification dataset with 75 750 training and 25 250 test images, and 101 target classes [182]. OxfordPets (Oxford-IIIT Pet) is a multi-class pet classification dataset with 37 target classes, 3680 training and 3669 test images [183]. Emotion is an English Twitter sentiment classification dataset with 16 000 training and 2000 test texts,

and 6 target classes [184]. Imdb is a binary sentiment classification dataset with 25 000 training texts, and 25 000 test texts [185].

For all the datasets, we use predefined training and testing sets, except that we split 10% of the training set as the validation set and only use the remaining 90% for training. Note that the training splits are further split disjointly across clients as described in Section 7.2 except the multi-view datasets. We have generated MultiViewPets datasets using OxfordPets images and labels following the common practice of simulating different camera angles. Instead of splitting the dataset disjointly, we crop regions of the same images and assign a region to each client. We crop 20 regions (via 4 vertical and 5 horizontal lines) with 50% overlap between neighbouring regions.

Remark 11. *Throughout the chapter, we assume that our privacy guarantees are local-model-level, which implies the same DP guarantees for the local datasets. However, in the case of multi-view classification on the MultiViewPets dataset, this is no longer valid since some parts of the data samples located in a client also exist in other clients due to the nature of the multi-view setting. This results in weaker DP guarantees at the local dataset level. Nevertheless, our local-model-level privacy guarantee and all of its benefits still hold.*

7.5.2 Experimental Setup

We repeat all the experiments with 5 different random seeds and report the average results. We randomly split the training data among the clients equally. We consider $n = 20$ clients with a participation probability of $p = 1.0$ and a channel SNR of 0 dB, except when they are changed gradually in Section 7.5.5.

We compute and report macro-averaged F1 (Macro-F1) scores by averaging per-class F1 scores, which is a commonly employed metric in multi-class classification since it treats each class with equal significance, ensuring a balanced evaluation across all classes.

7.5.3 Implementation Details

We use PyTorch [129], torchvision for vision tasks and Huggingface transformers for text tasks [186, 187]. For image datasets, we use MobileNetV3-Large [188] except we change its final layer to make it compatible with the target number of classes. Instead of training from scratch, we fine-tune a pre-trained model [186] for 50 epochs. Similarly, for text datasets, we fine-tune the DistilBERT-base-uncased model [189] for 50 epochs. To make the sizes of the images compatible with our MobileNetV3 network, we interpolate them to 224×224 images following PyTorch's pretraining recipe. Since the network accepts inputs with three channels, we convert grayscale datasets by repeating the grayscale image for all RGB channels. We employ stochastic gradient descent (SGD) optimizer with a batch size of 128, momentum 0.9, weight decay 5×10^{-4}, learning rate 0.05 and cosine annealing scheduling method [190]. We employ cross-entropy loss for training all local models. We also utilize label smoothing factor 0.1 for training vision models to better calibrate the beliefs [191].

7.5.4 Comparison with the Baselines

In Table 7.1, we compare the proposed OAC-based methods with the *Best Client* model and the ensemble methods with orthogonal transmission in terms of their Macro-F1 scores. We choose the model with the highest Macro-F1 score on the same validation set as the *Best Client* model. For fairness, the client having the best model transmits its inference over k channels. In orthogonal methods, denoted by *-Orth* suffix, all the devices transmit their inferences via different channels, i.e., $|\mathcal{P}_t| \times k$ channels in total.

We observe that, compared to the *Best Client* model, ensemble methods significantly improve the test scores, especially in the private setting. Moreover, while orthogonal and OAC-based methods perform competitively in the non-private setting, when privacy is involved, the *Best Client* model and orthogonal methods perform near-random, and significantly worse than the OAC-based methods. Note that orthogonal methods need $|\mathcal{P}_t| \times k$ channel uses, whereas OAC-based methods need only k channel uses; yet, OAC-based methods outperform orthogonal

Table 7.1: Comparison of our method with the baselines in terms of Macro-F1 scores

ε	Method	CIFAR-10	CIFAR-100	FashionMnist	Food101	OxfordPets	Emotion	Imdb	MultiViewPets
	Best Client	83.16±0.29	59.49±0.67	84.24±0.14	58.42±0.23	58.70±2.07	71.03±1.87	82.90±0.19	83.33±0.41
	BA-Orth	91.57±0.04	73.02±0.35	91.87±0.07	71.25±0.12	82.87±0.57	83.80±0.52	90.86±0.07	87.46±0.24
	WBA-Orth	91.56±0.04	73.02±0.34	91.86±0.08	71.24±0.12	82.88±0.64	83.84±0.50	90.86±0.07	87.43±0.23
∞	MV-Orth	91.55±0.09	72.12±0.26	91.84±0.04	70.75±0.17	82.03±0.70	83.69±0.58	90.83±0.06	86.58±0.22
	BA-OAC	91.79±0.10	**73.25±0.36**	**91.99±0.12**	**71.74±0.10**	**83.55±0.61**	83.85±0.63	**91.00±0.05**	**87.77±0.26**
	WBA-OAC	**91.80±0.10**	73.24±0.35	**91.99±0.12**	71.72±0.09	83.51±0.63	**83.86±0.64**	**91.00±0.05**	**87.77±0.24**
	MV-OAC	91.66±0.06	72.04±0.28	91.88±0.08	70.80±0.14	82.33±0.54	83.60±0.51	90.99±0.06	86.79±0.20
	Best Client	18.37±0.13	2.30±0.10	18.84±0.16	2.35±0.23	5.42±0.24	22.99±0.79	60.68±0.16	6.40±0.20
	BA-Orth	53.36±0.34	6.65±0.19	57.38±0.33	6.55±0.25	9.17±0.73	60.17±1.18	84.64±0.20	19.35±0.35
	WBA-Orth	53.36±0.33	6.66±0.20	57.36±0.35	6.55±0.25	9.20±0.76	60.16±1.16	84.64±0.20	19.32±0.37
5	MV-Orth	65.08±0.46	19.17±0.25	66.32±0.57	19.61±0.38	25.16±0.69	60.16±1.32	84.62±0.21	35.59±0.35
	BA-OAC	91.40±0.15	56.68±0.24	**91.89±0.13**	50.85±0.26	65.89±1.47	83.69±0.41	90.96±0.07	83.22±0.20
	WBA-OAC	91.39±0.14	56.67±0.24	91.87±0.11	50.84±0.27	65.90±1.52	**83.83±0.40**	90.96±0.07	83.21±0.20
	MV-OAC	**91.60±0.07**	**71.21±0.21**	91.80±0.07	**69.98±0.08**	**81.17±0.77**	83.51±0.86	**90.97±0.05**	**86.30±0.21**
	Best Client	11.55±0.12	1.21±0.13	11.71±0.23	1.26±0.13	3.21±0.20	16.42±0.99	52.65±0.18	3.38±0.18
	BA-Orth	17.21±0.26	1.56±0.11	17.81±0.15	1.52±0.13	3.54±0.30	23.55±1.24	61.15±0.16	4.34±0.22
	WBA-Orth	17.21±0.27	1.56±0.11	17.81±0.16	1.52±0.13	3.56±0.30	23.55±1.24	61.15±0.16	4.34±0.22
1	MV-Orth	19.31±0.14	2.22±0.19	19.44±0.16	2.26±0.11	4.92±0.35	23.60±1.23	61.14±0.18	5.76±0.22
	BA-OAC	71.14±0.37	12.72±0.26	76.07±0.36	12.14±0.15	13.80±0.99	**74.80±1.08**	89.14±0.13	32.39±0.95
	WBA-OAC	71.14±0.37	12.72±0.26	76.04±0.37	12.14±0.15	13.83±0.99	74.78±1.05	89.14±0.13	32.39±0.93
	MV-OAC	**82.43±0.33**	**36.24±0.58**	**83.73±0.19**	**36.66±0.19**	**40.95±1.60**	74.66±0.94	**89.19±0.12**	**56.40±0.37**

