Task-Oriented Communication for Edge Intelligence Enabled Connected Robotics Systems

Yufeng Diao

Submitted in fulfillment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

July 2025

Abstract

Traditional digital communication systems are built on the principle of source-channel separation, guided by rate-distortion theory and channel coding. This reconstruction-oriented communication paradigm served as a cornerstone through multiple generations of communication technologies. However, with the rise of machine-to-machine communications and human-to-machine interactions, task-specific representations are often more compact and more efficient than full-scale reconstructions, and End-to-End (E2E) trained communication systems have demonstrated superior task performance over traditional communications. This thesis explores task-oriented communication as a paradigm shift from traditional reconstruction-oriented transmission, focusing on optimizing data exchange for machine-driven decision-making rather than full data fidelity.

We develop a Task-Oriented Source-Channel Coding (TSCC) framework designed for edge-enabled autonomous driving. By integrating deep learning-based Joint Source-Channel Coding (JSCC) with an end-to-end autonomous driving agent, TSCC minimizes communication overhead while maintaining high inference accuracy, ensuring robustness against noisy channels. Our results demonstrate a 98.36% reduction in communication bandwidth while maintaining driving performance under low Signal-to-Noise Ratio (SNR) conditions.

To enhance compatibility with existing digital communication infrastructures, we propose Aligned Task- and Reconstruction-Oriented Communication (ATROC), which bridges task-oriented communication with traditional reconstruction-oriented paradigms. By leveraging an information reshaper and variational information bottleneck (VIB) theory, ATROC improves AI-driven inference on edge servers while ensuring seamless integration with digital communication standards. Experimental results validate that ATROC reduces 99.19% of the communication load while preserving autonomous driving efficiency.

Recognizing the need for a holistic approach, we introduce a task-oriented co-design of communication, computing, and control framework tailored for edge-enabled industrial Cyber-Physical Systems (CPS). This framework jointly optimizes data transmission, computational efficiency, and control decisions, and integrates task-oriented JSCC with Delay-aware Trajectory-guided Control Prediction (DTCP) to reduce E2E delay. Experimental results in autonomous driving simulations demonstrate that our co-design approach significantly improves driving performance under high latency scenarios.

To my mentors for wisdom, my colleagues for camaraderie, my family for their unwavering support, and my wife for her love.

Declaration

I confirm that this thesis, “Task-Oriented Communication for Edge Intelligence Enabled Connected Robotics Systems,” is my own original work and has not been submitted for any other degree or professional qualification. I affirm that all research conducted adheres to ethical guidelines and complies with the Code of Good Practice of the University of Glasgow. All necessary permissions for the inclusion of copyrighted materials have been obtained, and the work is presented in accordance with the regulations governing the presentation of theses at the University of Glasgow.

List of Publications

1. **Y. Diao***, Y. Zhang, D. De Martini, P. G. Zhao, and E. L. Li, "Task-Oriented Co-Design of Communication, Computing, and Control for Edge-Enabled Industrial Cyber-Physical Systems," *accepted by IEEE Journal on Selected Areas in Communications (JSAC)*, 2025. (* Corresponding Author)

2. **Y. Diao***, Y. Zhang, C. She, P. G. Zhao, and E. L. Li, "Aligning Task- and Reconstruction-Oriented Communications for Edge Intelligence," *in IEEE Journal on Selected Areas in Communications*, vol. 43, no. 7, pp. 2575-2588, 2025. (* Corresponding Author)

3. **Y. Diao**, Z. Meng, X. Xu, C. She, and G. Zhao, "Task-Oriented Source-Channel Coding Enabled Autonomous Driving Based on Edge Computing," *in Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 1-6.

4. **Y. Diao**, Y. Zhang, G. Zhao, and D. De Martini, "TAGIC: Task-Guided Image Communication Framework for Seamless Teleoperation," *in Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2024, pp. 1-2.

5. **Y. Diao**, Y. Zhang, G. Zhao, and M. Khamis, "Drone Authentication via Acoustic Fingerprint," *in Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC)*, 2022, pp. 658–668.

6. **Y. Diao**, H. Dai, G. Zhao, and D. De Martini, "Keyframe Selection, Communication, and Prediction for Teleoperated Driving Systems," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS Poster)*, 2022.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AEP	Asymptotic Equipartition Property
AI	Artificial Intelligence
AoI	Age of Information
AR	Augmented Reality
ATROC	Aligned Task- and Reconstruction-Oriented Communication
AWGN	Additive White Gaussian Noise
BER	Bit-Error Rate
BEV	Bird's-eye View
BLEU	Bilingual Evaluation Understudy
CARLA	Car Learning to Act
CPS	Cyber-Physical System
CSI	Channel State Information
CVAE	Conditional Variational Autoencoder
DMC	Discrete Memoryless Channel
DNN	Deep Neural Network
DPGM	Directed Probabilistic Graphical Model
DRL	Deep Reinforcement Learning
DTCP	Delay-aware Trajectory-guided Control Prediction
E2E	End-to-End
FID	Fréchet Inception Distance
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
IB	Information Bottleneck
IoT	Internet of Things

JSCC Joint Source-Channel Coding

KB Knowledge Base
KL Kullback-Leibler
KPI Key Performance Indicator

LDPC Low-Density Parity-Check

M2M Machine-to-Machine
MS-SSIM Multi-Scale Structural Similarity

OFDM Orthogonal Frequency-Division Multiplexing

PSNR Peak Signal-to-Noise Ratio

QAM Quadrature Amplitude Modulation
QoS Quality of Service

SNR Signal-to-Noise Ratio
SSIM Structural Similarity

TGCP Trajectory-Guided Control Prediction
TSCC Task-oriented Source-Channel Coding

URLLC Ultra-Reliable Low-Latency Communication

VAE Variational Autoencoder
VIB Variational Information Bottleneck
VoI Value of Information
VR Virtual Reality

Chapter 1

Introduction

1.1 Task-Oriented Communication

Before the emergence of task-oriented communication, the history of traditional mobile communications can be traced back to 1897 [1]. Since then, the field has witnessed remarkable advancements. In particular, modern 5G networks not only improve human-to-human communication but also enable seamless connectivity for human-to-machine, and machine-to-machine [2]. Traditional communication systems were designed to maximize signal fidelity while minimizing distortion. However, conventional approaches were mainly reconstruction-oriented, aiming to reconstruct the original signal at the receiver without considering whether transmitted information was necessary for performing the final task.

In classical Shannon's information theory, communication is modeled as a process of transmitting symbols through a noisy channel, where the objective is to minimize the error rate between transmitted and received messages. Based on this, separate source and channel coding were developed, such as JPEG [3] and JPEG2000 [4] for image compression and LDPC [5] codes for error correction. These methods, while efficient in preserving fidelity, do not differentiate between mission-critical and task-agnostic information, leading to unnecessary transmission overhead, especially in increasing machine-to-machine communications.

Task-oriented communication is a paradigm where the transmitted message does not necessarily need to be reconstructed exactly on the receiver side, unlike traditional reconstruction-oriented communication. Instead, the objective is to transmit minimal yet sufficient information that enables the receiver to effectively perform specific tasks (e.g., object detection, decision-making, and control). For example, the transmitter may send an image while the receiver recovers only a semantic summary (e.g., a descriptive sentence or latent feature representation).

This task-oriented framework is related to the classical rate-distortion theory, which provides a foundational theoretical connection. The rate-distortion theory seeks to optimize the balance between the compression rate (bandwidth or data transmission rate) and the fidelity of reconstructed data (distortion). Task-oriented communication extends this classical concept by redefining the distortion to represent task performance rather than data fidelity. In other words, the distortion metric in the rate-distortion theory is replaced with a task-specific

performance measure. Thus, the communication system aims to minimize the rate while satisfying a constraint on the task performance (task-oriented distortion) or, equivalently, to minimize distortion while adhering to a given communication rate constraint. Task-oriented communication significantly reduces bandwidth consumption, reduces latency, and enhances robustness in noisy environments, making it particularly suitable for AI-driven applications with large data throughput.

Driven by the growing demand for task-oriented communication designs, researchers have increasingly explored the Information Bottleneck (IB) approach. The IB [6] method seeks to maximize the preservation of task-specific information while minimizing task-agnostic information from input. The traditional IB approach relied on the computationally intensive Blahut-Arimoto algorithm [7], [8], which was impractical for deep learning applications due to its complexity [9]. This limitation was addressed by the introduction of a variational approach to the IB method, known as Variational Information Bottleneck (VIB) [10], which made it feasible to apply IB principles in deep learning by approximating the true posterior with a variational distribution.

Recent studies have successfully integrated VIB with deep Joint Source-Channel Coding (JSCC), formalizing task-oriented communication strategies that outperform traditional reconstruction-oriented frameworks. For example, recent works [11], [12] have demonstrated that combining VIB with deep JSCC can significantly improve communication efficiency and robustness, particularly in scenarios where it is essential to prioritize task-specific information over the fidelity of raw data. Another study [13] focused on applying semantic communication for camera relocalization, optimizing the trade-off between inference accuracy and End-to-End (E2E) latency.

1.1.1 From Semantic Communication to Task-Oriented Communication

Semantic communication marks a major shift from bit-wise fidelity to meaning preservation. However, in many real-world cyber-physical and autonomous systems, even meaning is not the end goal. Instead, the objective is to complete a task, such as classifying an object, controlling a drone, or localizing a robot, based on sensed or transmitted input. In these settings, even an accurate semantic reconstruction might be redundant if it includes task-agnostic features.

Task-oriented communication goes one step further: It only preserves the information that is directly relevant to a downstream task. This makes it a subset of semantic communication:

- Semantic communication seeks mutual understanding or meaning alignment (e.g., in natural language translation).
- Task-oriented communication transmits minimal sufficient information to optimize task performance, regardless of whether the transmitted data can be semantically interpreted in isolation.

For example, in autonomous driving, semantic communication can seek to transmit a segmented road map, while task-oriented communication transmits only the features necessary for a lane-keeping decision, thus improving bandwidth and latency efficiency.

1.1.2 Theoretical Foundations: Information Bottleneck and Minimal Sufficient Statistics

Task-oriented communication is rigorously grounded in the IB principle, introduced by [6]. The IB method formalizes the goal of compressing an input signal X into a representation Z that retains maximum relevance to a target variable Y (e.g., a classification label or control action), while minimizing its mutual information with the input:

$$\min_{p(z|x)} I(X;Z) - \beta I(Z;Y). \tag{1.1}$$

This formulation encourages models to filter out task-agnostic details, making it ideally suited for task-driven systems operating under communication constraints.

In communication systems, the encoder now plays an active role in jointly learning what to transmit for maximal task accuracy at the receiver, often under energy, latency, and bandwidth constraints. Task-oriented communication also aligns closely with concepts like:

- Minimal sufficient statistics,
- Rate-distortion tradeoffs,
- Functional compression (transmitting functions of input data, rather than the input itself).

These principles ensure that the communication pipeline is tightly integrated with machine learning and control goals, rather than functioning as an isolated module.

1.1.3 Learning-Based Architectures and Joint Optimization

A cornerstone of modern task-oriented communication systems is their reliance on E2E learning architectures, particularly those built on Deep Neural Networks (DNNs) that jointly optimize for both communication efficiency and task performance. Unlike traditional communication models, which separate the pipeline into source encoding, channel encoding, modulation, and finally decoding, task-oriented systems fuse these components into a single, trainable framework that learns to extract and transmit only mission-critical information.

The Novelty of JSCC in Task-Oriented Communication

At the heart of this transformation lies JSCC, a concept that violates the classic Shannon separation theorem. Traditionally, Shannon's theory asserts that optimal performance in communication systems can be achieved by separating source and channel coding under the assumption of infinite block lengths and latency tolerance. However, in low-latency, high-mobility, and edge-deployed systems, these assumptions do not hold. JSCC, particularly deep JSCC, enables the encoding of input features directly into channel symbols, bypassing the need for rigid modular compression and coding schemes.

The milestone work in this direction is [14], which introduced an autoencoder-based framework for wireless image transmission where the encoder and decoder were trained as neural networks under the presence of a simulated noisy channel (e.g., AWGN). The encoder directly mapped the input images to channel symbols, and the decoder reconstructed the image at the receiver. Importantly, the model learned to allocate redundancy and compression adaptively based on content and channel conditions, something hard-coded schemes struggle to do.

This work demonstrated the key benefit of JSCC:

- **Graceful degradation:** Unlike digital schemes that exhibit catastrophic failure under poor channel conditions, deep JSCC degrades smoothly.
- **E2E differentiability:** The entire encoder–channel–decoder system can be trained jointly with gradient descent.
- **Bandwidth-quality tradeoff:** The system learns to balance semantic richness with transmission constraints implicitly.

Extension to Task-Oriented Architectures

While the [14] targeted image reconstruction, subsequent work extended this model to task-oriented scenarios, where the goal is not reconstruction but classification, control, or regression based on received features.

In a typical task-oriented JSCC framework, the system consists of the following modules:

- **Task-Aware Encoder:** Maps the input (e.g., image, time-series, point cloud) into a low-dimensional latent space that is task-discriminative.
- **Channel Layer:** Simulates realistic transmission conditions, such as AWGN, Rayleigh fading, or quantization noise. This layer must be differentiable to allow for gradient flow during training.
- **Task Decoder:** receives the noisy representation and performs the downstream tasks, such as classification, control decision, or bounding-box estimation.

1.2 Motivation and Research Questions

The growing complexity and autonomy of connected robotic systems demand a shift from traditional, reconstruction-oriented communication paradigms to more efficient and task-aware paradigms. In real-world scenarios, such as autonomous driving or industrial control, communication systems must prioritize providing the most relevant information for decision making, not simply reproducing fidelity. This transition is motivated by pressing challenges: limited bandwidth, stringent latency requirements, and extreme channel conditions in edge-deployed settings. Task-oriented communication, which focuses on transmitting only mission-critical data, emerges as a promising solution to meet these demands. However, fundamental

questions remain about how to design robust, learning-based communication schemes that are compatible with existing infrastructure and capable of operating in dynamic environments.

This thesis addresses the fundamental challenge of efficiently transmitting mission-critical information in edge-enabled autonomous systems, particularly under constraints of bandwidth, latency, and reliability. The main questions answered in this thesis are the following.

- **How can task-oriented communication frameworks be designed to prioritize task-specific features while maintaining robustness under limited bandwidth and noisy channels?**
  This question investigates the design of deep learning-based source-channel coding strategies that optimize end-to-end task performance, such as autonomous driving accuracy, rather than traditional reconstruction fidelity.

- **How can task-oriented communication be aligned with existing digital communication infrastructures to ensure seamless integration and compatibility?**
  This addresses the practical challenges of deploying task-oriented systems within existing edge intelligence infrastructures, where compatibility with standard modulation schemes and data formats is essential.

- **How can communication, computing, and control be co-designed in a task-oriented manner to meet the requirements of Ultra-Reliable Low-Latency Communication (URLLC) in industrial Cyber-Physical System (CPS)?**
  This question targets the combination of task-specific encoding with delay-aware prediction models to mitigate end-to-end latency and ensure reliable decision-making in real-time mission-critical applications.

1.3 Thesis Statement

This thesis proposes that task-oriented communication, when co-designed with computing and control strategies, can significantly enhance the performance, robustness, and efficiency of edge-enabled autonomous systems operating under constrained and dynamic environments. Departing from the traditional reconstruction-oriented paradigm, this work focuses on optimizing information transmission for task-specific objectives, such as real-time control decisions, rather than maximizing data fidelity. The central hypothesis of this thesis is:

Hypothesis *By prioritizing mission-critical information and suppressing task-agnostic content through a task-oriented communication framework – jointly optimized with computation and control components – edge-enabled autonomous systems can achieve superior performance and reliability under bandwidth constraints, high latency, and noisy channel conditions.*

1.4 Thesis Contributions

The main contributions of this thesis are summarized as follows:

Task-Oriented Source-Channel Coding (TSCC): We propose a novel deep TSCC framework based on modified Conditional Variational Autoencoder (CVAE) to enhance edge-enabled autonomous driving. Inspired by JSCC-enabled image transmission [14] and the integration of JSCC with Variational Autoencoder (VAE) [15], our approach jointly designs data transmission with autonomous driving agent, building a resilient E2E communication system under low SNR scenarios. Our approach prioritizes task-critical information by deploying an E2E autonomous driving agent as a training metric. We innovatively integrate β-CVAE with the autonomous driving agent, offering a holistic view of task-oriented source-channel coding. The major contributions of TSCC include:

- We design β-CVAE combining the autonomous driving agent with the source-channel coding, demonstrating a novel approach that integrates communications and computing in autonomous systems.
- We implement TSCC within an edge-enabled state-of-the-art E2E autonomous driving agent, demonstrating notable improvements in driving performance over traditional communication methods and state-of-the-art deep JSCC approaches in terms of driving score.

Aligned Task- and Reconstruction-Oriented Communications (ATROC): This work introduces a novel communication framework compatible with reconstruction-oriented communication, especially for edge inference. By extending IB theory [6] and incorporating JSCC modulation, Aligned Task- and Reconstruction-Oriented Communication (ATROC) is designed to enhance AI-driven machine-to-machine communication. It prioritizes task-specific information in data transmission, shifting focus from traditional signal reconstruction fidelity to operational efficiency and effectiveness in real-world task performance. The key contributions of ATROC are summarized as follows:

- Based on IB theory, we develop a framework that aligns task-oriented communications with reconstruction-oriented communications. The framework focuses on maximizing mutual information between inference results and encoded features, minimizing mutual information between the encoded features and the input data, and preserving task-specific information through the information reshaper.
- We introduce an information reshaper within our extended IB theory, laying a foundational aspect of ATROC. This reshaper is expert at transforming received symbols into task-specific data, maintaining the same data structure as the input while ensuring the preservation of task-specific information. This component is crucial for adapting the communication to the specific needs of the task without compromising the integrity of the transmitted data.

- Due to the intractability of mutual information in the training and inference of deep neural networks, we adapt a variational approximation approach, known as VIB. This approach allows us to establish a tractable upper bound for these terms, enabling training and inference of deep neural networks.
- We design a JSCC modulation scheme that aligns the JSCC symbols with a predefined constellation scheme. This scheme ensures compatibility of ATROC with classic modulation techniques, making it more adaptable to existing communication infrastructures.
- In our simulation, we validate that the ATROC framework reduced the communication load by 99. 19% in terms of bits per service, compared to existing methods, without compromising the driving score of the autonomous driving agent.