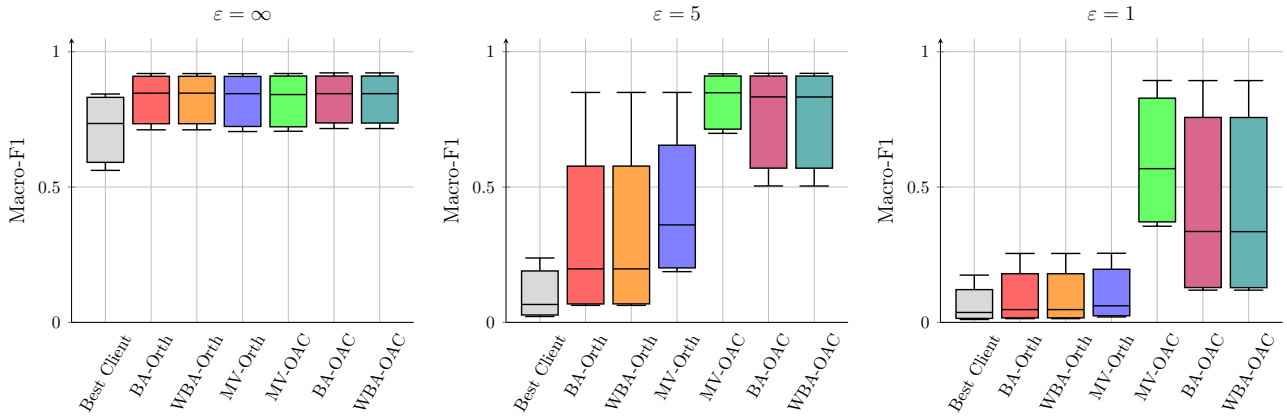


Figure 7.3: Box plots for the Macro-F1 score for all the datasets in the case of non-private ($\varepsilon = \infty$, left), moderately private, ($\varepsilon = 5$, middle), strongly private ($\varepsilon = 1$, right) scenarios.

ones in the private setting. In all the privacy settings, i.e., $\varepsilon \in \{1, 5, \infty\}$, *Weighted Belief Averaging* and standard (unweighted) *Belief Averaging* methods perform very close for both orthogonal and OAC-based variants, so we refer to both *Belief Averaging* and *Weighted Belief Averaging* via *Belief Averaging* term in the rest of this subsection.

Previous studies suggest that ensembling via *Belief Averaging* generally performs better than *Majority Voting* [192, 193]. Our non-private results also support this argument as beliefs contain more information compared to conveying local decisions. However, interestingly, when $\varepsilon = 1$

or $\varepsilon = 5$, *Majority Voting* outperforms *Belief Averaging* for both orthogonal and OAC-based settings. This can be explained by the fact that the increasing noise levels result in relatively unreliable beliefs, since the individual values of beliefs are smaller, and thus more sensitive to the noise added for privacy.

There is no significant winner between the *Majority Voting* and *Belief Averaging* schemes, especially in the non-private case, for both orthogonal and OAC cases. This situation contradicts the fact that averaging, which is equivalent to our *Belief Averaging*, is the clear winner in previous studies [192, 193], in which there is no noise added to the predictions due to privacy or wireless channel. This is probably caused by the channel and privacy noise terms, which result in the situation that more precise inferences per model result in a better ensemble. This is also supported by our results, which show that as noise increases, the relative performance of *Majority Voting* increases compared to *Belief Averaging* counterparts.

Moreover, as the number of classes decreases, the difference between *Belief Averaging* and *Majority Voting* decreases, which is probably due to the spread of the beliefs among classes. This can be observed through the difference between *Belief Averaging* and *Majority Voting* experiments on Imdb (2 classes), Emotion (6 classes), CIFAR-10 (10 classes) and CIFAR-100 (100 classes) datasets.

In the multi-view setting on the MultiViewPets dataset, the methods behave similarly in terms of Macro-F1 score compared to the ensemble setting. OAC-based methods outperform all of the orthogonal methods and *Best Client* method. In the non-private setting, *Belief Averaging*-based methods outperform *Majority Voting* while in weak-private and private settings *Majority Voting* outperforms *Belief Averaging*-based methods. Similar to the ensemble setting, *Best Client* performs the worst among all the methods in the same privacy setting.

Figure 7.3 shows the boxplots of the evaluated methods over all the datasets for different privacy levels. In addition to the abovementioned points, as shown in the box plots, we observe an increase in the standard deviations of all the methods over different datasets as we decrease ε. This increase is caused by the privacy noise added as we decrease ε. These box plots also validate our above-mentioned claims.

7.5.5 Comparison with Varying Conditions

Figures 7.4 to 7.6 show the performance on the CIFAR-10 dataset for varying channel SNR, p, d and ε values of *Majority Voting*, *Belief Averaging* and *Weighted Belief Averaging* fusion methods, respectively.

These three fusion methods perform similarly with respect to the evaluated channel SNR, p and d. The left figures show that the performance of the methods slightly increases with the channel SNR, especially for SNR values below 2 dB. In the middle figures, we observe that higher p improves the performance significantly in the private settings. Although lower p has a privacy amplification effect which decreases the additional noise variance required to attain the desired privacy level, we observe that its privacy amplification effect is not as significant as the impact of fewer client participation on the inference performance. In the non-private setting ($\varepsilon = \infty$), having higher participation also helps to get a higher macro-F1 score, but not as much as in the private setting. These plots also show that the private setting is more sensitive to varying conditions for both p and channel SNR. The right hand side figures clearly demonstrate that the result improves as the projection dimension d increases until $d = k$ and reaches its peak at $d = k$. Further increase in d does not benefit the result since we consider average power constraint in this chapter, which keeps the noise ratio fixed with respect to the signal.

7.5.6 Ablation Study of Privacy and Projection Methods

Here, we perform an ablation study evaluating different projection methods for private and non-private cases, which is presented in Table 7.2. We employ validation set to prevent leakage from the test set by exhaustive search for hyperparameters.

Again, OAC-based methods outperform their orthogonal counterparts in all our experiments. Both Gaussian and Rademacher projections are among the most common in the compressed sensing literature. In Gaussian projection, we sample $\mathbf{P}$ from random normal distribution, whereas in Rademacher projection, we sample $\mathbf{P}$ from Rademacher distribution that consists of -1 and 1 with equal probability. Orthogonal projection clearly outperforms both Gaussian

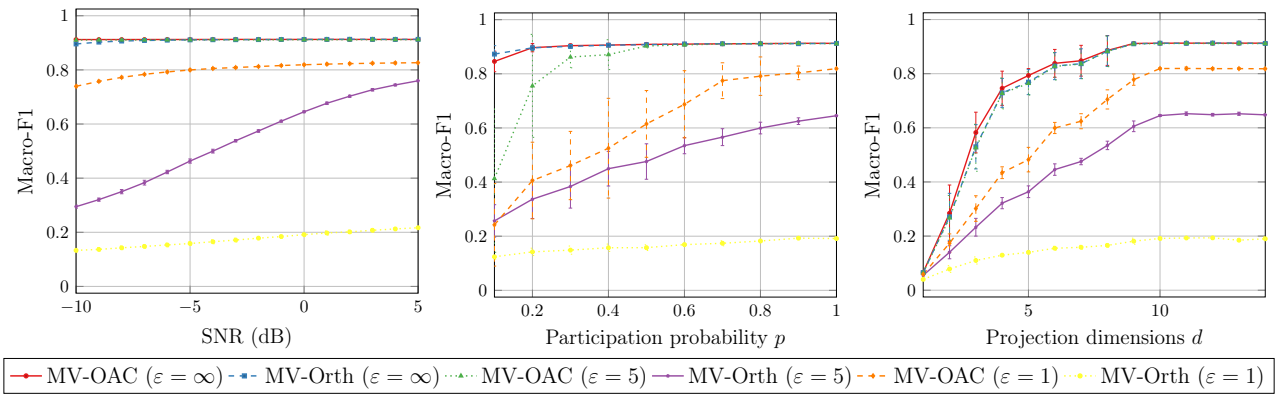


Figure 7.4: Comparison of MV-OAC and MV-Orth with different DP levels as a function of channel SNR (left), participation probability p (middle) and projection dimensions d (right) on the validation split of CIFAR-10 dataset.

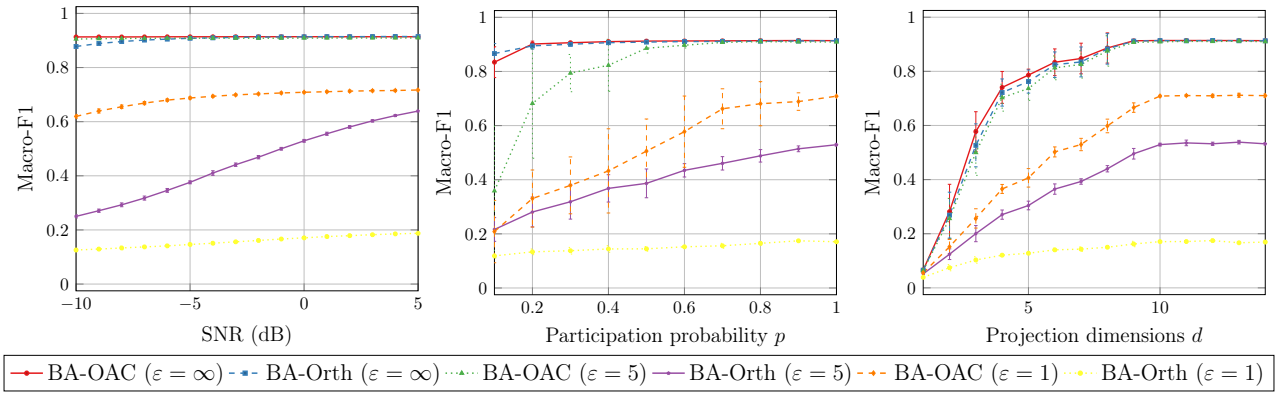


Figure 7.5: Comparison of BA-OAC and BA-Orth with different DP levels as a function of channel SNR (left), participation probability p (middle) and projection dimensions d (right) on the validation split of CIFAR-10 dataset.

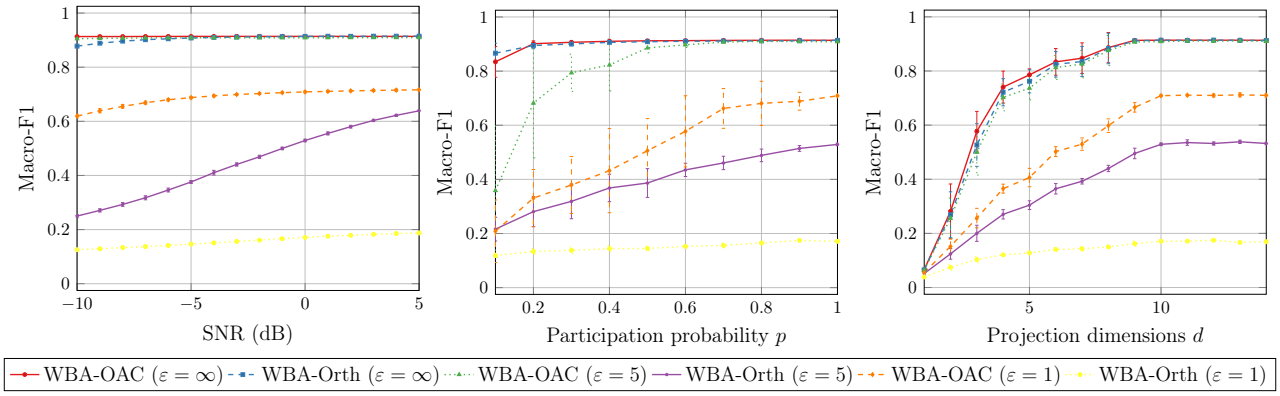


Figure 7.6: Comparison of WBA-OAC and WBA-Orth with different DP levels as a function of channel SNR (left), participation probability p (middle) and projection dimensions d (right) on the validation split of CIFAR-10 dataset.

and Rademacher projections for all the evaluated ε values, and performs very closely to identity projection, while having the additional benefit of expanding/shrinking the bandwidth.

In the private setting, we evaluate the projection of private labels, in which we add the privacy noise to the beliefs $f_i(\mathbf{x}_{i,t})$, i.e., before projection. These results are denoted by appending "of labels" to the method name, and they perform better than the private projection methods that add privacy noise after projection, which are described in [194]. This is because DP accounts for privacy noise in the worst-case, so although projection has some randomization and distortion effect on the signal, it still tries to protect that part, and this results in worse performance.

Among all these results, orthogonal projection of labels, which is described in Section 7.3.3, performs the best, and is very close to the identity projection.

As a fully digital baseline, in Table 7.2, we also report the results of randomized response (RR) [195] as a private inference method. Since RR is a discrete scheme, we only employ it for *Majority Voting*. In this technique, each client conveys its true prediction with probability $\frac{e^{\varepsilon}}{e^{\varepsilon}+k}$ and tells otherwise a lie by uniform randomly picking a wrong label. For *MV-OAC*, we further benefit from shuffling although its effect is limited for $n = 20$ clients [196]. We note that the MAC acts as a shuffler since signals lose the order via summation. Since RR works for digital schemes, and it does not act as additive noise, the summation of randomized predictions from the clients does not improve the overall privacy even if we use OAC.

Table 7.2: Ablation study of different projection methods on the validation split of CIFAR-10

ε	Projection	MV-Orth (%)	MV-OAC (%)
∞	Identity Projection	91.15±0.20	**91.33±0.17**
	Gaussian Projection	68.36±22.99	87.15±8.78
	Rademacher Projection	34.17±39.50	39.83±46.29
	Orthogonal Projection	91.16±0.16	91.31±0.21
1	Identity Projection	19.07±0.94	**82.08±0.51**
	Gaussian Projection	8.41±1.89	20.03±7.84
	Gaussian Projection of Labels	9.84±2.82	58.01±18.89
	Rademacher Projection	7.15±2.81	13.05±11.16
	Rademacher Projection of Labels	7.66±3.69	29.33±32.54
	Orthogonal Projection	19.15±0.64	81.87±0.13
	Orthogonal Projection of Labels	19.13±0.58	81.91±0.31
	Orthogonal Projection of RR	43.70±0.98	52.81±0.52

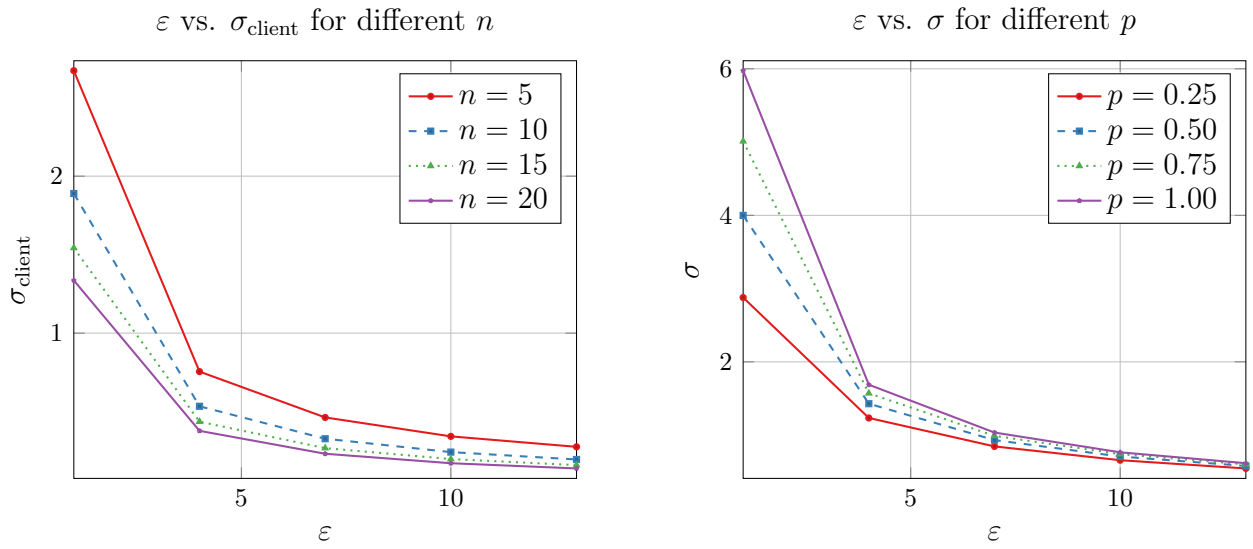


Figure 7.7: The standard deviation of noise for different system and privacy conditions.

7.5.7 Analysis of Privacy Quantities

In this section, we show the quantitative values of standard deviations for the DP noise to be added in different scenarios. Figure 7.7 illustrates the effect of the number of users n and participation probability p on the privacy noise. The amount of privacy noise, measured by its variance, decreases as ε increases. This is expected because higher ε corresponds to lower privacy, which requires less noise to be added. As participation probability p increases, we need to add more noise in total to guarantee the same level of DP since partial participation is another source of randomness. As the number of clients n increases, we need to add less noise per client to guarantee the same level of DP since noise is distributed among the participating clients.

7.5.8 Analysis of Scalability

Table 7.3 demonstrates the scalability of the proposed method on CIFAR-10 dataset for various number of clients. As the number of clients increases, the performance of all methods tends to decline, primarily due to the assumption that the dataset is split into non-overlapping portions among different clients. This leads to less diverse data at each user, reducing the effectiveness of individual models in capturing the overall data distribution. The *Best Client* method, which

Table 7.3: Scalability analysis of the introduced methods on CIFAR-10 in terms of Macro-F1