Task-Oriented Co-Design of Communication, Computing, and Control: In this work, our objective is to address three fundamental questions for edge-enabled mission-critical industrial CPS:

1. How can data transmission be optimized for bandwidth-constrained and latency-sensitive applications to ensure that task-specific information is prioritized?
2. How can predictive models be utilized to ensure that edge inference systems make decisions that reduce perceived E2E delay?
3. How can communication, computing, and control be jointly designed and optimized to meet the demands of URLLC in mission-critical applications?

The key contributions of this work are summarized as follows:

- We develop a comprehensive task-oriented co-design framework that jointly optimizes communication, computing, and control. This framework seamlessly integrates task-oriented JSCC with a delay-aware autonomous driving agent, addressing the critical challenges of bandwidth constraints, noise interference, and E2E delay to maximize performance for edge-enabled autonomous driving.
- We formulate the problem of task-oriented communication using the IB approach and employ a variational approximation to derive a tractable upper bound, resulting in the VIB method. Additionally, we extend the standard VIB framework to incorporate conditional information, such as vehicle and channel state information, ensuring better alignment with mission-critical applications. Our formulation improves communication efficiency in dynamic and noisy environments, which is essential for the reliable operation of industrial CPS.
- We establish the Delay-aware Trajectory-guided Control Prediction (DTCP) strategy for autonomous driving, which combines two dominant autonomous driving paradigms: trajectory planning and control prediction. The DTCP processes JSCC symbols, state information, and channel state to predict optimal driving actions that reduce perceived

E2E delay. In addition, DTCP is co-designed with the task-oriented JSCC and is jointly trained for machine-to-machine communication.

1.5 Thesis Organization

This thesis is organized as follows:

- In Chapter 2, we introduce the fundamental theories and prior research that form the basis of this thesis. It starts with rate-distortion theory, including the Asymptotic Equipartition Property (AEP) and channel capacity. And then it explores the Information Bottleneck (IB) Approach, which extends the rate-distortion framework for optimizing task-relevant information. Other important topics include JSCC, edge intelligence and predictions in URLLC applications.

- In Chapter 3, we introduce the foundation of task-oriented communication for autonomous systems. We propose a novel TSCC framework that jointly optimizes source coding and channel coding in a task-oriented manner. Specifically, to reduce communication overhead and guarantee autonomous driving performance, we leverage an autonomous driving agent to guide source-channel coding based on a modified CVAE. We test the proposed framework on a well-known autonomous driving platform with different communication channel conditions. The results show that compared to traditional communication and state-of-the-art deep JSCC, our proposed framework achieves superior performance by saving 98.36% communication overhead and maintains an 83.24% driving score even at 0 dB SNR.

- In Chapter 4, we discuss the integration of task-oriented communication with the existing communication paradigms and infrastructure. We propose a communication framework that aligns task-oriented and reconstruction-oriented communications for edge intelligence. The idea is to extend the Information Bottleneck (IB) theory to optimize data transmission by minimizing task-relevant loss function, while maintaining the structure of the original data by an information reshaper. Such an approach integrates task-oriented communications with reconstruction-oriented communications, where a variational approach is designed to handle the intractability of mutual information in high-dimensional neural network features. We also introduce a JSCC modulation scheme compatible with classical modulation techniques, enabling the deployment of AI technologies within existing digital infrastructures. The proposed framework is particularly effective in edge-based autonomous driving scenarios. Our evaluation in the Car Learning to Act (CARLA) simulator demonstrates that the proposed framework significantly reduces bits per service by 99.19% compared to existing methods, such as JPEG, JPEG2000, and BPG, without compromising the effectiveness of task execution.

- In Chapter 5, we explore the design of holistic systems based on task-oriented communication. We propose a task-oriented co-design framework that integrates communication,

computing, and control to address the key challenges of bandwidth limitations, noise interference, and latency in mission-critical industrial Cyber-Physical Systems (CPS). To improve communication efficiency and robustness, we design a task-oriented Joint Source-Channel Coding (JSCC) using Information Bottleneck (IB) to enhance data transmission efficiency by prioritizing task-specific information. To mitigate the perceived End-to-End (E2E) delays, we develop a Delay-Aware Trajectory-Guided Control Prediction (DTCP) strategy that integrates trajectory planning with control prediction, predicting commands based on E2E delay. Moreover, the DTCP is co-designed with task-oriented JSCC, focusing on transmitting task-specific information for timely and reliable autonomous driving. Experimental results in the CARLA simulator demonstrate that, under an E2E delay of 1 second (20 time slots), the proposed framework achieves a driving score of 48.12, which is 31.59 points higher than using Better Portable Graphics (BPG) while reducing bandwidth usage by 99.19%.

- In Chapter 6, we summarize the contributions of the thesis and discuss potential research directions.

Chapter 2

Background and Related Work

2.1 Historical Evolution of Communication Systems

The evolution of communication systems represents one of the most pivotal technological advances in human history. From early analog signals to modern intelligent, context-aware communication paradigms, each generation of systems has been built on the foundations of information theory, with increasing demands for speed, fidelity, and intelligence.

2.1.1 Early Foundations and Analog Communication

The roots of modern communication can be traced back to the late 19th century with the invention of the telegraph and telephone. Samuel Morse's telephone (1837) and Alexander Graham Bell's telephone (1876) established the foundations of real-time electrical communication over distances. These analog systems transmitted signals in continuous waveforms, susceptible to degradation due to noise and interference. Despite these limitations, they marked the birth of long-distance human-to-human communication.

2.1.2 The Digital Shift and Birth of Information Theory

A monumental shift occurred in 1948 when Claude Shannon introduced the mathematical theory of communication, now widely regarded as the foundation of digital communications. Shannon's work [16] established two key ideas: the **separation of source and channel coding**, and the **concept of channel capacity**, which defines the theoretical maximum rate at which data can be transmitted with arbitrarily low error.

This led to the design of modern modular digital systems where:

- Source coding (e.g., JPEG [3] for images, MP3 [17] for audio) compresses data to reduce redundancy.
- Channel coding (e.g., Hamming codes [18], convolutional codes [19], and Low-Density Parity-Check (LDPC) [5]) adds controlled redundancy to combat noise in communication channels.

The primitive digital communication systems assumed that the goal of communication is accurate signal reconstruction, an assumption that was held for decades and influenced generations of system design.

2.1.3 Generations of Mobile Communication: From Voice to Data

The generational progression of mobile communication technologies has been one of continuous transformation, driven by the need for higher data throughput, lower latency, and greater connectivity. From the analog foundations of 1G to the intelligent, task-aware networks of 5G and beyond, each generation has introduced core technical shifts that redefined how humans and machines communicate.

The first generation (1G) of mobile networks, which emerged in the late 1970s and 1980s, was characterized by analog voice transmission. Technologies such as the Advanced Mobile Phone System (AMPS) [20] in the United States, the Nordic Mobile Telephone (NMT) [21] in Scandinavia, and the Total Access Communication System (TACS) [22] in the United Kingdom were among the early implementations. These systems operated primarily in the 800-900 MHz bands and utilized Frequency Division Multiple Access (FDMA) [23] to allocate separate frequency bands to individual users. While these analog systems enabled mobile telephones for the first time, they suffered from poor spectral efficiency, high levels of noise interference, and an absence of encryption or secure handover mechanisms, making them highly vulnerable and inefficient by modern standards [24].

The transition to second-generation (2G) mobile systems in the early 1990s marked a significant breakthrough in adopting digital modulation techniques. The Global System for Mobile Communications (GSM) [24], developed in Europe, became the dominant 2G standard worldwide. Meanwhile, IS-95, based on Code Division Multiple Access (CDMA) [25], gained traction in North America and parts of Asia. These systems offered more efficient bandwidth utilization and supported services beyond voice, most notably Short Message Service (SMS) [26] and basic packet-switched data. GSM operated in the 900 and 1800 MHz frequency bands and utilized Gaussian Minimum Shift Keying (GMSK) [27] for modulation, while IS-95 introduced spread-spectrum techniques. The early 2G systems offered data rates of up to 14.4 kbps, which was extended to more than 384 kbps through the General Packet Radio Service (GPRS) [28] and the Enhanced Data Rates for GSM Evolution (EDGE) [29], often referred to as 2.5G and 2.75G, respectively. Additionally, 2G systems introduced basic Subscriber Identity Module (SIM)-based authentication and encryption, improving the security and privacy of mobile users [30].

The arrival of third-generation (3G) networks in the early 2000s addressed the increasing demand for mobile internet access and multimedia services such as email, video calling, and mobile web browsing. The Universal Mobile Telecommunications System (UMTS) [31], based on Wideband Code Division Multiple Access (WCDMA) [32], became the predominant 3G standard. It operated in the 2.1 GHz frequency band and utilized a 5 MHz channel bandwidth, enabling simultaneous support for voice and high-speed data. 3G systems

introduced adaptive modulation schemes such as Quadrature Phase Shift Keying (QPSK) [33] and, later, 16- and 64-Quadrature Amplitude Modulation (QAM) through technologies such as High Speed Packet Access (HSPA+) [34]. These enhancements pushed downlink data rates to several megabits per second under favorable conditions. Technical innovations such as rake receivers, soft handoff, and packet switching made 3G much more versatile and robust than its predecessors. Most importantly, it laid the foundation for an IP architecture, which would become central in subsequent generations [35], [36].

Building on this momentum, fourth-generation (4G) networks – led by Long-Term Evolution (LTE) [37] – initiate the era of high-speed broadband mobility. Launched in the 2010s, LTE eliminated circuit-switched architecture in favor of a fully packet-switched, IP-based system, significantly reducing latency and increasing spectral efficiency. LTE operated over a broad frequency range (from 700 MHz to 2.6 GHz) and employed Orthogonal Frequency Division Multiple Access (OFDMA) [38] in the downlink and Single Carrier-Frequency Division Multiple Access (SC-FDMA) [39] in the uplink. Through advanced modulation schemes (e.g., QPSK, 16-QAM, and 64-QAM) and technologies such as Multiple-Input Multiple-Output (MIMO) [40] and carrier aggregation, LTE supported peak downlink rates approaching 100 Mbps for mobile users and up to 1 Gbps in static environments. The evolution to LTE-Advanced and LTE-Advanced Pro further improved these rates and enabled features such as Voice over LTE (VoLTE) and IPv6 support, transforming smartphones into full-fledged multimedia devices and paving the way for real-time applications such as high-definition video conferencing and cloud-based gaming [41], [42].

The most recent advancement, fifth-generation (5G) networks, represents a paradigm shift not just in speed but also in network intelligence and flexibility. Standardized through 3GPP Releases 15 and beyond, 5G networks are designed to serve a diverse range of applications through three main service categories: enhanced Mobile Broadband (eMBB) [43], Ultra-Reliable Low-Latency Communications (URLLC) [44], and massive Machine-Type Communications (mMTC) [45]. These use cases target everything from immersive augmented reality and 4K/8K video streaming to autonomous driving, industrial automation, and the Internet of Things (IoT). Technically, 5G networks operate over sub-6 GHz bands and millimeter wave (mmWave) frequencies (above 24 GHz), enabling extremely high data rates and ultra-low latency. With channel bandwidths reaching up to 400 MHz, massive MIMO with beamforming, and adaptive modulation schemes up to 256-QAM, 5G promises peak data rates of 10 Gbps and latency as low as 1 millisecond. Additionally, network slicing and Multi-access Edge Computing (MEC) [46] allow the creation of virtualized network environments optimized for different tasks, supporting heterogeneous services with stringent performance requirements [47].

Looking ahead, sixth-generation (6G) mobile networks are envisioned not merely as faster or more capacious versions of 5G, but as fundamentally transformative infrastructures that integrate communication, sensing, computing, and intelligence into a unified framework. According to the International Telecommunication Union (ITU) and the major 6G research initiatives worldwide, including the Hexa-X project in Europe, the Next G Alliance in North

America, and China's 6G Innovation Hub, 6G will operate in terahertz (THz) frequency bands, potentially offering data rates exceeding 1 Tbps, submillisecond latency, and high-precision localization capabilities [48], [49]. Key pillars of 6G research include ubiquitous AI integration, Intelligent Reflecting Surfaces (IRS), cell-free massive MIMO, and native support for eXtended Reality (XR) applications and holographic communications [50], [51]. In addition, 6G is expected to mark a paradigm shift from data-centric to task-driven communications, where the network not only transmits information, but also understands and optimizes for the task to be performed. In this context, semantic communication, which aims to transmit meaning rather than raw bits, and task-oriented communication, which focuses on delivering just enough information to complete a downstream AI task, have emerged as foundational concepts.

Together, these generational advancements illustrate a clear trajectory: from basic voice communication to intelligent, context-aware systems designed for task-oriented, ultra-reliable, and low-latency applications. This evolution sets the stage for the paradigm explored in this thesis, task-oriented communication for edge intelligence enabled connected robotics systems, where communication is no longer just about data fidelity, but about precisely delivering the information needed to perform autonomous actions.

2.1.4 Emergence of AI-Driven Systems and the Need for New Communication Paradigms

The past decade has witnessed a profound transformation in the architecture and functionality of intelligent systems. With the maturation of deep learning, the growth of IoT devices, and the advent of edge computing, we are entering an era in which autonomous systems are expected to understand, decide, and act on sensory data in real time, ranging from self-driving vehicles and drones to smart factories and augmented reality platforms. This shift toward AI-driven autonomy challenges the foundational assumptions of conventional communication paradigms, where the main goal has historically been the faithful reconstruction of transmitted signals, independent of the task for which the data are ultimately used.

In traditional communication systems based on Shannon's theory, the network is modeled as a pipe for delivering bits from the sender to the receiver with minimal distortion and delay. Information is encoded, transmitted, and decoded with the objective of reconstructing the original signal as accurately as possible. This design philosophy works well when the recipient is a human or a general-purpose computing system tasked with storing or rendering the data. However, in many modern Machine-to-Machine (M2M) applications, the end goal is not to reconstruct the data per se, but rather to make a task-specific decision based on them, such as steering an autonomous vehicle, identifying an object in a video stream, or adjusting parameters in an industrial controller [52].

This task-centric nature of intelligent systems creates a mismatch between current communication methods and the actual performance requirements of the applications they support. For example, let's consider a drone navigating through a dense urban environment. It may

transmit high-resolution video to an edge server for object detection and path planning. Traditional communication methods aim to deliver the full image with minimal loss, consuming bandwidth and incurring latency. However, the drone's control system only requires high-confidence information about the location of obstacles or targets, not pixel-perfect reconstructions. In this case, transmitting task-irrelevant data is inefficient and even counterproductive in latency-constrained environments [14].

To meet the needs of such applications, the computational paradigm is shifting toward edge intelligence, a model that distributes processing closer to the data source (i.e., at the edge of the network). This model enables low-latency inference and decision-making by allowing data to be compressed, encoded, and interpreted locally, before being transmitted to centralized servers for further processing if needed [53]. However, even in edge-enabled architectures, communication remains a bottleneck. Wireless links are bandwidth-limited and prone to interference, while AI models are increasingly data-hungry. This has led to a growing recognition that communication itself must be co-optimized with computation and control, particularly in latency-sensitive domains such as autonomous driving, remote robotics, and AR/VR [54].

In response to these demands, researchers have begun to explore new communication paradigms, which seek to transmit only the information necessary for a given downstream task. A key development in this area is the concept of semantic communication, which aims to preserve the meaning of the transmitted information rather than its raw form. Unlike traditional communication, where every bit contributes equally to the reconstruction objective, semantic communication prioritizes semantically relevant features, often derived from DNN encoders, to reduce redundancy and enhance efficiency [55]. Another emerging and related direction is task-oriented communication, which formalizes this idea through information-theoretic frameworks such as the IB method [10]. Here, the encoder is optimized not to preserve all information in the input, but only the information that is maximally informative about the task output, such as classification labels or control actions.

These new paradigms fundamentally redefine the role of communication in AI-driven systems, from being a neutral data transporter to an active participant in task execution. In semantic and task-oriented communication, the system is designed not only to transmit but to understand what needs to be transmitted, depending on context, task criticality, and network constraints. Importantly, these approaches are especially valuable in edge-enabled CPS, where data must be processed and acted upon within tight latency and reliability bounds.

This thesis builds on these developments, proposing a set of novel architectures that unify task-oriented communication with edge intelligence, deep JSCC, and autonomous control. The goal is to design cooperative systems that selectively transmit mission-critical information, ensuring robust inference and decision making even under challenging network conditions such as low SNR, high mobility, and stringent delay budgets.

2.2 Semantic Communication

As modern communication systems shift focus from human-centric applications to machine-centric, task-driven services, a critical question arises: Is it always necessary to transmit all the data for effective communication, or is it sufficient to convey only what is meaningful for a given task? The concept of semantic communication emerges from this question, representing a fundamental evolution in how information is encoded, transmitted, and interpreted. Rather than aiming for bit-level fidelity, semantic communication seeks to preserve the meaning of transmitted content, aligning the communication process more closely with cognitive and contextual understanding.

2.2.1 Rethinking Communication Models

In Shannon's model, information is abstracted as a stream of bits, and the goal is to maximize the transmission rate under the constraints of bandwidth and noise. However, this model disregards the actual content or meaning of the message. For example, the phrases "It's raining" and "Rain is falling" may convey identical semantic content but differ significantly at the bit level. Semantic communication breaks from this constraint by shifting the performance objective from bit-wise accuracy to semantic fidelity.

The modern architecture of a semantic communication system includes semantic encoders and decoders built on DNNs, which are trained to capture and reconstruct meaning from data. These systems sometimes incorporate a shared Knowledge Base (KB) between the sender and the receiver, allowing them to interpret the same symbols in semantically aligned ways. The semantic encoder filters out redundant information and extracts meaningful features, while the semantic decoder reconstructs the transmitted message based on contextual and learned semantic representations.