ε	Method	5 Users	15 Users	20 Users	50 Users	100 Users
∞	Best Client	86.35±0.09	83.97±0.44	83.16±0.29	68.85±1.28	71.59±1.01
	BA-Orth	93.37±0.09	92.13±0.14	91.57±0.04	83.02±0.27	83.89±0.27
	WBA-Orth	93.37±0.11	92.13±0.14	91.56±0.04	83.06±0.27	83.98±0.24
	MV-Orth	93.30±0.09	92.16±0.10	91.55±0.09	82.61±0.42	83.73±0.21
	BA-OAC	**93.80±0.10**	92.32±0.09	91.79±0.10	83.29±0.23	83.95±0.19
	WBA-OAC	**93.80±0.11**	92.31±0.07	91.80±0.10	83.28±0.22	84.03±0.17
	MV-OAC	93.62±0.06	92.28±0.11	91.66±0.06	82.92±0.26	83.74±0.17
5	Best Client	19.20±0.28	19.24±0.62	18.37±0.13	16.86±0.64	17.09±0.36
	BA-Orth	31.00±0.48	47.93±0.52	53.36±0.34	57.37±0.43	64.12±0.85
	WBA-Orth	30.99±0.49	47.92±0.50	53.36±0.33	57.37±0.45	64.16±0.85
	MV-Orth	35.24±0.54	57.77±0.58	65.08±0.46	67.45±0.59	79.13±0.50
	BA-OAC	78.91±0.47	91.74±0.16	91.40±0.15	83.07±0.21	83.92±0.20
	WBA-OAC	78.91±0.45	91.75±0.15	91.39±0.14	83.09±0.20	84.03±0.21
	MV-OAC	85.36±0.22	**92.14±0.11**	91.60±0.07	82.82±0.26	83.74±0.15
1	Best Client	11.97±0.16	11.93±0.43	11.55±0.12	11.32±0.23	11.27±0.29
	BA-Orth	13.78±0.30	16.18±0.19	17.21±0.26	18.25±0.39	20.86±0.37
	WBA-Orth	13.78±0.30	16.18±0.18	17.21±0.27	18.25±0.39	20.86±0.36
	MV-Orth	14.44±0.31	17.63±0.06	19.31±0.14	21.12±0.40	29.36±0.28
	BA-OAC	22.46±0.38	58.45±0.64	71.14±0.37	78.34±0.45	82.34±0.39
	WBA-OAC	22.47±0.37	58.47±0.65	71.14±0.37	78.34±0.40	82.41±0.39
	MV-OAC	25.20±0.27	69.74±0.51	82.43±0.33	81.05±0.45	**83.52±0.22**

relies on the model of a single user, performs well initially but experiences a sharp decline in Macro-F1 scores, particularly as privacy constraints are introduced. For instance, at $\varepsilon = 5$, its performance drops from 19.20 with 5 clients to 17.09 with 100 clients, and similarly at $\varepsilon = 1$, from 11.97 to 11.27. This demonstrates the method's limited scalability, especially in scenarios where privacy is essential and the number of clients grows.

In contrast, OAC Methods–*BA-OAC*, *WBA-OAC*, and *MV-OAC*–exhibit better scalability, especially under privacy constraints. These methods leverage the superposition property of MACs, allowing them to efficiently aggregate information from users without needing orthogonal communication channels. Even with 100 users and at the strictest privacy level ($\varepsilon = 1$), *MV-OAC* still maintains a high Macro-F1 score of 83.52. This makes OAC methods more resilient to increasing numbers of users and privacy requirements compared to orthogonal methods, which experience more significant performance drops under the same conditions. Overall, OAC-based methods provide a more scalable and privacy-efficient solution for distributed inference in

large-scale, privacy-sensitive environments.

7.5.9 Statistical Significance of the Results

In this section, we demonstrate the statistical significance of the results using a non-parametric test.

We first perform Friedman test and obtain p-value less than 0.05, which means that distributions are significantly different. We then proceed with the post-hoc Nemenyi test at 0.05 significance level, in which we found the critical distance (CD) as 1.42, which is the minimum difference to have between two methods for statistical distinguishability. We can reject the null hypothesis for the models with an average rank larger than the CD, i.e., it means these models are drawn from significantly different distributions.

The top plot in Fig. 7.8 shows the CD diagram for the non-private case, $\varepsilon = \infty$. It shows that *BA-OAC* and *WBA-OAC* perform significantly better than all the other methods. The second and the third plots in Fig. 7.8 show the CD diagram for the moderately private ($\varepsilon = 5$) and strongly private ($\varepsilon = 1$) cases, respectively. In both, *MV-OAC* becomes the method with the highest rank due to the inclusion of privacy noise, and all the OAC-based methods perform significantly better than other alternatives. There is also no significant difference among orthogonal methods and among OAC-based methods. In all three scenarios, the *Best Client* method performs the poorest due to the performance improvement of ensembling.

7.5.10 Analog vs Digital Implementation

In this section, we comment on the possibility of adapting the introduced methods in Sections 7.5.4 and 7.5.6 to the digital setting.

In the non-private case, *Majority Voting* can be directly implemented digitally since we use a predefined code $\mathbf{P}$. Again, in the non-private case, *Belief Averaging*-based methods can be implemented by quantizing the beliefs, and *Weighted Belief Averaging*-based methods can

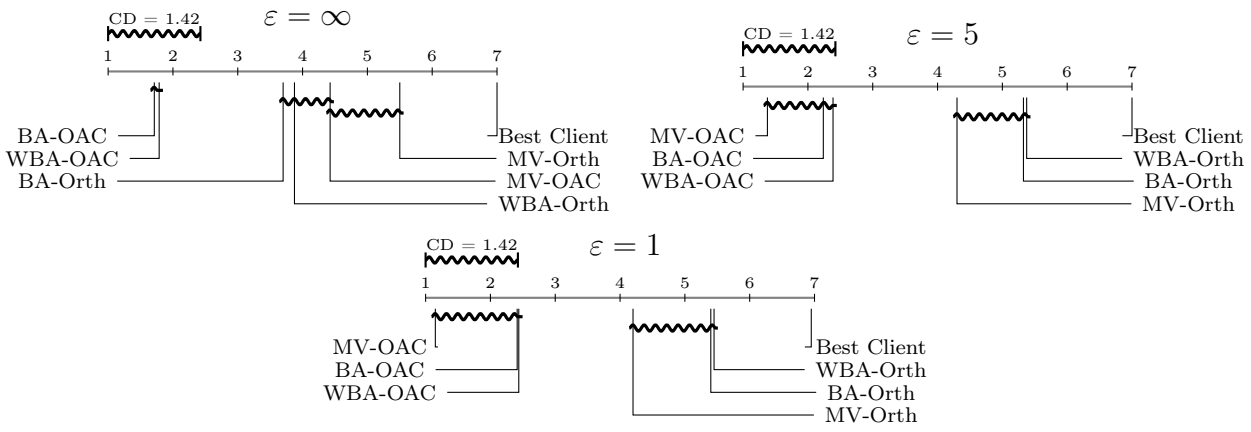


Figure 7.8: Comparison of the methods for different privacy levels ($\varepsilon = \infty, \varepsilon = 5, \varepsilon = 1$), respectively, in terms of average rank with the post-hoc Nemenyi test using a critical difference diagram. Connected methods are not significantly different at the 0.05 significance level.

be implemented after quantizing the weighted beliefs. These methods cannot be directly implemented in the cases with privacy noise, i.e., in the cases when $\varepsilon \in \{1, 5\}$. In the weak-private and private cases, *Orthogonal Projection of RR*, introduced in Section 7.5.6, can be implemented digitally since privacy noise is digital as well.

Remark 12. *For simplicity, we employ analog OAC in our system model and focus on the private collaborative inference task. To enable compatibility with digital communication systems, the digital OAC frameworks proposed in [197, 198, 199] provide a strong foundational basis. We leave this adaptation for future exploration and encourage readers to consult the survey papers [200, 167] for additional perspectives.*

7.5.11 Practical Considerations

Synchronization and Mobility: In practical deployments, achieving perfect time and frequency synchronization among a large number of devices is challenging due to factors like clock drift, network delays, hardware limitations, and device mobility [167]. As devices move during the data consensus process, changes in network topology and neighborhood relationships can occur, which may impact system performance by introducing additional asynchrony and communication delays. By incorporating participation probability into our method, we not only improve privacy guarantees but also model the impact of asynchrony and device mobility.

This approach provides a practical assessment of the proposed method's resilience to dynamic network conditions, offering insights into its effectiveness in real-world scenarios where perfect synchronization and static network topologies are unattainable.

Instantaneous channel state information (CSI): In this chapter, we assume each client has access to its instantaneous CSI (channel gain), enabling transmission power scaling for optimal signal aggregation at the inference server. However, acquiring real-time CSI can introduce delays and increase signaling overhead, especially in high-mobility or large-scale networks. To address this, recent research explores blind OAC methods that bypass per-user instantaneous CSI by leveraging statistical channel information or adaptive transmission strategies.

Blind OAC designs [201, 167] aim to reduce signaling overhead and latency by eliminating the need for real-time CSI exchange. Instead, these methods rely on techniques such as power control (adjusting transmission power based on long-term channel statistics or pre-configured heuristics), fixed modulation schemes (using predetermined modulation formats independent of real-time channel conditions), or statistical estimations based on historical channel data. While our current approach leverages CSI for accurate signal alignment, blind OAC strategies offer a compelling solution for dense or high-mobility networks, where signaling delays can severely impact performance. Exploring how to adapt our method to incorporate blind OAC principles is an exciting direction for future research.