2.2.2 Key System Components: A New Architecture

Semantic communication systems integrate several new elements beyond classical communication systems:

- **Semantic Encoder:** Extracts high-level semantic features from the source data using DNNs. Examples include LSTM and Transformer models, which map linguistic or visual inputs into compact semantic embeddings.
- **Channel Encoder/Decoder:** Manages the actual transmission over noisy channels, typically integrated into the deep learning pipeline as differentiable layers.
- **Semantic Decoder:** Reconstructs the semantic message using the received symbols and the KB, allowing for intelligent inference even under partial or distorted input.
- **Knowledge Base (optional):** A shared semantic framework that facilitates mutual understanding between sender and receiver, essential for tasks such as contextual reasoning and disambiguation.

This architecture supports both semantic-level communication, which focuses on meaning preservation, and effectiveness-level communication, which aligns communication with the success of a downstream task.

2.2.3 Semantic Metrics

Given that traditional metrics such as Bit-Error Rate (BER), Peak Signal-to-Noise Ratio (PSNR), or Structural Similarity (SSIM) do not capture the preservation of meaning, the development of semantically aware metrics has become essential for evaluating performance in semantic communication systems. These metrics fall broadly into three categories:

Semantic Similarity Metrics: These aim to quantify how well the semantics of the received message align with the original.

- **Bilingual Evaluation Understudy (BLEU):** Originally used in machine translation to assess the overlap between predicted and reference sentences. Useful for evaluating textual semantics.
- **BERTScore:** Leverages deep language models (e.g., BERT) to compute cosine similarity between embeddings of sentences. More context-sensitive than BLEU.
- **Fréchet Inception Distance (FID):** Commonly used in image tasks to compare distributions of high-level features between generated and ground-truth data.
- **Scene Graph Similarity:** For visual semantics, scene graphs (nodes = objects, edges = relations) allow for evaluating whether core entities and relationships are preserved.

Effectiveness-Oriented Metrics: Used primarily in effectiveness-level semantic communication systems, where the goal is task success rather than data recovery.

- **Task Accuracy:** Classification or control performance (e.g., object detection F1-score, autonomous driving collision rate) is used to indirectly measure the sufficiency of transmitted semantics.
- **Value of Information (VoI):** Evaluates how much a received message contributes to improving task outcomes or decisions.
- **Age of Information (AoI):** Measures the freshness of the data; especially relevant for real-time semantic updates in robotic and CPS systems.

Compression-Efficiency Metrics: Semantic communication also seeks transmission efficiency, balancing fidelity and bandwidth:

- **Semantic Compression Ratio:** The ratio of compressed semantic representation size to original data size.

- **Transmission Gain vs. Semantic Loss Trade-Off:** Explores the trade-off between fewer bits and decreased task/semantic accuracy.

2.3 Basic Information Theory

2.3.1 Asymptotic Equipartition Property

The Asymptotic Equipartition Property (AEP) is formalized as follows [56]:

Theorem 2.3.1 (AEP)**:** *If $X_1, X_2, \ldots$ are i.i.d.$\sim p(x)$, then*

$$-\frac{1}{n}\log p(X_1, X_2, \ldots, X_n) \to H(X) \quad \textit{in probability}. \tag{2.1}$$

Proof:

$$-\frac{1}{n}\log p(X_1, X_2, \ldots, X_n) = -\frac{1}{n}\sum_i \log p(X_i) \tag{2.2}$$

Since the functions of independent random variables are also independent random variables, according to the weak law of large numbers,

$$-\frac{1}{n}\sum_i \log p(X_i) \to -E\log p(X) \quad \text{in probability} \tag{2.3}$$

$$= H(X). \tag{2.4}$$

□

In particular, $p(X_1, X_2, \ldots, X_n)$ is close to $2^{-nH(X)}$ with a high probability, which can be stated as

$$\Pr\{(X_1, X_2, \ldots, X_n) : 2^{-n(H(X)+\epsilon)} \leq p(X_1, X_2, \ldots, X_n) \leq 2^{-n(H(X)-\epsilon)}\} \approx 1 \tag{2.5}$$

Definition 2.3.1 (Typical Set)**:** The typical set $A_\epsilon^{(n)}$ is a set of the sequence $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$ respect to $p(x)$ with the following property:

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}. \tag{2.6}$$

The previous work [56] had proven that the typical set has a probability close to 1, with all elements within the typical set being almost equally likely, and the count of elements in the typical set is approximately $2^{nH(X)}$. This work mainly focuses on the discussion of the typical set.

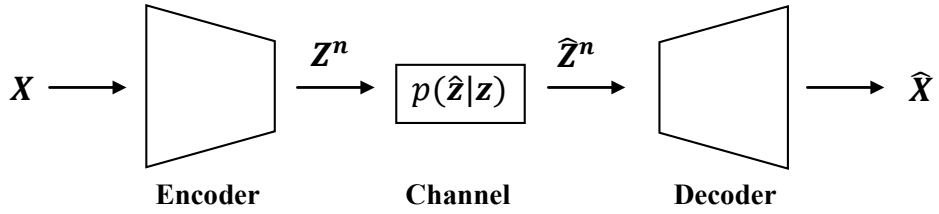


Figure 2.1: An example system framework of communication.

2.3.2 Channel Capacity

The primary objective of a communication system is to ensure that the transmitted information is accurately received despite the presence of noise and other channel impairments. A simplified model of a communication system is illustrated in Fig. 2.1. The encoder takes the source message X and encodes it into a sequence of channel symbols $Z^n = [Z_1, \cdots, Z_n]$, where n denotes the length of the symbol sequence. The channel symbols Z^n are then transmitted through the channel, resulting in the output sequence $\hat{Z}^n = [\hat{Z}_1, \cdots, \hat{Z}_n]$. The distribution of $\hat{Z}^n$ depends on the input sequence Z^n and the characteristics of the channel. From the output sequence $\hat{Z}^n$, the traditional communication aims to accurately reconstruct the original source message X.

According to Section 2.3.1, the number of elements in the typical set ($\hat{\mathcal{Z}}^n$) is about $2^{nH(\hat{Z})}$. Similarly, for each typical input sequence $(z_1, z_2, \ldots, z_n) \in \mathcal{Z}^n$, there are approximately $2^{nH(\hat{Z}|Z)}$ possible output sequences.

In order to distinguish the corresponding input sequence, the typical set of the output sequence $\hat{\mathcal{Z}}^n$ should be divided into sets of size $2^{nH(\hat{Z}|Z)}$ without overlapping. The number of divided sets is less than or equal to $2^{nH(\hat{Z})}/2^{nH(\hat{Z}|Z)} = 2^{n(H(\hat{Z})-H(\hat{Z}|Z))} = 2^{nI(Z;\hat{Z})}$, where $I(\cdot;\cdot)$ denotes mutual information. In that case, up to $2^{nI(Z;\hat{Z})}$ distinguishable sequences of length n can be transmitted without confusion. This perspective provides an intuitive sense of information *channel capacity*, which has the following formal definition [56].

Definition 2.3.2: The information channel capacity of a Discrete Memoryless Channel (DMC) is defined as

$$C = \max_{p(z)} I(Z; \hat{Z}), \tag{2.7}$$

where the maximum channel capacity is taken over all possible $p(z)$.

As shown in Fig. 2.1, assume that the message X is obtained from the index set $\{1, 2, \ldots, M\}$. Considering a DMC without feedback, we have the following definitions [56].

Definition 2.3.3: An (M, n) code for the DMC without feedback consists of the following:

1. An index set $\{1, 2, \ldots, M\}$.
2. An encoding function $f_{\text{Enc}} : \{1, 2, \ldots, M\} \to \mathcal{Z}^n$.
3. A decoding function $f_{\text{Dec}} : \hat{\mathcal{Z}}^n \to \{1, 2, \ldots, M\}$.

Definition 2.3.4: We define the probability of error of the given sent index i as

$$\lambda_i = \Pr(f_{\text{Dec}}(\hat{Z}^n) \neq i \mid Z^n = f_{\text{Enc}}(i)). \tag{2.8}$$

Definition 2.3.5: We define the maximum probability of error for an (M, n) code as

$$\lambda_{\max} = \max_{i \in \{1,2,\ldots,M\}} \lambda_i. \tag{2.9}$$

Definition 2.3.6: The rate R of an (M, n) code is

$$R = \frac{\log M}{n} \quad \text{bits per transmission.} \tag{2.10}$$

Definition 2.3.7: If there exists a sequence of $\left(\left\lceil 2^{nR} \right\rceil, n\right)$ that $\lambda_{\max} \to 0$ as $n \to \infty$, the rate R is said to be achievable.

Definition 2.3.8: The capacity of DMC is the upper bound of all achievable rates.

2.3.3 Rate-Distortion and Distortion-Rate Theory

Since using a finite rate to precisely describe a random information source X with infinite precision is a challenge, the actual problem is to find the optimal representation under a given data rate. Let X_q denote the quantized codebook of the information source X, which can be characterized by a conditional probability $p(x|x_q)$. As discussed in Section 2.3.1, the number of elements in the space of X is $2^{H(x)}$. In addition, the average number of elements in the space of X that can be mapped to the identical element in the space of X_q is $2^{H(x|x_q)}$. In that case, the elements in the space of X can be divided with the average elements of $2^{H(X)}/2^{H(X|X_q)} = 2^{H(X)-H(X|X_q)} = 2^{I(X;X_q)}$, where the mutual information $I(X; X_q)$ is an indicator for the quality of quantization.

In rate-distortion theory [6], [56], this problem is addressed by defining a distortion function $d : \mathcal{X} \times \mathcal{X}_q \to \mathbb{R}^+$, where a smaller value of the distortion function indicates a more accurate representation.

The expected distortion is defined as

$$\mathbb{E}_{x,x_q}[d(x, x_q)] = \sum_{x \in \mathcal{X}} p(x) \sum_{x_q \in \mathcal{X}_q} p(x_q|x) d(x, x_q). \tag{2.11}$$

In order to minimize the rate while satisfying the expected distortion boundary D, the rate-distortion function of an i.i.d. information source X with distribution $p(x)$ characterizes this trade-off via a rate-distortion function $R(D)$:

$$R(D) = \min_{\{p(x_q|x):\mathbb{E}_{x,x_q}[d(x,x_q)] \leq D\}} I(X; X_q). \tag{2.12}$$

Introducing the Lagrange multiplier β, we can instead minimize the following function:

$$\mathcal{L}_{R(D)}[p(x_q|x)] = I(X;X_q) + \beta \mathbb{E}_{x,x_q}[d(x,x_q)]. \tag{2.13}$$

Similarly, in order to minimize the expected distortion while satisfying the rate constraint R, we can formulate the distortion-rate function $D(R)$:

$$D(R) = \min_{\{p(x_q|x):I(X;X_q)\leq R\}} \mathbb{E}_{x,x_q}[d(x,x_q)]. \tag{2.14}$$

Introducing the Lagrange multiplier β, we can instead minimize the following function:

$$\mathcal{L}_{D(R)}[p(x_q|x)] = \mathbb{E}_{x,x_q}[d(x,x_q)] + \beta I(X;X_q). \tag{2.15}$$

Traditional data compression and communication research primarily focus on the rate-distortion function (Eq. (2.12)), which characterizes the trade-off between compression efficiency and reconstruction fidelity. This principle is widely applied in areas such as image compression, video streaming, and signal processing, where the goal is to minimize the required transmission rate while maintaining an acceptable level of distortion.

On the other hand, the distortion-rate function (Eq. (2.14)) is particularly well-suited for neural network-based research, where the structure of the network imposes a fixed capacity constraint on intermediate representations. In this setting, rather than minimizing the rate for a given distortion threshold, the objective shifts to minimizing performance loss (D) under a predefined neural network architecture (R). This is especially relevant in applications such as knowledge distillation, feature compression, and information bottleneck theory, where preserving the most critical information within a limited representation space is crucial for maintaining model performance.

2.4 Information Bottleneck Approach

2.4.1 Information Bottleneck

The rate-distortion theory provides a fundamental perspective on data compression by balancing information rate and fidelity. However, defining an appropriate distortion function that generalizes across different data types remains a challenge. Since distortion is typically determined by the Key Performance Indicators (KPIs) of a given task, a metric that works well in one scenario may fail in another. A promising approach to addressing this issue is to evaluate distortion based on the information conveyed rather than on high-fidelity reconstruction.

Considering a Directed Probabilistic Graphical Model (DPGM)

$$A \to X \to X_q, \tag{2.16}$$

where A denotes a relevant variable to X. In this case, we are interested in A, so that we want the quantized variable X_q to maintain maximum information about A. The amount of information about A in X_q is expressed by

$$I(A;X_q) = \sum_{a\in\mathcal{A}} \sum_{x_q\in\mathcal{X}_q} p(a,x_q) \log \frac{p(a,x_q)}{p(a)p(x_q)}. \tag{2.17}$$

According to the data processing inequality, we have $I(A;X) \geq I(A;X_q)$, which indicates that quantization cannot increase the relative information of A in X_q in statistics. Instead of designing a distortion function $d(\cdot,\cdot)$ to measure the difference between X and X_q, we would like to shift the objective of Eq. (2.14) to retain the maximum information about A in X_q subject to the rate constraint. This optimization problem can be formulated as

$$\begin{aligned} \min_{p(x_q|x)} \quad & -I(A;X_q) \\ \text{s.t.} \quad & I(X;X_q) - R \leq 0, \end{aligned} \tag{2.18}$$

where the 'bottleneck' is the process from X to X_q. Introducing the Lagrange multiplier β, we can instead minimize the following function:

$$\mathcal{L}_{\text{IB}}[p(x_q|x)] = -I(A;X_q) + \beta I(X;X_q). \tag{2.19}$$

The IB theory (Eq. (2.18)), which extends from the foundational rate-distortion theory [56], aims to find an optimal trade-off by maximizing the preservation of task-specific information in the latent representations, while minimizing the inclusion of task-agnostic information from the input data. Initially proposed by [6], the practical application of IB theory in training deep neural networks remained theoretical until significantly later [9].

2.4.2 Variational Information Bottleneck

The application of IB theory in deep learning was primarily hindered by computational challenges. The traditional optimization of the IB objective function relied on the iterative Blahut-Arimoto algorithm [7], [8], which is infeasible for deep learning applications due to its computational complexity and inefficiency in handling large-scale data [9]. Addressing this limitation, [10] introduced a variational approach to construct a tractable lower bound on the IB objective, leading to the development of the VIB method. This approach enabled the practical application of the IB principles in deep learning by approximating the intractable true posterior with a variational distribution.

Although Section 2.4.1 identifies the quantization as the 'bottleneck', it can be more broadly understood as an encoding process for information source X, leading to the following DPGM

$$A \to X \to Z, \tag{2.20}$$

where Z denotes the encoded data. The encoding process $X \to Z$ is defined by a parametric encoder $p_\phi(z|x)$. Thus, we can construct an objective as

$$\begin{aligned} \min_{\phi} \quad & -I(A;Z) \\ \text{s.t.} \quad & I(X;Z) - R \leq 0. \end{aligned} \tag{2.21}$$

Introducing the Lagrange multiplier β, we can instead minimize the following function:

$$\mathcal{L}_{\text{IB}}(\phi) = -I(A;Z) + \beta I(X;Z). \tag{2.22}$$

With the objective function Eq. (2.22), we illustrate how to compute each term in turn. We start with $-I(A;Z)$, which can be written as

$$\begin{aligned} -I(A;Z) &= -\int p(a,z) \log \frac{p(a,z)}{p(a)p(z)} \,\mathrm{d}a\,\mathrm{d}z && (2.23)\\ &= -\int p(a,z) \log \frac{p(a|z)}{p(a)} \,\mathrm{d}a\,\mathrm{d}z && (2.24)\\ &= -\int p(a,z) \log p(a|z) \,\mathrm{d}a\,\mathrm{d}z + \int p(a,z) \log p(a) \,\mathrm{d}a\,\mathrm{d}z && (2.25)\\ &= -\int p(a,z) \log p(a|z) \,\mathrm{d}a\,\mathrm{d}z - H(A), && (2.26) \end{aligned}$$

where $p(a|z)$ is the posterior probability, which can be derived through the DPGM (Eq. (2.20)) as

$$\begin{aligned} p(a|z) &= \int p(a,x|z) \,\mathrm{d}x && (2.27)\\ &= \int p(x|z)p(a|x) \,\mathrm{d}x && (2.28)\\ &= \int \frac{p(x)p_\phi(z|x)p(a|x)}{p(z)} \,\mathrm{d}x. && (2.29) \end{aligned}$$

Given the complexity of this integration, let $q_\psi(a|z)$ be a variational approximation to $p(a|y)$.

According to the definition of KL divergence D_{KL} [56], we can derive the following expression:

$$\begin{aligned} D_{\text{KL}}(p(a|z) \| q_\psi(a|z)) &= \int p(a,z) \log \frac{p(a|z)}{q_\psi(a|z)} \,\mathrm{d}a\,\mathrm{d}z && (2.30)\\ &= \int p(a,z) \log p(a|z) \,\mathrm{d}a\,\mathrm{d}z - \int p(a,z) \log q_\psi(a|z) \,\mathrm{d}a\,\mathrm{d}z. && (2.31) \end{aligned}$$

Since the KL divergence is always non-negative, we have:

$$\int p(a,z) \log p(a|z) \,\mathrm{d}a\,\mathrm{d}z \geq \int p(a,z) \log q_\psi(a|z) \,\mathrm{d}a\,\mathrm{d}z, \tag{2.32}$$

which derives

$$-I(A;Z) \leq -\int p(a,z)\log q_\psi(a|z)\,\mathrm{d}a\,\mathrm{d}z - H(A). \tag{2.33}$$
$$= -\int p(a,x,z)\log q_\psi(a|z)\,\mathrm{d}a\,\mathrm{d}x\,\mathrm{d}z - H(A) \tag{2.34}$$
$$= -\int p(a,x)p_\phi(z|x)\log q_\psi(a|z)\,\mathrm{d}a\,\mathrm{d}x\,\mathrm{d}z - H(A) \tag{2.35}$$
$$= \mathbb{E}_{a,x}\left[\mathbb{E}_{z|x;\phi}\left[-\log q_\psi(a|z)\right]\right] - H(A) \tag{2.36}$$

Note that the entropy $H(A)$ is independent of the optimization and thus can be ignored.

The second term $I(X;Z)$ can be formulated as:

$$I(X;Z) = \int p(x,z)\log\frac{p(x,z)}{p(x)p(z)}\,\mathrm{d}x\,\mathrm{d}z \tag{2.37}$$
$$= \int p(x,z)\log\frac{p_\phi(z|x)}{p(z)}\,\mathrm{d}x\,\mathrm{d}z \tag{2.38}$$
$$= \int p(a,x,z)\frac{p_\phi(z|x)}{p(z)}\,\mathrm{d}a\,\mathrm{d}x\,\mathrm{d}z \tag{2.39}$$
$$= \int p(a,x)p_\phi(z|x)\frac{p_\phi(z|x)}{p(z)}\,\mathrm{d}a\,\mathrm{d}x\,\mathrm{d}z \tag{2.40}$$
$$= \mathbb{E}_{a,x}\left[D_{\mathrm{KL}}(p_\phi(z|x)\|p(z))\right], \tag{2.41}$$

where $p(z)$ is the intractable prior probability of z. Let $q_\epsilon(z)$ be the variational approximation of $p(z)$. Since $D_{\mathrm{KL}}(p(z)\|q_\epsilon(z)) \geq 0$, we have

$$\int p(z)\log p(z)\,\mathrm{d}z \geq \int p(z)\log q_\epsilon(z)\,\mathrm{d}z. \tag{2.42}$$

So that

$$I(X;Z) \leq \int p(x,z)\log\frac{p_\phi(z|x)}{q_\epsilon(z)}\,\mathrm{d}x\,\mathrm{d}z \tag{2.43}$$
$$= \int p(a,x,z)\frac{p_\phi(z|x)}{q_\epsilon(z)}\,\mathrm{d}a\,\mathrm{d}x\,\mathrm{d}z \tag{2.44}$$
$$= \int p(a,x)p_\phi(z|x)\frac{p_\phi(z|x)}{q_\epsilon(z)}\,\mathrm{d}a\,\mathrm{d}x\,\mathrm{d}z \tag{2.45}$$
$$= \mathbb{E}_{a,x}\left[D_{\mathrm{KL}}(p_\phi(z|x)\|q_\epsilon(z))\right]. \tag{2.46}$$

Combining Eq. (2.36) and Eq. (2.46), the upper bound of the Eq. (2.22) is given by

$$\mathcal{L}_{\mathrm{VIB}}(\phi) = \mathbb{E}_{a,x}\left[\mathbb{E}_{z|x;\phi}\left[-\log q_\psi(a|z)\right] + D_{\mathrm{KL}}(p_\phi(z|x)\|q_\epsilon(z))\right] \tag{2.47}$$
$$\geq \mathcal{L}_{\mathrm{IB}}(\phi) + H(A) \tag{2.48}$$
$$= -I(A;Z) + \beta I(X;Z) + H(A), \tag{2.49}$$

which can be optimized using stochastic gradient descent through Monte Carlo sampling, providing a practical framework for empirical estimation and subsequent optimization.

Recent work has seen the integration of VIB with deep JSCC, which has been effectively used to formalize task-oriented communication strategies. In particular, the results [11], [12] have demonstrated that combining VIB with deep JSCC offers superior performance over reconstruction-oriented communication frameworks. These studies showcase the potential of VIB in improving the efficiency and robustness of communication systems, particularly in scenarios where preserving task-specific information and discarding task-agnostic information are crucial.

Integrating JSCC and IB methods to protect user privacy is an advanced direction in current research. FedSem [57] had collaboratively trained semantic-channel encoders of multiple devices coordinated by a semantic-channel decoder using IB theory based on base stations. Unlike traditional centralized learning approaches, FedSem reduces communication overhead and mitigates privacy concerns by enabling the sharing of semantic features rather than raw data. In addition, the author of [58] introduced a privacy-preserving JSCC scheme for image transmission, using a disentangled IB objective to effectively separate private information from public data. This approach ensures the protection of privacy-sensitive information while maintaining high image quality. Although these works show impressive progress in the integration of JSCC with IB theory, they often require specialized designs that are challenging to combine with existing systems and devices.

There is a need to design an advanced framework aligning two communication paradigms – task-oriented communications and reconstruction-oriented communications – and develop a JSCC modulation scheme for practical deployment.

2.5 Joint Source-Channel Coding

2.5.1 Introduction and Motivation

Communication systems traditionally follow the principles established by Claude Shannon, notably the source-channel separation theorem. According to Shannon's theorem, optimal communication performance can theoretically be achieved by independently optimizing source coding (data compression) and channel coding (error correction) under the assumptions of infinite block length and stationary, memoryless channels [16]. This separation principle has significantly shaped the design of modern communication networks due to its modular simplicity, allowing separate advancements in source compression methods such as JPEG for images and MP3 for audio, and channel coding techniques such as LDPC or turbo codes for error correction [59].

Despite the practical advantages offered by source-channel separation, such as modularity and ease of standardization, several limitations arise under realistic operating conditions. One notable limitation is the suboptimal performance in scenarios involving finite block lengths, which is typical in real-time and latency-critical applications [60]. When the block length is finite, separation-based approaches may not fully exploit the potential capacity of a given channel, often leading to degraded performance and inefficiencies. Furthermore,

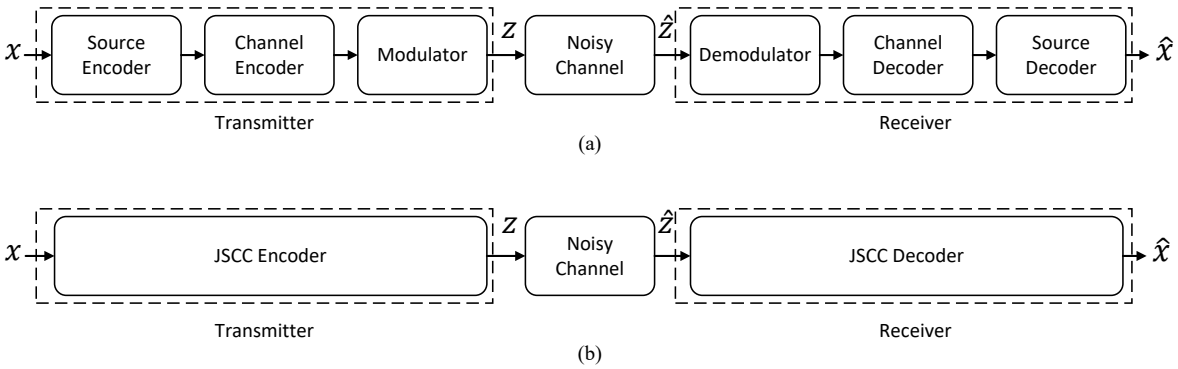


Figure 2.2: Examples of the basic point-to-point communication systems: (a) the traditional communication system and (b) the joint source-channel communication system.

the separation theorem assumes perfect knowledge of channel conditions, which is rarely achievable in dynamic wireless environments that vary in time such as vehicular networks, drone communications, and IoT ecosystems [61]. Under rapidly fluctuating channel conditions, separate encoding and decoding schemes frequently suffer from the "cliff effect," where system performance drastically deteriorates when channel quality falls below certain thresholds [62].

Joint Source-Channel Coding (JSCC) emerges as an alternative communication paradigm aimed at addressing these limitations. JSCC involves jointly optimizing the processes of source compression and channel coding in a single integrated encoding and decoding framework, as shown in Fig. 2.2. Unlike separate coding, JSCC does not explicitly compress the data to a minimal representation before transmission; instead, it directly maps the source information to the input signals of the channel, optimizing the performance for the specific characteristics of both the source and the channel simultaneously [63]. This joint optimization allows JSCC to inherently adapt to channel conditions, ensuring robustness against varying noise levels and preventing the cliff effect commonly observed in separation-based systems [64].

Recently, the importance of JSCC has been amplified due to the rise of semantic and task-oriented communications. These modern communication paradigms focus on efficiently transmitting task-specific or semantic information rather than faithfully reconstructing the original transmitted signals. For example, in applications such as autonomous driving, Augmented Reality (AR), Virtual Reality (VR), and remote sensing, it is often unnecessary or inefficient to reconstruct all transmitted data exactly. Instead, it is more beneficial to transmit only the relevant information required to perform specific downstream tasks such as detection, classification, or decision making [14], [65]. In such scenarios, JSCC demonstrates significant benefits by inherently focusing on the relevance of the transmitted information and optimizing directly for task performance, resulting in substantial reductions in latency and bandwidth requirements compared to conventional systems.

Given these emerging trends and practical considerations, this section introduces the fundamental principles, practical implementations, and recent advancements in JSCC, particularly emphasizing its utility in modern semantic and task-oriented communication scenarios.

2.5.2 Theoretical Foundations of JSCC

The theoretical foundations of JSCC are rooted in the study of the source-channel communication problem by information theory, where the goal is to reconstruct a source signal over a noisy channel with the required fidelity. At the heart of classical information theory is Shannon's separation theorem, which posits that separate optimization of source and channel codes does not incur any performance loss, provided the blocklength is infinite and the source and channel statistics are stationary and known. Under these conditions, the channel capacity C (Eq. (2.7)) and the rate-distortion function $R(D)$ (Eq. (2.12)) fully characterize the system's performance: reliable communication is achievable if and only if $R(D) \leq C$ [66].

However, in many real-world applications, especially in wireless communication systems, these ideal assumptions do not hold. That is, in practical settings characterized by finite blocklength, time-varying channels, and stringent latency constraints, the optimality of source–channel separation collapses. The work [60] established finite blocklength bounds that reveal a non-negligible penalty when operating at short delays, leading to performance degradation in separate source-channel coding schemes. In these regimes, JSCC becomes advantageous by jointly optimizing encoding strategies for both source characteristics and channel impairments, achieving improved rate-distortion performance and robustness.

The benefits of JSCC also become evident in scenarios involving continuous or analog sources transmitted over memoryless Gaussian channels. The seminal work [67] showed that a linear mapping, essentially an analog transmission scheme, is optimal to transmit a memoryless Gaussian source over an AWGN channel with mean squared error distortion, thus illustrating a case where separation is not required. Such examples indicate that, under certain distortion measures and source-channel matching conditions, JSCC can achieve optimality with simpler uncoded transmission schemes.

Furthermore, in multi-user or multi-terminal scenarios such as the multiple access channel, broadcast channel, and relay networks, the separation theorem generally fails. For example, in distributed source coding over a multiple access channel, correlated sources must be encoded jointly to effectively exploit source correlations [68]. Similarly, for the transmission of correlated sources over a broadcast channel, a separate design leads to suboptimal solutions due to mismatched coding strategies and inefficient resource use [69]. JSCC enables correlated encoding and decoding strategies that align with network constraints, yielding performance gains in both efficiency and fidelity.

From an optimization perspective, JSCC can be formulated as a joint minimization problem in the encoder-channel-decoder system to avoid expected distortion. While this problem is nonconvex and generally intractable in closed form, advances in information-theoretic bounds and iterative optimization techniques have provided insight into the structure of near-optimal joint designs. In addition, hybrid digital-analog schemes represent a practical design approach that combines the robustness of analog transmission with the efficiency of digital codes, especially useful in time-varying channels where Channel State Information (CSI) may be imperfect or outdated [14].

2.5.3 JSCC for Semantic Communications

JSCC provides substantial benefits for semantic and task-oriented communications by directly optimizing the transmitted information for specific tasks or meaningful interpretations rather than precise signal reconstruction. This section introduces the applications of JSCC in three primary areas: text, audio, and image/video communication.

Text Communication

In semantic text communication, the focus shifts from the exact reproduction of transmitted bitstreams to the accurate transmission of intended meaning. Traditional separate source and channel coding systems, which tokenize and encode text before applying error-correcting codes, often fail under noisy conditions, leading to significant semantic degradation. In contrast, JSCC approaches, particularly those leveraging deep learning, offer robust alternatives by jointly optimizing the encoding and transmission processes to preserve semantic integrity even in adverse channel conditions.

[70] introduced a variable-length JSCC scheme for text using deep learning, which adapts the encoding length based on the sentence structure and content. This method improves the efficiency and reliability of text transmission over noisy channels by dynamically adjusting to the semantic complexity of the input.

In addition, [71] proposed the Iterative Semantic Joint Source-Channel Coding (IS-JSCC), a semi-neural framework designed specifically for text communication over wireless channels. Unlike traditional neural network-based JSCC, IS-JSCC iteratively refines semantic decoding by using intermediate decoded text as prior knowledge in subsequent decoding iterations. The semantic information of the candidate words is synthesized in the embedding space, weighted by posterior probabilities, thus effectively reducing the spread of errors and improving the robustness against varying channel conditions [71]. This iterative approach demonstrates superior performance over fully neural end-to-end models in text reconstruction quality, particularly in dynamic wireless environments.

Moreover, semantic communication systems for text transmission use advanced deep learning models, notably Transformers [72], to extract and encode semantic meaning directly from textual content. The DeepSC system developed by [73] is a significant advancement in this direction. This system utilizes a Transformer-based neural network architecture to perform JSCC, prioritizing the preservation of semantic meaning over traditional bit-error metrics. DeepSC employs transfer learning to rapidly adapt to various communication environments, maintaining performance under challenging low SNR conditions. By optimizing the semantic accuracy of reconstructed sentences rather than individual symbols or bits, DeepSC demonstrates enhanced robustness against channel noise and distortions.

The other notable example is the Transformer-based JSCC framework proposed by [74], which employs advanced natural language processing techniques to model and encode sentences. This system utilizes a Transformer encoder to extract semantic features from tokenized text, which are then quantized into fixed-length binary sequences for transmission. Upon

reception, a Transformer decoder reconstructs the sentences from these sequences. The framework demonstrates superior performance in maintaining semantic similarity and translation accuracy over various noisy channels, including binary erasure and deletion channels.

Audio Communication

In semantic audio communication, the emphasis is on preserving the comprehensibility and meaning of speech rather than achieving perfect waveform reconstruction. This approach is particularly beneficial in applications such as voice-controlled systems, telemedicine, and emergency communications, where understanding the message conveyed is paramount.

Recent work [75] proposed DeepSC-S, a semantic communication system specifically designed for deep learning-based speech signals. DeepSC-S incorporates a squeeze-and-excitation network, which leverages an attention mechanism that identifies and emphasizes essential speech features. This attention-based approach allows DeepSC-S to assign higher weights to critical semantic components, thereby enhancing the accuracy of the reconstruction of audio signals. The system demonstrates superior performance in various channel conditions without retraining and exhibits enhanced robustness, particularly under low SNR regimes, making it suitable for applications such as telephone systems and multimedia transmissions.

A notable advancement in this domain is the development of DeepSC-ST [76], a deep learning-based semantic communication system designed for speech transmission. This system integrates speech recognition and synthesis tasks, enabling the extraction of semantic features from speech inputs, which are then transmitted over the channel. At the receiver end, the system reconstructs the speech using the recognized text and speaker information. This approach significantly reduces the amount of data transmitted without compromising performance, particularly excelling in low SNR scenarios.

Another significant contribution is the low-latency deep JSCC framework for speech transmission over analog Gaussian wireless channels [77]. This system employs a deep neural network that performs joint source-channel encoding and decoding, facilitating real-time speech communication with minimal latency. The design is particularly suited for applications requiring ultra-low delay, such as hearing aids and live broadcasting, demonstrating superior performance over traditional methods in terms of speech quality and intelligibility under low-latency constraints.

Furthermore, JSCC has leveraged the predictive capabilities of large language models (LLMs) to create resilient audio transceivers. The SoundSpring transceiver, introduced by [78], utilizes dual-functional masked language modeling to achieve high audio compression efficiency while maintaining robustness against packet loss. This system employs residual vector quantization (RVQ) to encode latent features into tokens, which are contextually modeled by masked language models (MLMs). These MLMs serve dual functions: optimizing entropy coding efficiency during transmission and performing robust packet loss concealment at the receiver. Extensive experiments show that SoundSpring significantly outperforms traditional and neural audio codecs under various packet-loss conditions, providing improved signal fidelity and perceptual audio quality.

Image/Video Communication

Traditional separate source and channel coding methods, which involve compressing images using codecs like JPEG or H.264 followed by channel coding, often suffer from the "cliff effect," where minor degradations in channel quality can lead to significant drops in reconstruction fidelity. Deep learning-based JSCC approaches address this limitation by jointly optimizing encoding and transmission processes, resulting in more robust and efficient communication systems.

A seminal work in this area is the DeepJSCC framework introduced by [14], which employs CNNs to directly map image pixel values to channel input symbols. This end-to-end learning approach eliminates the need for explicit source and channel coding, demonstrating superior performance over traditional methods, especially in SNR scenarios and under varying channel conditions.

Building on this foundation, [79] proposed DeepJSCC-f, an extension that incorporates channel output feedback into the JSCC framework. Using feedback information, DeepJSCC-f enhances image reconstruction quality and reduces transmission latency, showcasing the benefits of integrating feedback mechanisms into deep learning-based communication systems.

Further advances have been made in adapting JSCC for video transmission. [80] developed DeepWiVe, an end-to-end JSCC system for wireless video transmission that combines video compression, channel coding, and modulation into a single neural network. DeepWiVe introduces a reinforcement learning-based bandwidth allocation strategy, optimizing the distribution of limited channel resources among video frames to maximize overall visual quality. This approach outperforms traditional video compression methods like H.264 and H.265 when combined with channel coding, particularly in dynamic channel environments.

The work [81] introduced two innovative JSCC frameworks: InverseJSCC and GenerativeJSCC, specifically designed for semantic image transmission. InverseJSCC utilizes pre-trained Generative Adversarial Networks (GANs), particularly StyleGAN, to refine noisy image reconstructions by solving an inverse optimization problem. GenerativeJSCC, on the other hand, integrates a StyleGAN-based decoder with an end-to-end optimized encoder-decoder network trained on both mean squared error (MSE) and learned perceptual image patch similarity (LPIPS) losses. Experimental results indicate that GenerativeJSCC achieves substantial improvements in both distortion and perceptual quality compared to conventional DeepJSCC methods, particularly in low-bandwidth and low-SNR scenarios.