Energy Efficiency: Our framework includes a channel gain threshold for optimizing transmission power, ensuring that clients in poor channel conditions reduce their energy expenditure. However, as the probability of non-participation due to poor channel gains is extremely low, the threshold has minimal impact on overall client participation. This design effectively balances energy efficiency with fairness, allowing nearly all clients to participate equitably in the inference process without significantly affecting power constraints. Additionally, in our system model, we assume that $h_{i,t}$'s are i.i.d. in which case the participation and performance remains fair. Thus, our approach ensures sustainable operation across IoT networks, maintaining high client participation while minimizing energy usage. For a deeper exploration on energy efficiency, we refer the reader to [202].

Common Randomness: In our method, clients use a pseudorandom seed to determine participation based on a shared common randomness, eliminating the need for inter-client communication. This setup is particularly beneficial in preserving privacy as it reduces communication overhead and potential exposure of client status or model information. However, we recognize that under non-ideal conditions, achieving exact synchronization is challenging. To mitigate this, our model incorporates a probabilistic participation parameter p that approximates asynchronous conditions by probabilistically excluding clients from inference rounds. This probabilistic exclusion serves as a proxy for potential synchronization errors, thus modeling the impact of communication delays or asynchrony that may prevent clients from transmitting in real-time. These approaches, combined with our probabilistic participation model, enhance our framework's adaptability to dynamic network conditions, such as mobility and varying signal quality, ensuring robustness even when perfect synchronization cannot be achieved.

7.6 Conclusion

We have introduced a private collaborative inference framework at the wireless edge. We have exploited OAC for bandwidth-efficient and accurate multi-class classification for multiple views while preserving the privacy of the local models and their data. We have provided DP guarantees exploiting both distributed noise addition after projection and random participation. We have systematically evaluated the introduced methods with OAC and shown that distributed edge inference with OAC performs significantly better than its orthogonal counterpart while using fewer resources. We have observed that while transmitting local beliefs from each client is more informative, enforcing and transmitting hard local decisions can be more reliable when noise is introduced to guarantee privacy.

Chapter 8

Conclusion

This thesis has investigated multi-user semantic communication systems, presenting frameworks for perceptual quality optimization, distributed source coding with deep learning, non-orthogonal semantic communications, and privacy-preserving collaborative inference. Through systematic analysis and experimental simulations, we have demonstrated that the integration of machine learning with wireless communications can provide efficient, private, and semantically aware information exchange.

8.1 Summary of Contributions

This thesis has presented contributions to wireless communications, distributed machine learning, and information theory, establishing frameworks for semantic communication systems that address efficiency, privacy, and performance requirements. The technical chapters are organized along a progression of increasing multi-user complexity, starting from single-user transmission and progressing to large-scale collaborative inference.

Chapter 3 began with the single-user point-to-point setting and introduced perceptual quality optimization by integrating denoising diffusion models into wireless image transmission. This work demonstrated that prioritizing perceptual quality over traditional distortion metrics can

lead to improved user experiences, establishing trade-offs between perceptual quality and traditional metrics while achieving semantically consistent reconstructions under challenging channel conditions.

Chapter 4 extended to a distributed setting, bridging classical information theory with modern deep learning by extending the Wyner-Ziv problem to neural network-based communication systems. Our framework demonstrated how decoder-only side information can be exploited in practical distributed compression scenarios, with adaptive mechanisms that dynamically adjust compression strategies.

Chapters 5 and 6 moved to simultaneous multi-user transmission, investigating non-orthogonal approaches over shared channels. Chapter 5 introduced NOMA principles to DeepJSCC for two users, showing performance advantages over separation-based approaches in practical finite blocklength regimes. Chapter 6 scaled to many users (demonstrated with up to 64 in our experiments), developing learning-based interference management strategies that represent a shift from interference avoidance to strategic interference utilization, demonstrating that carefully designed interference patterns can enhance overall system performance.

Chapter 7 concluded the progression by addressing large-scale collaborative inference, introducing privacy-preserving methods leveraging wireless channel properties for natural privacy amplification. Our OAC approach demonstrates that privacy and efficiency can be compatible objectives, achieving bandwidth savings while maintaining differential privacy guarantees through channel noise. The multi-view classification framework showed that collaborative inference benefits from diverse perspectives while maintaining privacy.

8.2 Limitations and Critical Discussion

While this thesis has demonstrated the potential of learned JSCC in multi-user settings, several limitations affect the practical relevance of the proposed methods and of the JSCC paradigm more broadly. This section provides an honest assessment of these limitations, situates the

contributions within the broader standardisation landscape, and identifies open problems that must be addressed before the techniques developed here can be deployed in real systems.

8.2.1 Channel Model Assumptions

All experimental evaluations in this thesis employ idealised channel models, namely AWGN and Rayleigh block-fading, with perfect CSI at both the transmitter and receiver. While these models are standard analytical benchmarks throughout the semantic communications literature, they omit many phenomena encountered in real propagation environments, including frequency-selective fading, time-varying channels with outdated CSI, inter-cell interference, hardware impairments (phase noise, non-linear amplification), and correlated multiple-input multiple-output (MIMO) channels.

The assumption of perfect CSI deserves particular scrutiny. In practice, channel estimation errors degrade the performance of learned encoders and decoders that are conditioned on SNR or channel coefficients. Although some of our architectures, such as the SNR-adaptive modules in Chapter 6, partially address robustness to channel variations, a systematic evaluation under estimated or outdated CSI has not been performed. Testing with more advanced channel models, such as standardised 3GPP multipath profiles, ray-tracing-based propagation, or measured channel data, remains necessary to substantiate the practical claims of this work.

8.2.2 JSCC as a Paradigm: Adoption Challenges

JSCC is not a novel concept and is information-theoretically motivated by the fact that Shannon's separation theorem is strictly optimal only in the limit of infinite blocklength. Despite this theoretical appeal, deep learning-based JSCC in particular has seen virtually no adoption in practical communication standards. Several fundamental challenges explain this gap.

Protocol stack integration. Modern communication systems are built around a rigorous layered architecture where source coding, channel coding, modulation, and medium access

are designed, standardised, and optimized independently. Adopting JSCC would require a substantial cross-layer redesign affecting physical layer processing, MAC-layer resource allocation, and higher-layer protocols. Existing standards such as 4G LTE and 5G NR represent decades of engineering investment in separate, highly optimized, and interoperable source and channel coding pipelines.

Rate and bandwidth adaptation. Conventional systems handle rate adaptation through well-defined mechanisms: modulation and coding scheme (MCS) selection, link adaptation via CQI feedback, and bandwidth part (BWP) allocation. In JSCC, the mapping from source to channel symbols is learned end-to-end, making it unclear how to dynamically allocate bandwidth or adjust the transmission rate in response to changing channel conditions without retraining or maintaining multiple models. Even if JSCC achieves higher spectral efficiency, the question of how to redistribute freed bandwidth among users or services remains open and requires new resource allocation frameworks.

Competing standards-compatible approaches. Deep learning-based source compression methods such as JPEG AI [203] and neural video codecs are being developed within existing standardisation frameworks, offering performance gains comparable to those of JSCC while remaining fully compatible with conventional channel coding and modulation pipelines. These approaches benefit from the mature engineering ecosystem of current wireless standards and may represent a more pragmatic path to deploying neural compression in wireless systems.

8.2.3 Evaluation Scope

Our experimental evaluations, while extensive in terms of the parameter space explored (number of users, SNR ranges, bandwidth ratios, dataset diversity), are conducted entirely through software simulation. No hardware prototyping or over-the-air experiments have been performed. Furthermore, the computational overhead of neural encoder–decoder architectures, including inference latency, energy consumption, and memory requirements on edge devices, has not been systematically characterised. These factors are critical for assessing the feasibility of deploying the proposed methods in resource-constrained wireless environments.

8.3 Impact and Future Directions

This thesis has implications for wireless communications and distributed machine learning, challenging certain assumptions in communication theory. Our demonstration that interference can be beneficial rather than purely detrimental opens new research directions in multi-user communication design. The approach to privacy preservation through natural channel properties provides alternatives to traditional cryptographic techniques, showing that privacy and efficiency can be complementary objectives in specific scenarios.

The move from syntactic to semantic communication represents a paradigm shift, aligning communication systems with their ultimate purpose in machine learning applications. This work bridges multiple disciplines including information theory, machine learning, wireless communications, and privacy, creating interdisciplinary research opportunities.