In addition, JSCC techniques have significantly benefited from recent deep learning advancements, notably through the integration of nonlinear transform coding and deep JSCC frameworks. The recent work [82] introduced the Deep Video Semantic Transmission (DVST) framework, a sophisticated JSCC method specifically for video transmission over wireless channels. DVST integrates non-linear transform coding and a contextual deep JSCC encoder-decoder architecture, which adaptively extracts and transmits semantic features based on temporal correlations between frames. Unlike traditional image coding schemes, DVST utilizes an entropy model to rate-adaptively allocate bandwidth, transmitting semantic information more efficiently and robustly. Experiments demonstrate the superior performance

of DVST in perceptual quality metrics, significantly outperforming traditional separated source channel coding schemes such as H.264/H.265 with LDPC coding.

2.5.4 Paradigm Shift to Task-Oriented Communications

Deep learning-based JSCC has emerged as a robust solution in scenarios characterized by limited bandwidth and low SNR. Research in deep JSCC for reconstruction-oriented communication [14], [79], [83] has demonstrated its superiority over traditional source coding methods, such as JPEG [3] and JPEG2000 [4], as well as channel coding techniques, such as LDPC codes [5], particularly in environments with low SNR.

Existing reconstruction-oriented communication research primarily focused on data-centric metrics (e.g., PSNR [79], [83]–[86], SSIM [14], [79], [85], [86], and Multi-Scale Structural Similarity (MS-SSIM) [79], [85], [86]) to evaluate the effectiveness of deep JSCC. However, these metrics often lead to suboptimal task performance since high-fidelity reconstructions are not always necessary from the machine's perspective, whereas task-specific semantic information plays the most important role [87]–[92]. For example, in text transmission, the fidelity of words might be compromised to improve communication efficiency while still conveying the intended meanings [73], [93]. Similarly, in image transmission, image fidelity can be sacrificed for less communication overhead and higher task performance [65], [94], [95].

Nonetheless, existing works, such as [96], assumed that the amplitudes and phases of channel symbols are analog. Thus, it is not viable to implement them directly in digital communication systems [97]. To address this issue, the authors of [98] explored image transmission over the discrete channel (binary symmetric channel) using variational learning with a Bernoulli prior. This work was further extended by the authors of [99], who introduced adversarial regularization to enhance robustness. Furthermore, recent works [84], [100] investigated the transmission of natural images over an Additive White Gaussian Noise (AWGN) channel model with a finite channel input alphabet. Despite a good fit between the learned constellation diagram and the latent representation, the irregularity of the constellation diagram still poses significant challenges for deployment on commercial hardware. The author of [101] developed a digital task-oriented communication framework employing a hardware-limited scalar quantization approach, specifically tailored for computation-constrained situations, such as IoT. The results of this work provide valuable insights for future task-oriented JSCC designs.

2.6 Edge Intelligence

Edge intelligence refers to the integration of AI capabilities with edge computing infrastructure to enable intelligent data processing near the source of data generation. Unlike the traditional cloud-centric paradigm, which requires raw data to be transmitted to remote data centers for processing, edge intelligence decentralizes computational workload, thereby reducing

communication latency, preserving user privacy, and optimizing bandwidth utilization [102]. The emergence of this paradigm is driven by the proliferation of devices IoT, the exponential growth of data at the edge, and the increasing demand for real-time, context-sensitive AI-driven services in sectors such as autonomous driving, smart healthcare, and industrial automation [103].

The conventional intelligence model relies on centralized cloud servers, where AI models are trained and deployed, often resulting in significant latency and communication overhead. This approach becomes infeasible in scenarios requiring ultra-low-latency responses or where privacy-sensitive data cannot be transferred to external servers. Edge intelligence addresses these limitations by allowing data collection, storage, training, and inference to occur at the network edge, such as smartphones, autonomous vehicles, drones, and base stations [104]. Through this decentralization, edge intelligence supports scalable and resilient AI services that are less dependent on connectivity to the cloud.

At its core, edge intelligence encompasses four essential components: edge caching, edge training, edge inference, and edge offloading. Each of these elements plays a critical role in establishing a robust and efficient edge AI ecosystem. Edge caching involves the strategic storage of data or computation results to minimize redundancy and enhance inference speed. Edge training leverages localized or distributed datasets to build models that capture user- or environment-specific patterns with privacy constraints. Edge inference ensures rapid and efficient execution of AI models at the edge, often through compressed or optimized networks. Edge offloading facilitates resource-aware task delegation between edge devices, edge servers, and cloud infrastructure to balance computation load and energy consumption [105].

This paradigm shift is further catalyzed by advances in federated learning, model compression techniques such as pruning and quantization, and lightweight model design, which collectively make it feasible to deploy sophisticated AI functionality on resource-constrained devices. Moreover, as the demand for emerging applications grows with the advent of 6G networks and ubiquitous AI, edge intelligence is regarded as a cornerstone technology to achieve the vision of ubiquitous and intelligent connectivity [106].

The key architectural approach that underpins recent advances is *split inference*, where the inference network is partitioned between the device and the edge [11], [12], [107]–[113].

In this architecture, a mobile device initially processes data using a lightweight neural network to extract a compact feature vector. Subsequently, this vector is transmitted to an edge server for further processing, where deep JSCC is integral to the entire procedure [11], [12], [110]–[113]. Notably, an end-to-end framework that efficiently compresses intermediate features to optimize the bandwidth and computational resources at the edge was introduced in [111]. In addition, the authors of [11] developed a method to flexibly adjust the length of the transmission signal to adapt to dynamic communication environments while maintaining targeted inference accuracy.

Recent studies have shifted from reconstruction-oriented communication, which focuses on accurately reconstructing a signal at the receiver, to a task-oriented approach that prioritizes inference accuracy as the primary performance metric [11], [12], [111], [114], [115]. This

paradigm shift underscores a move towards optimizing communication systems to support specific functional requirements rather than general data fidelity.

Note that implementing such split-design architectures often necessitates modifications on both the device and the edge, which pose challenges in terms of compatibility with existing communication infrastructures. This issue highlights a significant barrier to widespread adoption, indicating the need for more compatible solutions that can seamlessly integrate with current technologies.

Chapter 3

Foundations: Task-Oriented Communication for Autonomous Systems

3.1 Introduction

As the era of intelligent transportation systems approaches, the concept of autonomous driving has moved from a futuristic vision to an imminent reality. The integration of advanced communication systems with automotive technology is leading the development of connected automation systems, such as autonomous driving. Edge computing brings computing and data storage closer to where they are needed, reducing latency, saving computational overhead, and saving bandwidth. The need for edge-enabled autonomous driving is driven by the limitations of traditional cloud computing models, which struggle to meet the real-time, high-bandwidth demands of autonomous vehicles.

Edge-enabled autonomous driving presents formidable challenges to traditional communication systems. Key challenges include:

1. *High Computational Overhead*: Autonomous vehicles generate a substantial amount of sensor data (e.g., radar, camera, and GPS), necessitating real-time transmission and processing for safe and efficient driving [116].

2. *Unreliable Wireless Connectivity*: Although 5G supports ultra-reliable low latency, service continuity remains challenging due to vehicle mobility and obstacle blockage [117]. In addition, the degradation of SNR in the wireless channel can cause a sudden breakdown in communication performance, which is known as the *cliff effect*.

3. *Real-time Responsiveness and Latency*: Minimizing processing time for sensor data analysis and decision-making is crucial for timely response in edge-enabled autonomous driving [116].

To address challenges in connected autonomous systems, especially in low SNR environments, a paradigm shift that integrates communication with computing becomes essential. This shift drives rethinking the gap between bit-level transmission built upon Shannon's theory and the requirements of tasks in automation systems. Autonomous driving, as a mission-critical task,

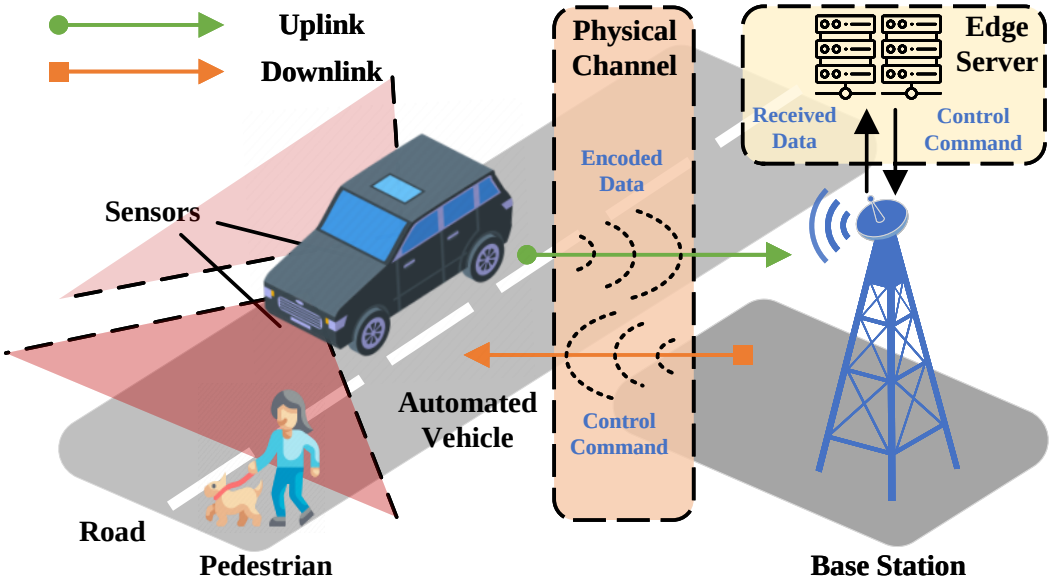


Figure 3.1: Edge computing enabled autonomous driving.

demands an innovative approach that extends communication efficiency beyond traditional information theory constraints.

From the machine's perspective, high-fidelity reconstructions are often not needed, whereas mission-critical semantic information plays the most important role. Semantic communication, which has emerged as a fundamental aspect of 6G technologies, seeks to revolutionize this landscape by prioritizing data semantics over bit-level data transmission. In particular, JSCC integrates the entire transmission process for direct semantic transmission, contrasting traditional segmented considerations [87]. Deep learning enabled JSCC demonstrates resilience to the *cliff effect*, showing superiority over separate source-channel coding, especially at lower SNRs, including the transmission of natural language [73], images [14], and videos [82].

In this chapter, we propose a novel deep TSCC framework based on modified CVAE to enhance edge-enabled autonomous driving. Inspired by JSCC-enabled image transmission [14] and the integration of JSCC with VAE [15], our approach jointly designs data transmission with pragmatic tasks, building a resilient E2E communication system against AWGN. Our methodology prioritizes task-oriented design, employing an E2E autonomous driving agent as the metric for training deep JSCC. We innovatively integrate β-CVAE with the autonomous driving agent, offering a holistic view of task-oriented source-channel coding.

The major contributions of this chapter are summarized as follows:

- We propose designing β-CVAE to synergize the autonomous driving agent with the source-channel coding, demonstrating a novel approach that integrates communications and computing in autonomous systems.

- By implementing TSCC within an edge-enabled state-of-the-art E2E autonomous driving agent, we demonstrate notable improvements in driving performance over traditional

Table 3.1: Summary of Main Symbols

Symbol	Explanation
$\boldsymbol{x}$	Input image
$\boldsymbol{z}$	Latent vector
$\check{\boldsymbol{z}}$	Channel input
$\tilde{\boldsymbol{z}}$	Normalized Channel input
$\hat{\boldsymbol{z}}$	Channel output
$\boldsymbol{m}$	State information
$\hat{\boldsymbol{m}}$	Received state information
$\boldsymbol{y}$	Reconstructed image
$\boldsymbol{a}$	Ground-truth action
$\hat{\boldsymbol{a}}$	Estimated action
$\boldsymbol{\mu}$	Mean of the latent vector
$\boldsymbol{\sigma}$	Standard deviation of the latent vector
$\beta_{\text{c-rec}}$	Hyperparameter
α, δ, ψ	Parameters of neural networks
$\boldsymbol{n}$	Gaussian noise
k	Dimension of the channel input
l	Dimension of the input image
P	Power constraint of transmitter
i, j	General index depended on context

communication methods and state-of-the-art deep JSCC approaches, particularly in terms of driving score.

- To the best of our knowledge, we pioneer the exploration of deep JSCC design for autonomous driving, illuminating the potential of comprehensive task-oriented deep JSCC.

Table 3.1 lists the main symbols used throughout this chapter.

3.2 System Model and Problem Formulation

The considered edge-enabled autonomous driving scenario is shown in Fig. 3.1. In our case study, we consider an automated vehicle equipped with a single RGB camera on the front, which is connected to the edge server. Sensor data are encoded and transmitted to the edge server via wireless communications, where the edge server, located at a base station, serves as a computational hub. The data are processed by an autonomous driving agent, deployed at the edge server, interpreting the information and generating appropriate control commands for the vehicle's current scenario. These commands are transmitted back to the vehicle for safe autonomous driving. Since the agent makes decisions based on the images obtained from

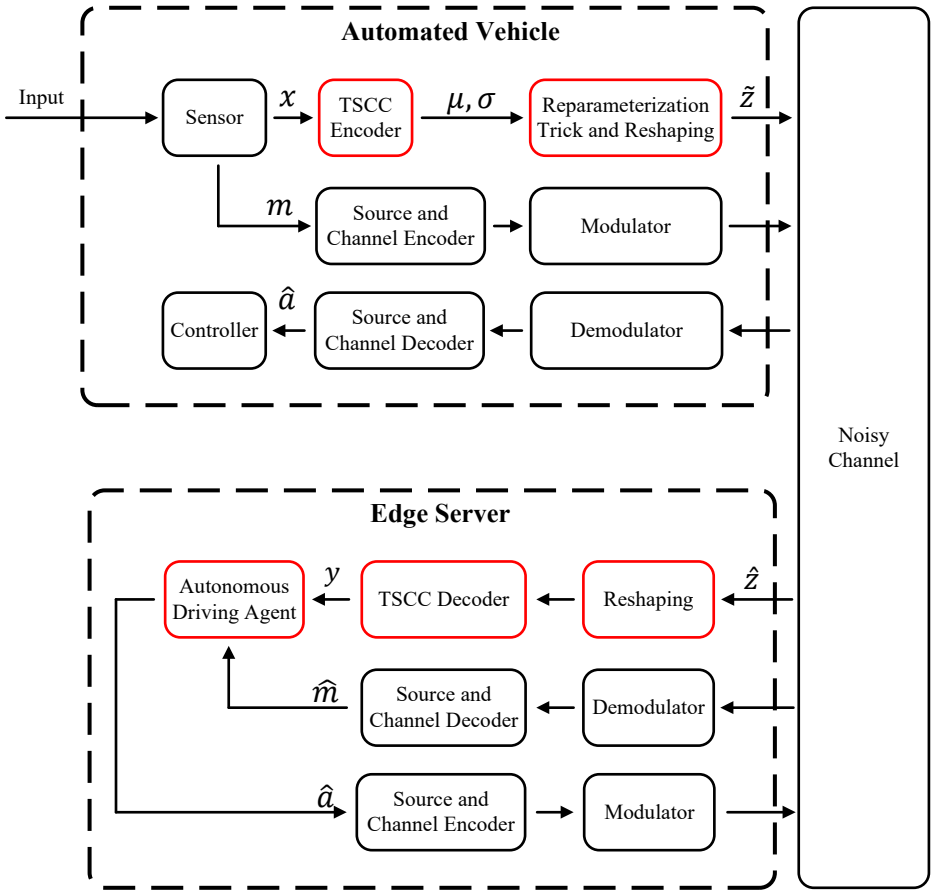


Figure 3.2: Framework of the TSCC enabled autonomous driving.

the RGB camera, image compression and transmission are critical to the performance of the edge-enabled autonomous driving system.

The following sections present a case study demonstrating our proposed TSCC's implementation and effectiveness in edge-enabled autonomous driving, which focuses on bandwidth utilization and noise resistance for image transmission. We demonstrate the loss function and the training process, using an E2E autonomous driving framework based on a monocular RGB camera as a baseline. Figure 3.2 illustrates our case study implementation, providing a practical perspective on the theoretical concepts. In particular, our approach integrates deep JSCC within the edge computing framework to address the challenges of bandwidth constraint and data transmission efficiency.

The input RGB image is denoted by $\boldsymbol{x} \in \mathbb{R}^l$, where $l = C \times H \times W$ is defined as the source bandwidth [14]. C stands for the number of color channels in the image. H and W represent the height and width of the image, respectively, measured in pixels.

On the vehicle side, instead of directly mapping RGB images to channel inputs [14], we first obtain the distribution of the latent vector from the encoder,

$$(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \mathrm{T}_{\alpha}(\boldsymbol{x}), \tag{3.1}$$

where $\mathrm{T}_\alpha(\cdot)$ is the joint task encoder with the parameters α. The encoder outputs the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of the latent vector. The latent vector $\boldsymbol{z}$ is sampled by the *reparameterization trick*

$$\boldsymbol{z} = \boldsymbol{\epsilon} \odot \boldsymbol{\sigma} + \boldsymbol{\mu}, \tag{3.2}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^d$ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, $\odot$ is elements-wise multiplication. The channel inputs $\check{\boldsymbol{z}} \in \mathbb{C}^k$ can be obtained from the latent vector $\boldsymbol{z}$ by combining every two neighboring elements into a single complex number. The dimension of the complex vector is half of the original latent vector, $k = d/2$. This operation is represented by a reshaping function,

$$\check{\boldsymbol{z}} = \mathrm{F}(\boldsymbol{z}). \tag{3.3}$$

To ensure that the channel inputs $\tilde{\boldsymbol{z}}$ satisfied the power constraint $\frac{1}{k}\mathbb{E}\|\tilde{\boldsymbol{z}}\|^2 \leq P$, where P is the average transmit power constraint, $\check{\boldsymbol{z}}$ is normalized as:

$$\tilde{\boldsymbol{z}} = \mathrm{N}(\check{\boldsymbol{z}}) \triangleq \sqrt{kP}\frac{\check{\boldsymbol{z}}}{\sqrt{\check{\boldsymbol{z}}^* \check{\boldsymbol{z}}}}, \tag{3.4}$$

where $\mathrm{N}(\cdot)$ denotes the normalization function, $\tilde{\boldsymbol{z}}^*$ is the conjugate transpose of $\tilde{\boldsymbol{z}}$. In the case of $k < l$, we define k/l as the *compression ratio* [14].

For the AWGN channel, the channel output can be given as follows,

$$\hat{\boldsymbol{z}} = \tilde{\boldsymbol{z}} + \boldsymbol{n}, \tag{3.5}$$

where $\hat{\boldsymbol{z}}$ is received channel inputs. The reconstructed image $\boldsymbol{y}$ can be obtained from the decoder, i.e.,

$$\boldsymbol{y} = \mathrm{T}_\delta^{-1}(\mathrm{F}^{-1}(\hat{\boldsymbol{z}})), \tag{3.6}$$

where $\mathrm{T}_\delta^{-1}(\cdot)$ is the task decoder with parameters δ, and F^{-1} is the inverse reshaping function.

Since the goal of our task-oriented design is to maximize autonomous driving performance rather than to minimize the difference between $\boldsymbol{x}$ and $\boldsymbol{y}$, the loss function should preserve the most task-relevant information for the autonomous driving agent and take driving performance into account as the optimization target.

In autonomous driving scenarios, accurate control commands are crucial and rely on environmental data (e.g., RGB images) and state information (including navigation information, vehicle speed, throttle, brake, and steering angle). As depicted in Fig. 3.2, this state information, denoted by $\boldsymbol{m}$, is transmitted using traditional communication methods for two main reasons: 1) semantic communication may not guarantee the fidelity required for critical information; 2) the required data rate for the transmission of state information is very low.

Upon receiving state information $\hat{\boldsymbol{m}}$ and the reconstructed image $\boldsymbol{y}$, the edge-deployed autonomous driving agent generates control commands as follows:

$$\hat{\boldsymbol{a}} = \mathrm{A}_\psi(\boldsymbol{y}, \hat{\boldsymbol{m}}), \tag{3.7}$$

where A_ψ represents the autonomous driving agent, and $\hat{\boldsymbol{a}}$ are the control commands derived from the received information. The individual components of the inferred action $\hat{\boldsymbol{a}} = (\hat{v}, \hat{s}, \hat{\boldsymbol{w}}, \hat{\boldsymbol{f}}^{\text{traj}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{f}}^{\text{ctrl}})$ are defined as follows:

- $\hat{v}$: estimated target speed.
- $\hat{s}$: value of the extracted features estimated by the expert [118].
- $\hat{\boldsymbol{w}}$: predicted waypoints from the trajectory branch.
- $\hat{\boldsymbol{f}}^{\text{traj}}$: estimated extracted features for trajectory planning.
- $\hat{\boldsymbol{b}} = [\hat{\boldsymbol{b}}_0, \hat{\boldsymbol{b}}_1, \ldots, \hat{\boldsymbol{b}}_\Gamma]$: estimated control actions from the beta distribution in the control prediction branch, where Γ denotes the prediction length.
- $\hat{\boldsymbol{f}}^{\text{ctrl}} = [\hat{\boldsymbol{f}}_0^{\text{ctrl}}, \hat{\boldsymbol{f}}_1^{\text{ctrl}}, \ldots, \hat{\boldsymbol{f}}_\Gamma^{\text{ctrl}}]$: predicted informative features of the control prediction branch.

Assuming that there exists a corresponding optimal (ground-truth) action $\boldsymbol{a}$ for each input RGB image $\boldsymbol{x}$, the problem of the proposed system model can be formulated as Problem 1.

Problem 1:

$$\min_{\alpha,\delta} \mathbb{E}_{\boldsymbol{a}\sim p(\boldsymbol{a}|\boldsymbol{z},\boldsymbol{m})} \left\{ \mathbb{E}_{\hat{\boldsymbol{a}}\sim p(\hat{\boldsymbol{a}}|\boldsymbol{z},\boldsymbol{m})} \left[d(\boldsymbol{a}, \hat{\boldsymbol{a}}) \right] \right\} \tag{3.8}$$

$$\text{s.t.} \quad (3.1), (3.2), (3.3), (3.4), (3.5), (3.6), (3.7), \tag{3.9}$$

where $d(\cdot, \cdot)$ denotes a task performance metric.

Problem 1 is an abstract formulation that describes the goal of optimizing task performance through appropriate parameter selection. The next section introduces how to approach Problem 1 via β-CVAE.

3.3 Variational Autoencoder

3.3.1 Motivation

The rise of data-driven systems in autonomous robotics, edge computing, and semantic communication has required the development of models capable of extracting compact, meaningful representations from high-dimensional sensory inputs. Traditional compression techniques such as JPEG and MPEG, although widely used, are built on handcrafted features and fail to capture task-relevant semantics, especially under noisy or constrained bandwidth environments [119].

To address these limitations, deep generative models have emerged as powerful tools capable of learning latent structures in data. VAEs stand out because of their solid probabilistic foundation and tractable training methods. VAEs offer an elegant blend of Bayesian inference

and deep learning, enabling them to encode complex distributions in a low-dimensional latent space suitable for both data reconstruction and semantic understanding.

In the context of edge-enabled robotics and communication, the ability of VAEs to generate and compress mission-critical features makes them ideal for low-latency systems. Rather than transmitting raw data, systems can transmit latent codes that encapsulate semantic or task-specific information, drastically reducing bandwidth while maintaining utility [14], [120].

VAEs were introduced by [121] as a scalable approach to variational inference using deep neural networks. They approximate the posterior distribution over latent variables using an encoder network and optimize the Evidence Lower Bound (ELBO) on the data log-likelihood. This makes VAEs especially well-suited for settings where both dimensionality reduction and probabilistic representation are necessary.

Key motivations for using VAEs in intelligent communication systems include:

- **Latent compression:** VAEs compress high-dimensional data (e.g., images or sensor measurements) into compact latent vectors that can be efficiently transmitted over noisy channels.
- **Uncertainty modeling:** Unlike deterministic autoencoders, VAEs explicitly model uncertainty, a critical feature for robust inference in real-world robotics and control systems.
- **Semantic disentanglement:** Variants like β-VAE encourage disentangled representations, where each dimension of the latent vector corresponds to a distinct generative factor, useful for interpretability and control [122].
- **Generative modeling:** VAEs can sample from the latent space to generate realistic variations, useful for data augmentation, missing data reconstruction, and simulation-to-real transfer.

3.3.2 Mathematical Foundation

Consider a dataset $\mathcal{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ of observed data samples. We assume the data are generated by a random process with an unobserved random variable $\boldsymbol{z}$. This is a two-step process: 1) $\boldsymbol{z}$ is generated from a prior $p_{\theta^*}(\boldsymbol{z})$; and 2) $\boldsymbol{x}$ is generated from a conditional prior $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$, where $p_{\theta^*}(\boldsymbol{z})$ and $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$ are from the parametric families $p_{\theta^*}(\boldsymbol{z})$ and $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$, respectively. However, the true parameters θ^* and the latent variable $\boldsymbol{z}$ are unknown, which makes it difficult to model the generation process.

Based on Bayes' theorem, we can calculate the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ as follows:

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{x})}, \tag{3.10}$$

where $p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{z})p_\theta(\boldsymbol{x}|\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$ is intractable due to the integration. Therefore, this posterior is also intractable. To deal with this problem, we define $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with parameters ϕ as a

variational approximation of $p_\theta(\boldsymbol{z}|\boldsymbol{x})$. Here, we refer to $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ as a probabilistic encoder and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ as a probabilistic decoder.

Since we aim to optimize the parameters ϕ such that:

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) \approx p_\theta(\boldsymbol{z}|\boldsymbol{x}), \tag{3.11}$$

we minimize the Kullback-Leibler (KL) divergence between them:

$$\begin{aligned} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z}|\boldsymbol{x})) &= \int q_\phi(\boldsymbol{x},\boldsymbol{z}) \log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z} \\ &= \int q_\phi(\boldsymbol{x},\boldsymbol{z}) \log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})}\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[-\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] + \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z})}\right] \\ &\quad + \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x})\right], \end{aligned} \tag{3.12}$$

where $D_{\mathrm{KL}}(\cdot)$ denotes the KL divergence. The third term in Eq. (3.12) can be rewritten as

$$\begin{aligned} \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x})\right] &= \int q_\phi(\boldsymbol{x},\boldsymbol{z}) \log p_\theta(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z} \\ &= \int q_\phi(\boldsymbol{x}) q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{z}\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}) \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,\mathrm{d}\boldsymbol{z}\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x})\right] \\ &= \text{Constant}. \end{aligned} \tag{3.13}$$

In that case, the third term in Eq. (3.12) is irrelevant to optimization. The target loss function can be formalized as

$$\begin{aligned} \mathcal{L}(\boldsymbol{x};\phi,\theta) &= D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z}|\boldsymbol{x})) - \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x})\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[-\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] + \mathbb{E}_{q_\phi(\boldsymbol{x},\boldsymbol{z})}\left[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z})}\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\int q_\phi(\boldsymbol{z}|\boldsymbol{x})(-\log p_\theta(\boldsymbol{x}|\boldsymbol{z}))\,\mathrm{d}\boldsymbol{z} + \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z})}\,\mathrm{d}\boldsymbol{z}\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\underbrace{\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[-\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] + D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z}))}_{-\mathrm{ELBO}}\right]. \end{aligned} \tag{3.14}$$

Note that minimizing the loss function of VAE is equivalent to maximizing the ELBO, which is defined as

$$\text{ELBO} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{\text{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z})). \tag{3.15}$$

3.3.3 β-CVAE

In this chapter, the system model comprises two neural networks: an encoder $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ with parameters ϕ, and a decoder $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ with parameters θ. The encoder transforms the data sample $\boldsymbol{x}$ into a latent vector $\boldsymbol{z}$, while the decoder reconstructs the original data from the latent vector. For simplicity, we denote $p_\theta(\boldsymbol{z})$ by $p(\boldsymbol{z})$.

The training of VAE is to minimize the following loss function:

$$\begin{aligned}
\mathcal{L}_{\text{VAE}}(\boldsymbol{x};\phi,\theta) =& \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[-\ln p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] + \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\ln \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{p(\boldsymbol{z})}\right] && (3.16)\\
=& \frac{1}{t}\sum_{i=1}^{t}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_i\|^2 + \frac{1}{2}\sum_{j=1}^{d}(\mu_j^2 + \sigma_j^2 - \ln\sigma_j^2 - 1) && (3.17)\\
=& \mathcal{L}_{\text{rec}}(\boldsymbol{x};\phi,\theta) + \mathcal{L}_{\text{KL}}(\boldsymbol{x};\phi). && (3.18)
\end{aligned}$$

Equation (3.17) assumes that $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ follows a Gaussian distribution with constant standard deviation [123]. The reconstructed data $\hat{\boldsymbol{x}}_i$ is derived from the decoder $p_\theta(\boldsymbol{x} \mid \boldsymbol{z}_i)$, and t represents the number of latent vectors sampled from the latent space. The latent vector, $\boldsymbol{z} \in \mathbb{R}^d$, follows the Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and standard deviation $\boldsymbol{\sigma} \in \mathbb{R}^d$, where d denotes the dimension of the latent space. $\mathcal{L}_{\text{rec}}(\boldsymbol{x};\phi,\theta)$ is the first term in Eq. (3.17), which represents the reconstruction errors. Meanwhile, $\mathcal{L}_{\text{KL}}(\boldsymbol{x};\phi)$ is the second term in Eq. (3.17), which represents the Kullback-Leibler (KL) divergence. In addition, the prior distribution $p(\boldsymbol{z})$ is assumed to be a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

β-VAE [122] extended the loss function of vanilla VAE by adding an coefficient β_{rec}:

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{x};\phi,\theta) = \beta_{\text{rec}}\mathcal{L}_{\text{rec}}(\boldsymbol{x};\phi,\theta) + \mathcal{L}_{\text{KL}}(\boldsymbol{x};\phi), \tag{3.19}$$

where β_{rec} for $\mathcal{L}_{\text{rec}}(\boldsymbol{x};\phi,\theta)$ to control the training balance.

CVAE introduces a class label m as a condition, with which the new loss function can be extended in the following expression:

$$\begin{aligned}
\mathcal{L}_{\text{CVAE}}(\boldsymbol{x},\boldsymbol{m};\phi,\theta) =& \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{m})}\left[-\ln p_\theta(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{m})\right]\\
&+ \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{m})}\left[\ln \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{m})}{p(\boldsymbol{z}|\boldsymbol{m})}\right] && (3.20)\\
=& \frac{1}{t}\sum_{i=1}^{t}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_i\|^2 + \frac{1}{2}\sum_{i=1}^{d}(\mu_i^2 + \sigma_i^2 - \ln\sigma_i^2 - 1) && (3.21)\\
=& \mathcal{L}_{\text{c-rec}}(\boldsymbol{x},\boldsymbol{m};\phi,\theta) + \mathcal{L}_{\text{c-KL}}(\boldsymbol{x},\boldsymbol{m};\phi). && (3.22)
\end{aligned}$$

Algorithm 1 TSCC Training Algorithm

Initialization: Initialize the neural network parameters α and δ.
Input: Image dataset $\mathcal{X}$ with corresponding ground-truth agent output $\mathcal{A}$ and state information $\mathcal{M}$.
while not converged **do**
  Sample $\boldsymbol{a}$ from $\mathcal{A}$ with the corresponding $\boldsymbol{m}$ from $\mathcal{M}$.
  Sample $\boldsymbol{x}$ from $\mathcal{X}$ according to $\boldsymbol{a}$ and $\boldsymbol{m}$.
  Encode image $\boldsymbol{x}$ to mean and standard deviation: $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \mathrm{T}_\alpha(\boldsymbol{x})$.
  Sample latent vector using the *reparameterization trick*: $\check{\boldsymbol{z}} \leftarrow \boldsymbol{\epsilon} \odot \boldsymbol{\sigma} + \boldsymbol{\mu}$.
  Reshape and normalize latent vector: $\tilde{\boldsymbol{z}} \leftarrow \mathrm{N}(\mathrm{F}(\check{\boldsymbol{z}}))$.
  Reconstruct image from $\tilde{\boldsymbol{z}}$: $\boldsymbol{y} \leftarrow \mathrm{T}_\delta^{-1}(\mathrm{F}^{-1}(\tilde{\boldsymbol{z}}))$.
  Generate control commands: $\hat{\boldsymbol{a}} \leftarrow \mathrm{A}_\psi(\boldsymbol{y}, \boldsymbol{m})$.
  Compute $\mathcal{L}_{\text{c-rec}}(\boldsymbol{a}, \boldsymbol{m}; \alpha, \delta)$ and $\mathcal{L}_{\text{c-KL}}(\boldsymbol{a}, \boldsymbol{m}; \alpha)$.
  Update neural network parameters: $\alpha \xleftarrow{+} -\nabla_\alpha(\beta_{\text{c-rec}}\mathcal{L}_{\text{c-rec}}(\boldsymbol{a}, \boldsymbol{m}; \alpha, \delta) + \mathcal{L}_{\text{c-KL}}(\boldsymbol{a}, \boldsymbol{m}; \alpha))$.
  $\delta \xleftarrow{+} -\nabla_\delta \beta_{\text{c-rec}}\mathcal{L}_{\text{c-rec}}(\boldsymbol{a}, \boldsymbol{m}; \alpha, \delta)$.
end while
Output: The coding neural network $\mathrm{T}_\alpha(\cdot)$ and $\mathrm{T}_\delta^{-1}(\cdot)$.

The autoencoder has been used to train deep JSCC in an E2E manner [15], [73], [75]. Inspired by [15], we designed TSCC based on β-CVAE.

3.3.4 Training of Task-Oriented β-CVAE

Baseline control commands $\boldsymbol{a}$ are produced by a coach AI [118] using lossless images and state information. We design the β-CVAE for the mapping from $\boldsymbol{a}$ to $\hat{\boldsymbol{a}}$. Combining Eq. (3.19), Eq. (3.21), and Eq. (3.22), we design the task-oriented loss function of TSCC as

$$\begin{aligned}\mathcal{L}_{\text{TSCC}}(\boldsymbol{a}, \boldsymbol{m}; \alpha, \delta) =& \beta_{\text{c-rec}}\mathbb{E}_{\boldsymbol{z}\sim q_\alpha(\boldsymbol{z}|\boldsymbol{a},\boldsymbol{m})}[-\ln p_\delta(\boldsymbol{a}|\boldsymbol{z}, \boldsymbol{m})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim q_\alpha(\boldsymbol{z}|\boldsymbol{a},\boldsymbol{m})}\left[\ln \frac{q_\alpha(\boldsymbol{z}|\boldsymbol{a}, \boldsymbol{m})}{p(\boldsymbol{z}|\boldsymbol{m})}\right] \qquad (3.23)\\ =& \beta_{\text{c-rec}}\frac{1}{t}\sum_{i=1}^{t}\|\boldsymbol{a} - \hat{\boldsymbol{a}}_i\|^2 + \frac{1}{2}\sum_{i=1}^{d}(\mu_i^2 + \sigma_i^2 - \ln\sigma_i^2 - 1) \qquad (3.24)\\ =& \beta_{\text{c-rec}}\mathcal{L}_{\text{c-rec}}(\boldsymbol{a}, \boldsymbol{m}; \alpha, \delta) + \mathcal{L}_{\text{c-KL}}(\boldsymbol{a}, \boldsymbol{m}; \alpha). \qquad (3.25)\end{aligned}$$

The first term $\mathcal{L}_{\text{c-rec}}(\cdot)$ ensures the fidelity of actions $\hat{\boldsymbol{a}}$ under the given state information $\boldsymbol{m}$. Meanwhile, the second term $\mathcal{L}_{\text{c-KL}}(\cdot)$ is adversarial to the first term $\mathcal{L}_{\text{c-rec}}(\cdot)$, and introduces noise into the training process. Since $\mathcal{L}_{\text{c-KL}}$ introduces noise, we can train TSCC without channel noise and execute TSCC over practical channels with various SNRs. Furthermore, the hyperparameter $\beta_{\text{c-rec}}$ balances two terms to achieve a good trade-off between action fidelity and noise-resistant ability. The training process of the proposed TSCC is shown in Algorithm 1.