Future Research Directions include: (1) optimizing range-null space decomposition strategies for diffusion-enhanced DeepJSCC, investigating joint training objectives that balance perceptual quality metrics with traditional distortion measures across diverse channel conditions; (2) extending Wyner-Ziv deep learning architectures to handle imperfect or uncertain side information at the decoder, including robustness to time-varying correlation structures and theoretical analysis of achievable compression rates under practical side information mismatch; (3) developing adaptive interference management algorithms for NOMA-DeepJSCC systems that dynamically adjust user-specific projections based on channel conditions and data characteristics, with particular focus on synchronization mechanisms for large-scale deployments; (4) extending progressive fine-tuning frameworks to heterogeneous device capabilities, exploring optimal user grouping strategies and curriculum learning approaches for scaling beyond 64 users while maintaining parameter efficiency; (5) investigating the theoretical privacy-utility trade-offs in OAC-based collaborative inference, including precise differential privacy guarantees under correlated channel noise and optimal noise injection strategies for multi-view classification and ensembling; and (6) extending multi-user projection techniques and progressive fine-tuning algorithms to general multi-view and multi-task machine learning problems, exploring applications in FL, ensemble methods, and multi-modal learning scenarios beyond wireless communications.

In addition, several directions directly address the limitations discussed above: (7) evaluating all proposed architectures under realistic channel models, including standardised 3GPP multipath profiles, imperfect and outdated CSI, and hardware impairment models, to assess robustness beyond idealised AWGN and Rayleigh fading assumptions; (8) developing rate and bandwidth adaptation mechanisms for learned JSCC, including variable-rate neural codecs that can adjust the source-to-channel symbol mapping without retraining, and resource allocation frameworks that integrate JSCC into existing MAC-layer scheduling; (9) designing quality-aware feedback and retransmission protocols compatible with JSCC, replacing binary ACK/NACK with continuous quality indicators and incremental refinement strategies; (10) conducting hardware prototyping and over-the-air experiments using software-defined radio platforms to validate the proposed methods under real propagation conditions, characterise inference latency on edge devices, and identify practical bottlenecks; and (11) investigating hybrid architectures that combine learned JSCC components with standards-compatible codecs (e.g., JPEG AI with conventional channel coding), enabling incremental adoption within existing protocol stacks while retaining some benefits of end-to-end optimisation.

8.4 Concluding Remarks

This thesis has established a foundation for multi-user semantic communication systems that address efficiency, privacy, and performance requirements. The proposed architectures, namely diffusion-enhanced transmission, Wyner-Ziv neural coding, NOMA-DeepJSCC with user-specific projections, and OAC-based collaborative inference, demonstrate the potential of learned JSCC for multi-user scenarios in idealised settings.

At the same time, the gap between these results and real-world deployment remains substantial. The idealised channel models, perfect CSI assumptions, and simulation-only evaluation employed in this work are shared limitations across much of the semantic communications literature, but they constrain the practical conclusions that can be drawn. Bridging this gap will require not only more realistic evaluation and hardware prototyping, but also fundamental work on integrating

learned communication into existing protocol architectures, addressing rate adaptation, feedback mechanisms, and interoperability with established standards.

Despite these challenges, the core algorithmic contributions of this thesis, including perceptual quality optimisation via diffusion models, parameter-efficient multi-user scaling via learned projections, progressive training for interference management, and privacy amplification through channel noise, represent techniques that can be adapted and extended as the field matures toward practical deployment. The principles developed here contribute to the ongoing effort to build intelligent, efficient, and private communication systems for emerging applications in artificial intelligence, IoT, and beyond.

Bibliography

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

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

[3] J. Xu, T.-Y. Tung, B. Ai, W. Chen, Y. Sun, and D. Gündüz, "Deep joint source-channel coding for semantic communications," *IEEE communications Magazine*, vol. 61, no. 11, pp. 42–48, 2023.

[4] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[5] Y. Wang, J. Yu, and J. Zhang, "Zero-shot image restoration using denoising diffusion null-space model," *The Eleventh International Conference on Learning Representations*, 2023.

[6] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[7] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[8] L. Dai, B. Wang, Y. Yuan, S. Han, I. Chih-Lin, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, 2015.

[9] S. Verdu, *Multiuser detection*. Cambridge university press, 1998.

[10] 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.

[11] W. Li, H. Liang, C. Dong, X. Xu, P. Zhang, and K. Liu, "Non-orthogonal multiple access enhanced multi-user semantic communication," *IEEE Transactions on Cognitive Communications and Networking*, 2023.

[12] T. Wu, Z. Chen, M. Tao, B. Xia, and W. Zhang, "Fusion-based multi-user semantic communications for wireless image transmission over degraded broadcast channels," in *IEEE Global Communications Conference*, 2023, pp. 7623–7628.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[14] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[15] F. Wilhelmi, J. Hribar, S. F. Yilmaz, E. Ozfatura, K. Ozfatura, O. Yildiz, D. Gunduz, H. Chen, X. Ye, L. You, Y. Shao, P. Dini, and B. Bellalta, "Federated spatial reuse optimization in next-generation decentralized IEEE 802.11 WLANS," *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 2, pp. 117–133, Jul. 2022. [Online]. Available: https://doi.org/10.52953/tnyt6291

[16] N. Pavlidis, V. Perifanis, S. F. Yilmaz, F. Wilhelmi, E. Guerra, M. Miozzo, P. S. Efraimidis, R.-A. Koutsiamanis, P. Mulinka, and P. Dini, "Federated learning in mobile networks: A comprehensive case study on traffic forecasting," *IEEE Transactions on Sustainable Computing*, 2024.

[17] S. F. Yilmaz, B. Hasircioglu, L. Qiao, and D. Gunduz, "Private collaborative edge inference via over-the-air computation," *arXiv preprint arXiv:2407.21151*, 2024.

[18] S. F. Yilmaz, B. Hasırcıoğlu, and D. Gündüz, "Over-the-air ensemble inference with model privacy," in *IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 1265–1270.

[19] S. F. Yilmaz, C. Karamanlı, and D. Gündüz, "Distributed deep joint source-channel coding over a multiple access channel," in *IEEE International Conference on Communications*, 2023, pp. 1400–1405.

[20] V. Perifanis, N. Pavlidis, S. F. Yilmaz, F. Wilhelmi, E. Guerra, M. Miozzo, P. S. Efraimidis, P. Dini, and R.-A. Koutsiamanis, "Towards energy-aware federated traffic prediction for cellular networks," in *International Conference on Fog and Mobile Edge Computing (FMEC)*, 2023, pp. 93–100.

[21] S. F. Yilmaz, E. Ozyilkan, D. Gündüz, and E. Erkip, "Distributed deep joint source-channel coding with decoder-only side information," in *IEEE International Conference on Machine Learning for Communication and Networking*, 2024, pp. 139–144.

[22] S. F. Yilmaz, X. Niu, B. Bai, W. Han, L. Deng, and D. Gündüz, "High perceptual quality wireless image delivery with denoising diffusion models," in *IEEE Conference on Computer Communications Workshops*, 2024, pp. 1–5.

[23] S. F. Yilmaz, C. Karamanli, and D. Gunduz, "Learning to interfere in non-orthogonal multiple-access joint source-channel coding," *arXiv preprint arXiv:2504.03690*, 2025.

[24] J. Chen, S. F. Yilmaz, D. You, P. L. Dragotti, and D. Gündüz, "Sing: Semantic image communications using null-space and inn-guided diffusion models," *arXiv preprint arXiv:2503.12484*, 2025.

[25] A. Kurmukova, S. F. Yilmaz, E. Ozfatura, and D. Gunduz, "Transcoder: A neural-enhancement framework for channel codes," *arXiv preprint arXiv:2511.22539*, 2025.

[26] A. Goldsmith, "Joint source/channel coding for wireless channels," in *IEEE Vehicular Technology Conference*, vol. 2, 1995, pp. 614–618.

[27] 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.

[28] 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.

[29] T. Han, Q. Yang, Z. Shi, S. He, and Z. Zhang, "Semantic-preserved communication system for highly efficient speech transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 245–259, 2023.

[30] 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.

[31] C. Bian, Y. Shao, H. Wu, and D. Gündüz, "Deep joint source-channel coding over cooperative relay networks," *arXiv preprint arXiv:2211.06705*, 2022.

[32] 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.

[33] X. Han, Y. Wu, Z. Gao, B. Feng, Y. Shi, D. Gündüz, and W. Zhang, "Scsc: A novel standards-compatible semantic communication framework for image transmission," *IEEE Transactions on Communications*, 2025.

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

[35] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7939–7948.

[36] 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, 2021.

[37] X. Niu, X. Wang, D. Gündüz, B. Bai, W. Chen, and G. Zhou, "A hybrid wireless image transmission scheme with diffusion," in *Proc. of IEEE Int'l Workshop on Signal Proc. Adv. in Wireless Comm. (SPAWC)*, 2023.

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014.

[39] Y. M. Saidutta, A. Abdi, and F. Fekri, "VAE for joint source-channel coding of distributed Gaussian sources over AWGN MAC," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[40] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf

[41] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://openreview.net/forum?id=Hk99zCeAb

[42] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[43] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *International conference on machine learning*, 2017, pp. 537–546.

[44] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "Pulse: Self-supervised photo upsampling via latent space exploration of generative models," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2020, pp. 2437–2445.

[45] 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)*. IEEE, 2022, pp. 2989–2993.

[46] T. Marchioro, N. Laurenti, and D. Gündüz, "Adversarial networks for secure wireless communications," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8748–8752.

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

[48] G. Daras, J. Dean, A. Jalal, and A. G. Dimakis, "Intermediate layer optimization for inverse problems using deep generative models," in *International Conference on Machine Learning*, 2021.

[49] S. Tang, Q. Yang, D. Gündüz, and Z. Zhang, "Evolving semantic communication with generative modelling," in *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2024, pp. 1–6.

[50] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "Ilvr: Conditioning method for denoising diffusion probabilistic models. in 2021 ieee," in *CVF international conference on computer vision (ICCV)*, vol. 1, 2021, p. 2.

[51] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *The Eleventh International Conference on Learning Representations*, 2023.

[52] T. Wu, Z. Chen, D. He, L. Qian, Y. Xu, M. Tao, and W. Zhang, "CDDM: Channel denoising diffusion models for wireless semantic communications," *IEEE Transactions on Wireless Communications*, 2024.

[53] L. Guo, W. Chen, Y. Sun, B. Ai, N. Pappas, and T. Quek, "Diffusion-driven semantic communication for generative models with bandwidth constraints," *arXiv preprint arXiv:2407.18468*, 2024.

[54] M. Zhang, H. Wu, G. Zhu, R. Jin, X. Chen, and D. Gündüz, "Semantics-guided diffusion for deep joint source-channel coding in wireless image transmission," *arXiv preprint arXiv:2501.01138*, 2025.

[55] B. Li, Y. Liu, X. Niu, B. Bait, W. Han, L. Deng, and D. Gunduz, "Extreme video compression with prediction using pre-trained diffusion models," in *16th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2024, pp. 1449–1455.

[56] J. Chen, D. You, D. Gündüz, and P. L. Dragotti, "Commin: Semantic image communications as an inverse problem with INN-guided diffusion models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 6675–6679.

[57] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, Jun 2018, pp. 586–595.

[58] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," 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_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[59] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," in *International Conference on Learning Representations*, 2018.

[60] M. Letafati, S. A. A. Kalkhoran, E. Erdemir, B. H. Khalaj, H. Behroozi, and D. Gündüz, "Deep joint source channel coding for privacy-aware end-to-end image transmission," *IEEE Transactions on Machine Learning in Communications and Networking*, 2025.

[61] K. Anjum, Z. Qi, and D. Pompili, "Deep joint source-channel coding for underwater image transmission," in *International Conference on Underwater Networks & Systems*, New York, NY, USA, 2022. [Online]. Available: https://doi.org/10.1145/3567600.3568138

[62] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[63] T.-Y. Tung and D. Gündüz, "Deep joint source-channel and encryption coding: Secure semantic communications," in *IEEE International Conference on Communications*, 2023, pp. 5620–5625.

[64] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.

[65] N. Mital, E. Özyılkan, A. Garjani, and D. Gündüz, "Neural distributed image compression using common information," in *IEEE Data Compression Conference (DCC)*, 2022, pp. 182–191.

[66] E. Özyı lkan, J. Balle, and E. Erkip, "Learned Wyner–Ziv compressors recover binning," in *2023 IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 701–706.

[67] X. Zhang, J. Shao, and J. Zhang, "LDMIC: Learning-based distributed multi-view image coding," in *The Eleventh International Conference on Learning Representations*, 2023.

[68] R. Ahlswede, "Multi-way communication channels," in *Proceedings of the 2nd International Symposium on Information Theory*, Tsahkadsor, Armenian SSR, 1971, pp. 23–52.

[69] H. Liao, "Multiple access channels," Ph.D. dissertation, University of Hawaii, 1972.

[70] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[71] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[72] Y. Bo, S. Shao, and M. Tao, "Deep learning based superposition coded modulation for hierarchical semantic communications over broadcast channels," *IEEE Transactions on Communications*, 2024.

[73] Y. Zhang, R. Zhong, Y. Liu, W. Xu, and P. Zhang, "Interference suppressed NOMA for semantic-aware communication networks," *IEEE Transactions on Wireless Communications*, 2024.

[74] T. Tang, A. Wang, S. Muhaidat, S. Li, M. Li, and J. Liang, "MDC-NOMA: Multiple description coding-based nonorthogonal multiple access for image transmission," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3632–3641, 2020.

[75] S. Li, F. Meng, J. Xiong, L. Bariah, S. Muhaidat, and A. Wang, "STBC-assisted MDC-NOMA image transmission scheme for multi-antenna systems," *IEEE Transactions on Broadcasting*, vol. 68, no. 3, pp. 677–688, 2022.

[76] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deepsic: Deep soft interference cancellation for multiuser MIMO detection," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1349–1362, 2020.

[77] I. Sim, Y. G. Sun, D. Lee, S. H. Kim, J. Lee, J.-H. Kim, Y. Shin, and J. Y. Kim, "Deep learning based successive interference cancellation scheme in nonorthogonal multiple access downlink network," *Energies*, vol. 13, no. 23, p. 6237, 2020.

[78] T. Van Luong, N. Shlezinger, C. Xu, T. M. Hoang, Y. C. Eldar, and L. Hanzo, "Deep learning based successive interference cancellation for the non-orthogonal downlink," *IEEE Transactions on Vehicular Technology*, 2022.

[79] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 364–375, 1996.

[80] A. J. Grant, B. Rimoldi, R. L. Urbanke, and P. A. Whiting, "Rate-splitting multiple access for discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 873–890, 2001.

[81] B. Clerckx, H. Joudeh, C. Hao, M. Dai, and B. Rassouli, "Rate splitting for mimo wireless networks: a promising phy-layer strategy for lte evolution," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 98–105, 2016.

[82] B. Clerckx, Y. Mao, Z. Yang, M. Chen, A. Alkhateeb, L. Liu, M. Qiu, J. Yuan, V. W. S. Wong, and J. Montojo, "Multiple access techniques for intelligent and multifunctional 6g: Tutorial, survey, and outlook," *Proceedings of the IEEE*, vol. 112, no. 7, pp. 832–879, 2024.

[83] F. Karim, N. H. Mahmood, A. S. d. Sena, D. Kumar, B. Clerckx, and M. Latva-Aho, "Uplink rate splitting multiple access with imperfect channel state information and interference cancellation," *IEEE Wireless Communications Letters*, vol. 14, no. 5, pp. 1316–1320, 2025.

[84] Z. Qiu, Y. Mao, S. Ma, and B. Clerckx, "Robust max–min fair beamforming design for rate splitting multiple access-aided visible light communications," *IEEE Internet of Things Journal*, vol. 12, no. 8, pp. 10 043–10 057, 2025.

[85] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3579–3605, 2021.

[86] N. Shlezinger and I. V. Bajić, "Collaborative inference for AI-empowered IoT devices," *IEEE Internet of Things Magazine*, vol. 5, no. 4, pp. 92–98, 2022.

[87] Z. Hu, A. B. Tarakji, V. Raheja, C. Phillips, T. Wang, and I. Mohomed, "Deephome: Distributed inference with heterogeneous devices in the edge," in *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications*, 2019, pp. 13–18.

[88] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[89] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2604–2609.

[90] 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.

[91] ——, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.

[92] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3643–3658, 2021.

[93] M. S. E. Mohamed, W.-T. Chang, and R. Tandon, "Privacy amplification for federated learning via user sampling and wireless aggregation," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3821–3835, 2021.

[94] B. Hasırcıoğlu and D. Gündüz, "Private wireless federated learning with anonymous over-the-air computation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 5195–5199.

[95] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[96] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.

[97] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 170–185, 2020.

[98] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning: Privacy and Incentive*. Springer, 2020, pp. 240–254.

[99] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, and A. Y. Zomaya, "Federated learning for Covid-19 detection with generative adversarial networks in edge cloud computing," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 10 257–10 271, 2021.

[100] I. Shiri, A. Vafaei Sadr, A. Akhavan, Y. Salimi, A. Sanaat, M. Amini, B. Razeghi, A. Saberi, H. Arabi, S. Ferdowsi *et al.*, "Decentralized collaborative multi-institutional pet attenuation and scatter correction using federated deep learning," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 50, no. 4, pp. 1034–1050, 2023.

[101] M. B. Mashhadi, M. Jankowski, T.-Y. Tung, S. Kobus, and D. Gündüz, "Federated mmWave beam selection utilizing LIDAR data," *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2269–2273, 2021.

[102] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, 2006, pp. 265–284.

[103] T. G. Dietterich *et al.*, "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, no. 1, pp. 110–125, 2002.

[104] C. M. Bishop *et al.*, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[105] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018.

[106] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996.

[107] K. M. Ting and I. H. Witten, "Stacking bagged and dagged models," *(Working paper 97/09). Hamilton, New Zealand: University of Waikato, Department of Computer Science*, 1997.