3.4 Simulation Result

3.4.1 Dataset

CARLA is an open-source simulator [124], which offers diverse urban environments to facilitate research on autonomous driving. It offers dynamic agents and traffic scenarios as in real-world driving conditions. The image dataset from [125] is used to train TSCC. This dataset comprises images ($C = 3$, $H = 256$, and $W = 900$) across various urban maps, with training and testing sets containing 189,524 (four maps: Town01, Town03, Town04, and Town06) and 27,201 images (four maps: Town02, Town05, Town07, and Town10), respectively. In addition to offline image testing, we use Town05 to test our proposed framework in real time.

3.4.2 Evaluation Metrics

To evaluate the effectiveness of our approach, we compare the driving performance of TSCC with different baselines in the CARLA simulator. The driving score[1] has been widely used in the existing literature to evaluate the vehicle's ability to follow predetermined waypoints, reach the destination, and avoid violating traffic rules. The driving score of the i_{th} road is defined as

$$\text{Score}_i = R_i P_i, \tag{3.26}$$

where $R_i \in [0, 100]$ denotes the percentage of the route distance completed by an agent and P_i denotes the infraction penalty. The infraction penalty is defined as

$$P_i = \prod_j p_j^{\gamma_j}, \tag{3.27}$$

where $p_j^{\gamma_j}$ denotes the j_{th} infraction with value p^{γ_j} and type γ_j. There are six kinds of infractions:

- Collisions with pedestrians: $p^{\gamma_j} = 0.5$.
- Collisions with other vehicles: $p^{\gamma_j} = 0.6$.
- Collisions with static elements: $p^{\gamma_j} = 0.65$.
- Running a red light: $p^{\gamma_j} = 0.70$.
- Running a stop sign: $p^{\gamma_j} = 0.80$.
- Off-road driving: The percentage of off-road driving will be reduced from the infraction penalty, canceling out the part of the route completion.

[1] `https://leaderboard.carla.org/`

The tests are carried out in **Town05** with four different scenarios (clear noon, cloudy sunset, soft rain down, and hard rain night), where the test is repeated three times in each scenario.

3.4.3 Evaluation on CARLA

The impacts of the compression ratio on the driving score are shown in Fig. 3.3, where SNR = 10 dB. We compare our approach with traditional image coding methods: JPEG, JPEG2000, and BPG. The source (image) coding is followed by (2048, 6144) LDPC codes with 64-QAM digital modulation schemes. Specifically, their compression ratios range from 0.009 to 0.251. For the proposed TSCC neural network, we set $d = 4096$ and $k = 2048$, resulting in a significantly lower compression ratio k/l of 0.003. In addition, we set $\beta_{\text{c-rec}} = 2048$ and $P = 1$. In particular, we assume $\boldsymbol{m} = \hat{\boldsymbol{m}}$ in this case study. The results show that when the required driving score is 20, the TSCC framework could save 98.36% communication bandwidth compared to the existing methods.

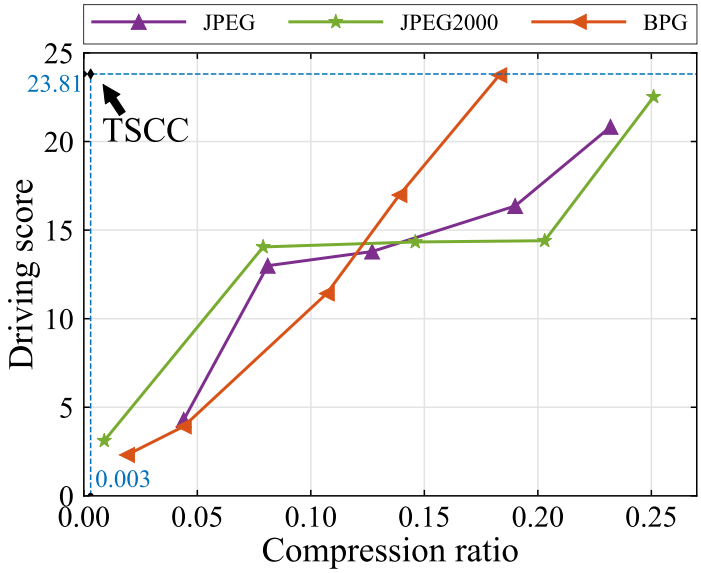


Figure 3.3: Driving score of traditional image coding method with varied compression ratio.

As shown in Fig. 3.4, the driving score highly depends on SNR. The compression ratios of JPEG, JPEG2000, and BPG are set to 0.232, 0.251, and 0.183, respectively. The compression ratio of our method is set to 0.003. In addition, we compare the proposed TSCC with the state-of-the-art JSCC method using the same neural network structure, with legends "JSCC-AE" [14] and "JSCC-VAE" [15]. We trained JSCC-AE with different SNRs and found that the testing performance is similar. In this figure, JSCC-AE is trained with 0 dB SNR. Like the training of TSCC, JSCC-VAE is trained without channel noise.

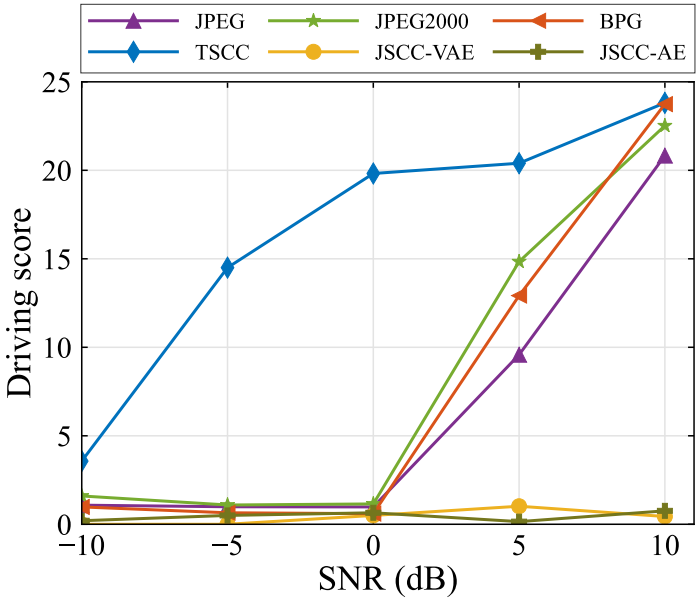


Figure 3.4: Driving score with varied SNR.

The results show that our TSCC achieves much higher driving scores at 0 dB SNR (19.82) and -5 dB SNR (14.5), respectively, compared to other methods. In contrast, the driving scores of traditional image coding methods drop dramatically in the low SNR region (below 0 dB).

Although traditional image coding, JSCC-AE, and JSCC-VAE yield high-quality image reconstructions for human vision, as shown in Fig. 3.5, they retain redundant information not critical to machine vision. In particular, some mission-critical information, such as pedestrians and road markers, is not clearly presented by JSCC-AE and JSCC-VAE, leading to poor performance for edge-enabled autonomous driving. In Table 3.2, we evaluate some other performance metrics, including PSNR, MS-SSIM, and FID, which are specially designed for human perception. The different preferences between human vision and machine vision indicate the importance of task-oriented source-channel coding design for machine vision.

Table 3.2: Human Perceptional Metrics

Method	PSNR(dB) ↑	MS-SSIM ↑	FID↓
JPEG	34.56	0.99	5.83
JPEG2000	37.54	0.99	7.17
BPG	34.93	0.98	6.68
JSCC-AE	19.74	0.56	173.46
JSCC-VAE	20.85	0.66	160.66
TSCC	7.72	0.13	347.30

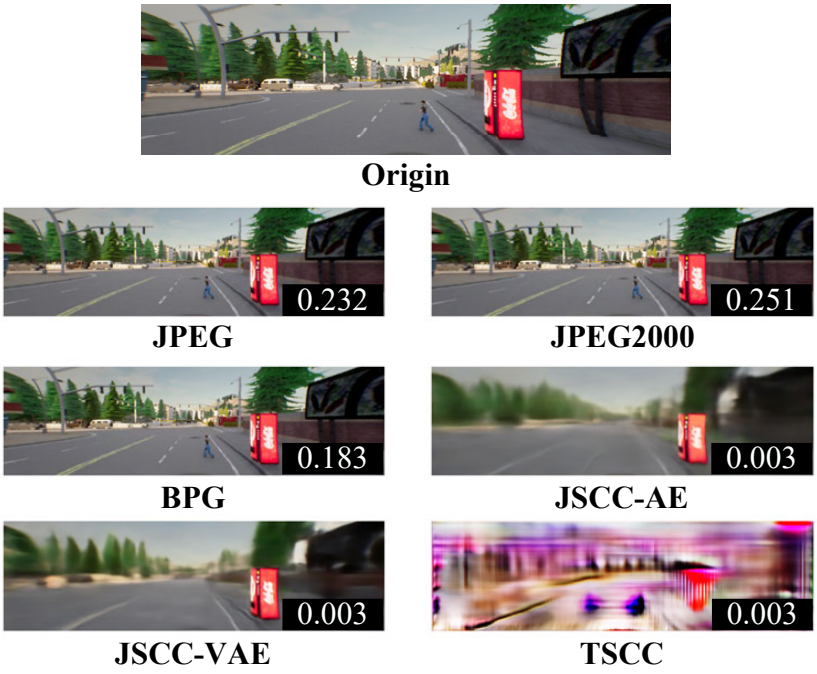


Figure 3.5: Qualitative example of TSCC and baseline methods at 10 dB SNR. The compression ratios of example images are shown in the lower right corner.

3.5 Conclusion

In this chapter, we proposed a TSCC framework for edge-enabled autonomous driving that takes advantage of the knowledge of a well-trained autonomous driving agent. We designed β-CVAE in a task-oriented way to guide the training of deep source-channel coding to preserve mission-critical information for machine vision. We carried out a case study in CARLA and the results showed that proposed TSCC save 98.36% communication bandwidth compared with the existing method when SNR is 10 dB. In the low SNR region (below 0 dB), all traditional methods and existing JSCC approaches do not work, while our TSCC approach can still achieve 83.24% or 60.9% driving scores when SNR is 0 dB or -5 dB.

Chapter 4

Integration: Aligning Task- and Reconstruction-Oriented Communication (ATROC) for Edge Intelligence

4.1 Introduction

Reconstruction-oriented communications are designed to recover the transmitted information at the receiver sides, often involving traditional source and channel coding techniques. This approach is commonly used in systems where the fidelity of the information is paramount, such as in audio or video streaming services. The structure of separate source and channel coding, a cornerstone in the design of communication systems, has been shown to be theoretically optimal via AEP with infinitely long source and channel blocks [56]. However, in practical scenarios, this separation often leads to inefficiencies and suboptimal performance, particularly for Artificial Intelligence (AI) driven applications [14].

The pervasive advancement of AI technologies, particularly in the context of deep learning, presents novel challenges for future communication systems, where the throughput required by AI agents could be much higher than that of human users.

Recent developments in deep learning have shown that JSCC can potentially address some of these inefficiencies and outperform traditional separate coding designs. This approach is especially potent in environments where traditional methods struggle to keep pace with the data demands of AI-driven applications. However, JSCC-based reconstruction-oriented communications, which focus on accurately reconstructing a signal on receiver sides, often waste communication resources by transmitting task-agnostic information [79].

To address these issues, task-oriented communication has emerged as a key technology and has attracted significant research interests [11], [12], [126], [127]. Using the capabilities of deep learning, task-oriented JSCC focuses on transmitting task-specific information, thus improving efficiency and reducing the data rate in critical applications. This requires joint optimization of the JSCC and inference network, which must be co-designed for effective task-oriented communication [11]. Note that existing JSCC designs are mainly based on analog

communication principles [14] and cannot be integrated with existing digital communication infrastructures.

Furthermore, cloud-based services introduce unacceptable latency for real-time applications, such as autonomous driving [116], [128]. To mitigate this issue, *edge inference* [11], [113], [129] has become a promising approach, enabling quick response to real-time AI applications. However, widely deployed AI agents bring significant communication loads to communication systems. Emergent methods based on JSCC have shown great potential to solve this problem [111]–[113].

Recognizing these multifaceted challenges, there is a growing interest in developing communication systems that are not only task-oriented but also aligned with reconstruction-oriented communication frameworks. This has led to the proposition of what we refer to as ATROC, which aims to bridge the gap between the efficiency of task-specific data transmission and the robustness of reconstruction-oriented communications, enabling the seamless integration of AI technologies with existing network infrastructures.

4.1.1 Contributions

This chapter introduces a novel communication framework compatible with reconstruction-oriented communication, especially for edge inference, termed Aligned Task- and Reconstruction-Oriented Communication (ATROC). By extending IB theory [6] and incorporating JSCC modulation, this framework is designed to enhance AI-driven applications. It prioritizes task relevance in data transmission strategies, shifting focus from traditional signal reconstruction fidelity to operational efficiency and effectiveness in real-world applications. The key contributions of this chapter are summarized as follows:

- **Development of an ATROC Framework:** Based on IB theory, we develop a framework that aligns task-oriented communications with reconstruction-oriented communications. The framework focuses on maximizing mutual information between inference results and encoded features, minimizing mutual information between the encoded features and the input data, and preserving task-specific information through the information reshaper. This reshaper is expert at transforming received symbols into task-specific data, maintaining the same data structure as the input while ensuring the preservation of task-specific information.

- **Innovation of an Information Reshaper:** We introduce an information reshaper within our extended IB theory, laying a foundational aspect of ATROC. This component is crucial for adapting the communication to the specific needs of the task without compromising the integrity of the transmitted data.

- **Variational Approximation for Tractable Information Estimation:** Due to the intractability of mutual information in the training and inference of deep neural networks, we employ a variational approximation approach, known as VIB. This approach allows

us to establish a tractable upper bound for these terms, enabling training and inference of deep neural networks.

- **Adaptation of a JSCC Modulation Scheme:** We design a JSCC modulation scheme that aligns JSCC symbols with a predefined constellation scheme. This scheme ensures compatibility of our framework with classic modulation techniques, making it more adaptable to existing communication infrastructures.
- **Performance Enhancement in Edge-Based Autonomous Driving:** In our simulation, we validate that the ATROC framework outperforms reconstruction-oriented methods for edge-based autonomous driving [125]. Specifically, our method reduced 99.19% communication load, in terms of bits per service, compared to existing methods, without compromising the driving score of the autonomous driving agent.

4.1.2 Organization and Notations

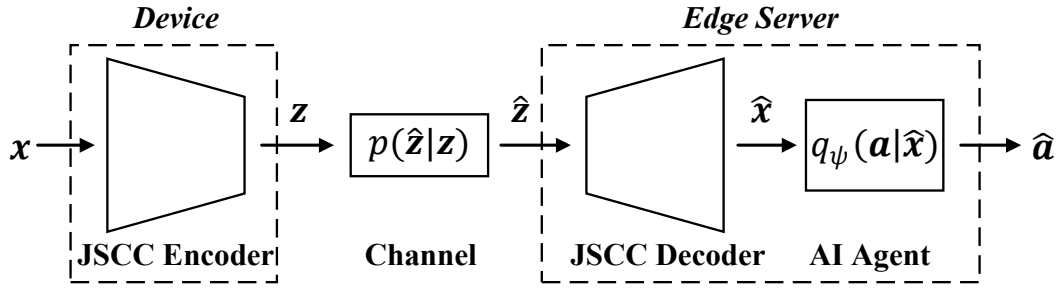
The rest of this chapter is organized as follows: Section 4.2 details the system model and discusses how the proposed framework advances reconstruction-oriented and non-aligned task-oriented communication approaches. Section 4.3 introduces the IB theory for ATROC and elaborates on the corresponding VIB derivation. In Section 4.4, we propose a JSCC modulation technique that is compatible with classical modulation methods, such as QAM. Section 4.5 extends the framework of VIB to enhance edge-based autonomous driving applications. The experimental results are presented in Section 4.6, which evaluates the performance of our proposed ATROC framework and the JSCC modulation. Finally, Section 4.7 concludes this chapter.

Table 4.1 lists the main symbols used throughout this chapter.

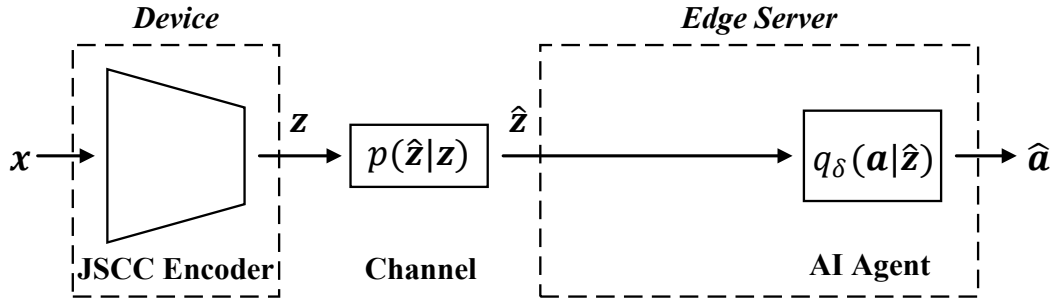
Table 4.1: Summary of Main Symbols

Symbol	Explanation
$\boldsymbol{x}$	Input data
$\hat{\boldsymbol{x}}$	Reconstructed input data
$\boldsymbol{z}$	JSCC symbols
$\bar{\boldsymbol{z}}$	Quantized JSCC symbols
$\boldsymbol{z}_{\text{in}}$	Channel input
$\boldsymbol{z}_{\text{out}}$	Channel output
$\check{\boldsymbol{z}}$	Equalized JSCC symbols
$\tilde{\boldsymbol{z}}$	Scaled JSCC symbols
$\hat{\boldsymbol{z}}$	Reconstructed JSCC symbols
$\boldsymbol{y}$	Task-specific data
$\boldsymbol{a}$	Target action
$\hat{\boldsymbol{a}}$	Inferred action
$\beta_1, \beta_2, \hat{\beta}_1, \hat{\beta}_2$	Lagrange multiplier
$\phi, \theta, \psi, \delta$	Parameters of neural networks
h	Channel coefficient
$\boldsymbol{n}$	Gaussian noise
k	Dimension of the JSCC symbols
l	Dimension of the input data
ζ	Upper bound of rate
Ω	Size of mini-batch
u	Number of constellation points
r	Constellation parameter
$e_{(\cdot)}$	Constellation point
P_{target}	Power constraint of transmitter
$P_{\bar{\boldsymbol{z}}}$	Power of quantized symbols
β_Q	Hyperparameter of quantization loss
$\Gamma, \lambda_{\text{feat}}, \lambda_{\text{traj}}, \lambda_{\text{ctrl}}, \lambda_{\text{aux}}$	Hyperparameters of edge AI agent
J_1, J_2	Sampling number
i, j	General index depended on context

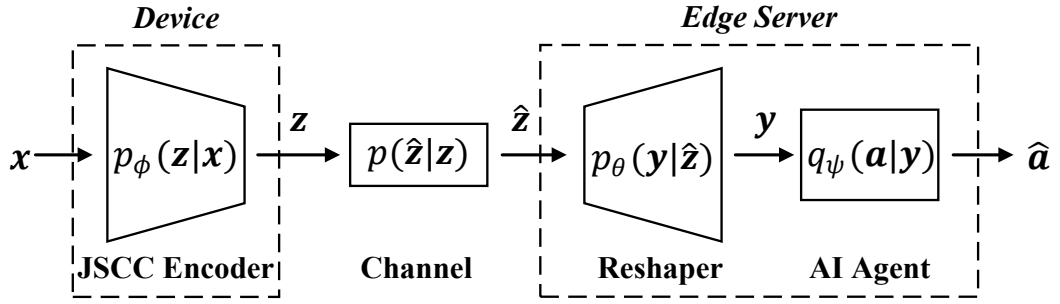
4.2 ATROC Framework for Edge Intelligence



(a) Reconstruction-oriented communication for edge intelligence.