[108] S. Shi, F. Nie, R. Wang, and X. Li, "When multi-view classification meets ensemble learning," *Neurocomputing*, vol. 490, pp. 17–29, 2022.

[109] B. Mandıra, D. Giritlioglu, S. F. Yilmaz, C. U. Ertenli, B. F. Akgür, M. Kınıklıoglu, A. G. Kurt, M. N. Doganlı, E. Mutlu, S. C. Gürel *et al.*, "Spatiotemporal and multimodal analysis of personality traits," in *15th International Summer Workshop on Multimodal Interfaces*, 2019, pp. 32–44.

[110] D. Giritlioğlu, B. Mandira, S. F. Yilmaz, C. U. Ertenli, B. F. Akgür, M. Kınıklıoğlu, A. G. Kurt, E. Mutlu, Ş. C. Gürel, and H. Dibeklioğlu, "Multimodal analysis of personality traits on videos of self-presentation and induced behavior," *Journal on Multimodal User Interfaces*, vol. 15, no. 4, pp. 337–358, 2021.

[111] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[112] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[113] Z.-H. Zhou, *Ensemble Learning*. Singapore: Springer Singapore, 2021, pp. 181–210. [Online]. Available: https://doi.org/10.1007/978-981-15-1967-3_8

[114] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[115] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[116] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[117] L. Liu, Y. Li, Z. Kuang, J.-H. Xue, Y. Chen, W. Yang, Q. Liao, and W. Zhang, "Towards impartial multi-task learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235613618

[118] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7482–7491.

[119] D. Gündüz, M. A. Wigger, T.-Y. Tung, P. Zhang, and Y. Xiao, "Joint source–channel coding: Fundamentals and recent progress in practical designs," *Proceedings of the IEEE*, 2024.

[120] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Springer International Workshop on Similarity-Based Pattern Recognition*, 2015, pp. 84–92.

[121] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[122] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4004–4012.

[123] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

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

[125] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," *International Journal of Computer Vision*, vol. 130, no. 6, pp. 1526–1565, 2022.

[126] S. Dörner, S. Cammerer, J. Hoydis, and S. Ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2017.

[127] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[128] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, pp. 1398–1402.

[129] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[130] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[131] S. Shamai, S. Verdu, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 564–579, 1998.

[132] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, p. 359–366, Jul 1989.

[133] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[134] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, Jun 2016, pp. 3213–3223.

[135] S. Ayzik and S. Avidan, "Deep image compression using decoder side information," in *Springer European Conference on Computer Vision*, 2020, pp. 699–714.

[136] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[137] M. Menze, C. Heipke, and A. Geiger, "Joint 3D estimation of vehicles and scene flow," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W5, pp. 427–434, 08 2015.

[138] ——, "Object scene flow," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, 11 2017.

[139] N. Mital, E. Özyilkan, A. Garjani, and D. Gündüz, "Neural distributed image compression with cross-attention feature alignment," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 2498–2507.

[140] D. Gündüz, E. Erkip, A. Goldsmith, and H. V. Poor, "Source and channel coding for correlated sources over multiuser channels," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3927–3944, 2009.

[141] K. Ramchandran, A. Ortega, K. M. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 1, pp. 6–23, 1993.

[142] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Joint source-channel coding for video communications," *Handbook of Image and Video Processing*, pp. 1065–1082, 2005.

[143] O. Y. Bursalioglu, G. Caire, and D. Divsalar, "Joint source-channel coding for deep-space image transmission using rateless codes," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3448–3461, 2013.

[144] 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.

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

[146] 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.

[147] T.-Y. Tung, D. B. Kurka, M. Jankowski, and D. Gündüz, "Deepjscc-q: Channel input constrained deep joint source-channel coding," in *IEEE International Conference on Communications*, 2022, pp. 3880–3885.

[148] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, no. 1, 2015, pp. 1–30.

[149] 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, 2022.

[150] C. Bian, Y. Shao, and D. Gündüz, "A hybrid joint source-channel coding scheme for mobile multi-hop networks," in *ICC 2024-IEEE International Conference on Communications*, 2024, pp. 986–992.

[151] C. Bian, Y. Shao, H. Wu, E. Ozfatura, and D. Gündüz, "Process-and-forward: Deep joint source-channel coding over cooperative relay networks," *IEEE Journal on Selected Areas in Communications*, 2025.

[152] F. Mezzadri, "How to generate random matrices from the classical compact groups," *arXiv preprint math-ph/0609050*, 2006.

[153] F. Bellard, *Better Portable Graphics*, 2014 (accessed September 1, 2025), https://bellard.org/bpg/.

[154] W. Jiang and R. Wang, "MLIC++: Linear complexity multi-reference entropy modeling for learned image compression," in *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

[155] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[156] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[157] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia *et al.*, "Machine learning at Facebook: Understanding inference at the edge," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2019, pp. 331–344.

[158] Q. Lan, Q. Zeng, P. Popovski, D. Gündüz, and K. Huang, "Progressive feature transmission for split classification at the wireless edge," *IEEE Transactions on Wireless Communications*, vol. 22, no. 6, pp. 3837–3852, 2022.

[159] 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.

[160] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Guest editorial special issue on distributed learning over wireless edge networks—part i," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3575–3578, 2021.

[161] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Joint device-edge inference over wireless links with pruning," in *IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[162] 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.

[163] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3–18.

[164] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 601–618.

[165] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[166] M. Malekzadeh, B. Hasircioglu, N. Mital, K. Katarya, M. E. Ozfatura, and D. Gündüz, "Dopamine: Differentially private federated learning on medical data," *The Second AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI-21)*, 2021.

[167] A. Şahin and R. Yang, "A survey on over-the-air computation," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1877–1908, 2023.

[168] A. Sonee and S. Rini, "Efficient federated learning over multiple access channel with differential privacy constraints," *arXiv preprint arXiv:2005.07776*, 2020.

[169] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 1897–1914.

[170] J. Hayes, B. Balle, and S. Mahloujifar, "Bounding training data reconstruction in dp-sgd," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[171] T. Steinke, "Composition of differential privacy & privacy amplification by subsampling," *arXiv preprint arXiv:2210.00597*, 2022.

[172] A. Lowy, Z. Li, J. Liu, T. Koike-Akino, K. Parsons, and Y. Wang, "Why does differential privacy with large epsilon defend against practical membership inference attacks?" *arXiv preprint arXiv:2402.09540*, 2024.

[173] J. Lee and C. Clifton, "How much is enough? choosing ε for differential privacy," in *International Conference on Information Security*, 2011, pp. 325–340.

[174] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Advances in Neural Information Processing Systems*, 2015.

[175] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 115–124.

[176] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel *et al.*, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[177] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[178] Y. Shao, D. Gündüz, and S. C. Liew, "Federated edge learning with misaligned over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3951–3964, 2021.

[179] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 394–403.

[180] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 6280–6290.

[181] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[182] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[183] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3498–3505.

[184] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "CARER: Contextualized affect representations for emotion recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 3687–3697. [Online]. Available: https://www.aclweb.org/anthology/D18-1404

[185] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[186] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1485–1488.

[187] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[188] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[189] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[190] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[191] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[192] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[193] D. Wang, H. Xu, and Q. Wu, "Averaging versus voting: A comparative study of strategies for distributed classification," *Mathematical Foundations of Computing*, vol. 3, no. 3, p. 185, 2020.

[194] P. Li and X. Li, "Differential privacy with random projections and sign random projections," *arXiv preprint arXiv:2306.01751*, 2023.

[195] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[196] V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2022, pp. 954–964.

[197] S. Razavikia, J. M. B. Da Silva, and C. Fischione, "Channelcomp: A general method for computation by communications," *IEEE Transactions on Communications*, 2023.

[198] A. Şahin, "Over-the-air computation based on balanced number systems for federated edge learning," *IEEE Transactions on Wireless Communications*, 2023.

[199] L. Qiao, Z. Gao, M. B. Mashhadi, and D. Gündüz, "Massive digital over-the-air computation for communication-efficient federated edge learning," *IEEE Journal on Selected Areas in Communications*, 2024.

[200] A. Pérez-Neira, M. Martinez-Gost, A. Şahin, S. Razavikia, C. Fischione, and K. Huang, "Waveforms for computing over the air," *arXiv preprint arXiv:2405.17007*, 2024.

[201] M. M. Amiri, T. M. Duman, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Blind federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5129–5143, 2021.

[202] Z. Wang, Y. Zhao, Y. Zhou, Y. Shi, C. Jiang, and K. B. Letaief, "Over-the-air computation for 6g: Foundations, technologies, and applications," *IEEE Internet of Things Journal*, 2024.

[203] E. Alshina, J. Ascenso, and T. Ebrahimi, "Jpeg ai: The first international standard for image coding based on an end-to-end learning-based approach," *IEEE MultiMedia*, vol. 31, no. 4, pp. 60–69, 2024.