(b) Non-aligned task-oriented communication for edge intelligence.



(c) Proposed ATROC for edge intelligence.

Figure 4.1: Comparison of three JSCC-enabled communication frameworks for edge inference: Reconstruction-oriented, non-aligned task-oriented, and ATROC frameworks. All three frameworks can share a similar JSCC encoder structure on the device side. On the edge side, reconstruction-oriented communication aims to fully reconstruct the input data, including both task-specific and task-agnostic information. In contrast, non-aligned task-oriented communication focuses solely on preserving task-specific information and uses JSCC symbols directly for inference. ATROC merges the benefits of the previous two by transferring task-specific information and ensuring that data structures are compatible with existing AI agent networks, enhancing integration and efficiency.

Edge intelligence refers to an AI agent (system) that operates at edge servers rather than relying on centralized servers or cloud-based services. These systems process data locally on devices or at the edge of the network as shown in Fig. 4.1.

The reconstruction-oriented communication framework (see Fig. 4.1a) aims to preserve all information from the input data $\boldsymbol{x}$ in the reconstructed data $\hat{\boldsymbol{x}}$. The idea is to minimize the distance $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$, where $d(\cdot, \cdot)$ is a predefined data-centric metric. This task-agnostic strategy may result in transmitting redundant data for AI agents, leading to poor resource utilization efficiency.

To improve efficiency, the principle of IB has been developed to transmit task-relevant information [11], as shown in Fig. 4.1b. However, a significant challenge arises with this approach: the dimensions of the received symbols often do not align with the input dimensions required by the edge AI agent. This mismatch necessitates a redesign of the AI agent to accommodate different input sizes, leading to poor compatibility.

To address this, we propose an ATROC framework, as depicted in Fig. 4.1c, enabling the use of a unified inference network across both task-oriented and reconstruction-oriented communication paradigms.

In this framework, the JSCC encoder deployed on the mobile device is denoted by $p_\phi(\boldsymbol{z}|\boldsymbol{x})$, where ϕ represents the parameters. The encoder maps the input data $\boldsymbol{x} \in \mathbb{R}^l$ to JSCC symbols $\boldsymbol{z} \in \mathbb{C}^k$, where $\boldsymbol{z} = [z_1, \cdots, z_k]$. Here, l and k are the dimensions of the input data and the JSCC symbols, respectively. After quantization and power normalization, the JSCC symbols $\boldsymbol{z}$ are transmitted through a physical channel. In this chapter, we model the communications between the mobile device and the edge server as Gaussian or Rayleigh fading channels:

$$\boldsymbol{z}_{\text{out}} = h \cdot \boldsymbol{z}_{\text{in}} + \boldsymbol{n}, \tag{4.1}$$

where $\boldsymbol{z}_{\text{in}}$ represents channel input and $\boldsymbol{z}_{\text{out}}$ represents channel output. $\boldsymbol{n} \sim \mathcal{CN}(0, \boldsymbol{\sigma}_n^2 I)$ is a Gaussian noise with zero mean and standard deviation $\boldsymbol{\sigma}_n$. For the Gaussian channel, we set $h = 1$, whereas for the Rayleigh fading channel, h is modeled as a complex Gaussian variable, $h \sim \mathcal{CN}(0, 1)$, to represent the multipath fading effect.

After the process of equalization, scaling, and detection, the reconstructed symbols $\hat{\boldsymbol{z}}$ are transformed by the information reshaper $p_\theta(\boldsymbol{y}|\hat{\boldsymbol{z}})$ with parameters θ to provide task-specific data $\boldsymbol{y}$. These data are then utilized by the AI agent $q_\psi(\boldsymbol{a}|\boldsymbol{y})$ with parameters ψ, to generate the inferred action $\hat{\boldsymbol{a}}$, which approximates the ground truth action $\boldsymbol{a}$.

4.3 Information Bottleneck for ATROC

4.3.1 Problem Description

Following the standard IB framework [6], [10], we assume the joint distribution of the system variables as follows:

$$p(\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{z}, \hat{\boldsymbol{z}}, \boldsymbol{y}) = p(\boldsymbol{a})p(\boldsymbol{x}|\boldsymbol{a})p_\phi(\boldsymbol{z}|\boldsymbol{x})p(\hat{\boldsymbol{z}}|\boldsymbol{z})p_\theta(\boldsymbol{y}|\hat{\boldsymbol{z}}). \tag{4.2}$$

This sets up the Markov chain depicted as:

$$A \leftrightarrow X \leftrightarrow Z \leftrightarrow \hat{Z} \leftrightarrow Y. \tag{4.3}$$

The transformation from reconstructed symbols $\hat{\boldsymbol{z}}$ to task-specific data $\boldsymbol{y}$ is designed to preserve task-specific information, aligning task-oriented paradigms with traditional and reconstruction-oriented approaches. Based on the IB theory [6], [10], we formulate the following optimization problem:

$$\begin{aligned} \min \quad & -I(A;Y) && \text{(4.4a)} \\ \text{s.t.} \quad & I(X;\hat{Z}) - \zeta \leq 0, && \text{(4.4b)} \\ & I(A;Y) - I(A;\hat{Z}) = 0, && \text{(4.4c)} \end{aligned}$$

where ζ represents the upper bound of data rate depending on the channel. The data processing inequality [56] implies that, ideally, if Y and $\hat{Z}$ contain equivalent information about the action A, the equality $I(A;Y) - I(A;\hat{Z}) = 0$ holds.

We further formulate this problem as

$$\begin{aligned} \mathcal{L}_{\text{IB}}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) &= \underbrace{-I(A;Y)}_{\text{Distortion}} + \beta_1 (\underbrace{I(X;\hat{Z})}_{\text{Rate}} - \zeta) + \beta_2 \underbrace{(I(A;Y) - I(A;\hat{Z}))}_{\text{Alignment}} && \text{(4.5a)} \\ &\equiv -I(A;Y) + \hat{\beta}_1 I(X;\hat{Z}) - \hat{\beta}_2 I(A;\hat{Z}) && \text{(4.5b)} \\ &\equiv \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[-\log p(\boldsymbol{a}|\boldsymbol{y})] + \hat{\beta}_1 D_{KL}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}) \| p(\hat{\boldsymbol{z}})) \\ &\qquad + \hat{\beta}_2 \mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi}[-\log p(\boldsymbol{a}|\hat{\boldsymbol{z}})]], && \text{(4.5c)} \end{aligned}$$

where $\beta_1 > 0$ and $\beta_2 > 0$ are the Lagrange multipliers. The detailed derivation can be found in Section 4.3.2. The first term $-I(A;Y)$ and the second term $I(X;\hat{Z})$ formalize the classic information bottleneck, meanwhile, the third term $[I(A;Y) - I(A;\hat{Z})]$ aligns the task-relevant information between the task-specific data $\boldsymbol{y}$ and the reconstructed symbols $\hat{\boldsymbol{z}}$.

In the case $\beta_2 \neq 1$, we define $\hat{\beta}_1 = \frac{\beta_1}{1-\beta_2}$ and $\hat{\beta}_2 = \frac{\beta_2}{1-\beta_2}$. Then Eq. (4.5a) can be expressed as Eq. (4.5b). In the case $\beta_2 = 1$, Eq. (4.5a) is simplified to the classic IB formulation [6], [10], [11]:

$$\mathcal{L}_{\text{IB}}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) = \underbrace{-I(A;\hat{Z})}_{\text{Distortion}} + \beta_1 \underbrace{I(X;\hat{Z})}_{\text{Rate}}. \tag{4.6}$$

This extended IB theory preserves more task-specific information, and the bits per service is the same as the previous IB approaches. Meanwhile, it maintains the dimension and structure required for edge inference.

4.3.2 Variational Approach

With the objective function Eq. (4.5b), we first illustrate how to compute each term for training ϕ and θ. We start with the first term, $-I(A;Y)$, expressed as:

$$\begin{aligned}
-I(A;Y) &= -\int p(\boldsymbol{a},\boldsymbol{y}) \log \frac{p(\boldsymbol{a}|\boldsymbol{y})}{p(\boldsymbol{a})} \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} \\
&= -\int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} - \left[-\int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y}\right] \\
&= -\int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} - \left[-\int p(\boldsymbol{a}) \log p(\boldsymbol{a}) \,\mathrm{d}\boldsymbol{a}\right] \\
&= -\int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} - H(A),
\end{aligned} \tag{4.7}$$

where $H(A) = -\int p(\boldsymbol{a}) \log p(\boldsymbol{a}) \,\mathrm{d}\boldsymbol{a}$ denotes the entropy. $p(\boldsymbol{a}|\boldsymbol{y})$ is the posterior probability, which can be derived through the Markov Chain [10], [11] as:

$$\begin{aligned}
p(\boldsymbol{a}|\boldsymbol{y}) &= \int p(\boldsymbol{a},\boldsymbol{x},\boldsymbol{z},\hat{\boldsymbol{z}}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\,\mathrm{d}\hat{\boldsymbol{z}} \\
&= \int \frac{p(\boldsymbol{a})p(\boldsymbol{x}|\boldsymbol{a})p_\phi(\boldsymbol{z}|\boldsymbol{x})p(\hat{\boldsymbol{z}}|\boldsymbol{z})p_\theta(\boldsymbol{y}|\hat{\boldsymbol{z}})}{p(\boldsymbol{y})} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\,\mathrm{d}\hat{\boldsymbol{z}}.
\end{aligned} \tag{4.8}$$

Given the complexity of this integration, we employ a neural network $q_\psi(\boldsymbol{a}|\boldsymbol{y})$ as a variational approximation to $p(\boldsymbol{a}|\boldsymbol{y})$.

Denoting the KL divergence as D_{KL}. According to the definition of KL divergence [56], we can derive the following expression:

$$D_{\mathrm{KL}}(p(\boldsymbol{a}|\boldsymbol{y}) \parallel q_\psi(\boldsymbol{a}|\boldsymbol{y})) = \int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} - \int p(\boldsymbol{a},\boldsymbol{y}) \log q_\psi(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y}. \tag{4.9}$$

Based on the fact that

$$D_{\mathrm{KL}}(p(\boldsymbol{a}|\boldsymbol{y}) \parallel q_\psi(\boldsymbol{a}|\boldsymbol{y})) \geq 0, \tag{4.10}$$

we have

$$\int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} \geq \int p(\boldsymbol{a},\boldsymbol{y}) \log q_\psi(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y}. \tag{4.11}$$

According to the left part of Eq. (4.11), we can obtain

$$\begin{aligned}
&\int p(\boldsymbol{a},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} \\
=&\int p(\boldsymbol{a},\boldsymbol{x},\boldsymbol{z},\hat{\boldsymbol{z}},\boldsymbol{y}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\,\mathrm{d}\hat{\boldsymbol{z}}\,\mathrm{d}\boldsymbol{y} \\
=&\int p(\boldsymbol{a},\boldsymbol{x})p(\boldsymbol{z},\hat{\boldsymbol{z}},\boldsymbol{y}|\boldsymbol{a},\boldsymbol{x}) \log p(\boldsymbol{a}|\boldsymbol{y}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\,\mathrm{d}\hat{\boldsymbol{z}}\,\mathrm{d}\boldsymbol{y}
\end{aligned}$$

Considering the Markov chain $A \to X \to Z \to \hat{Z} \to Y$, we have $p(\boldsymbol{z}, \hat{\boldsymbol{z}}, \boldsymbol{y}|\boldsymbol{a}, \boldsymbol{x}) = p(\boldsymbol{z}, \hat{\boldsymbol{z}}, \boldsymbol{y}|\boldsymbol{x})$. Since $\int p(\boldsymbol{z}, \hat{\boldsymbol{z}}, \boldsymbol{y}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{z}\,\mathrm{d}\hat{\boldsymbol{z}} = p(\boldsymbol{y}|\boldsymbol{x})$, we can obtain

$$\begin{aligned}
&\int p(\boldsymbol{a}, \boldsymbol{x}) p(\boldsymbol{z}, \hat{\boldsymbol{z}}, \boldsymbol{y}|\boldsymbol{a}, \boldsymbol{x}) \log p(\boldsymbol{a}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\,\mathrm{d}\hat{\boldsymbol{z}}\,\mathrm{d}\boldsymbol{y} \\
=&\int p(\boldsymbol{a}, \boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) \log p(\boldsymbol{a}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y} \\
=&\mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[\log p(\boldsymbol{a}|\boldsymbol{y})]\right].
\end{aligned}$$

Similarly, we can obtain

$$\int p(\boldsymbol{a}, \boldsymbol{y}) \log q_\psi(\boldsymbol{a}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{y} = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[\log q_\psi(\boldsymbol{a}|\boldsymbol{y})]\right].$$

Based on Eq. (4.11), we can obtain

$$\mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[\log p(\boldsymbol{a}|\boldsymbol{y})]\right] \geq \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[\log q_\psi(\boldsymbol{a}|\boldsymbol{y})]\right],$$

so that

$$\mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[-\log p(\boldsymbol{a}|\boldsymbol{y})]\right] \leq \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[-\log q_\psi(\boldsymbol{a}|\boldsymbol{y})]\right]. \tag{4.12}$$

The second term $I(X;\hat{Z})$ [11] is formulated as:

$$I(X;\hat{Z}) = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[D_{\mathrm{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|p(\hat{\boldsymbol{z}}))\right], \tag{4.13}$$

where the marginal probability is

$$p(\hat{\boldsymbol{z}}) = \int p(\boldsymbol{a}) p(\boldsymbol{x}|\boldsymbol{a}) p_\phi(\boldsymbol{z}|\boldsymbol{x}) p(\hat{\boldsymbol{z}}|\boldsymbol{z})\,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}. \tag{4.14}$$

We adopt a Gaussian approximation $q(\hat{\boldsymbol{z}}) \sim \mathcal{N}(\boldsymbol{0}, I)$ as an estimation for $p(\hat{\boldsymbol{z}})$ [121]. It is reasonable as the JSCC encoder generates a Gaussian distribution $p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}), \boldsymbol{\sigma}_\phi^2(\boldsymbol{x})I)$, where $\boldsymbol{\mu}_\phi(\cdot)$ and $\boldsymbol{\sigma}_\phi(\cdot)$ are functions that map the input data $\boldsymbol{x}$ to the mean and standard deviation of the Gaussian distribution.

Since $D_{\mathrm{KL}}(p(\hat{\boldsymbol{z}}) \parallel q(\hat{\boldsymbol{z}})) \geq 0$, the following upper bound can be derived:

$$I(X;\hat{Z}) \leq \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[D_{\mathrm{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|q(\hat{\boldsymbol{z}}))\right], \tag{4.15}$$

where the KL divergence can be calculated analytically by the method in [130].

Similar to Eq. (4.12), by using $q_\delta(\boldsymbol{a}|\hat{\boldsymbol{z}})$ as a variational approximation of $p(\boldsymbol{a}|\hat{\boldsymbol{z}})$, we have

$$\mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi,\theta}[-\log p(\boldsymbol{a}|\hat{\boldsymbol{z}})]\right] \leq \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[\mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi,\theta}[-\log q_\delta(\boldsymbol{a}|\hat{\boldsymbol{z}})]\right]. \tag{4.16}$$

The above extended VIB formulation determines the upper bound of the IB objective function (Eq. (4.5c)), which can be expressed as:

$$\begin{aligned}\mathcal{L}_{\text{VIB}}(\boldsymbol{a},\boldsymbol{x};\phi,\theta) = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\Big\{&\mathbb{E}_{\boldsymbol{y}|\boldsymbol{x};\phi,\theta}[-\log q_\psi(\boldsymbol{a}|\boldsymbol{y})] \\ &+\hat{\beta}_1 D_{\text{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|q(\hat{\boldsymbol{z}})) \\ &+\hat{\beta}_2 \mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi,\theta}[-\log q_\delta(\boldsymbol{a}|\hat{\boldsymbol{z}})]\Big\}. \end{aligned} \tag{4.17}$$

Through Monte Carlo sampling, we train ϕ and θ by minimizing this objective function using stochastic gradient descent. Specifically, given a mini-batch of data $\{(\boldsymbol{a}_i,\boldsymbol{x}_i)\}_{i=1}^{\Omega}$ with batch size Ω, if the reconstructed JSCC symbols $\hat{\boldsymbol{z}}$ are sampled J_1 times and the task-specific data $\boldsymbol{y}$ are sampled J_2 times for each data pair, the following estimation can be obtained:

$$\begin{aligned}\mathcal{L}_{\text{VIB}}(\boldsymbol{a},\boldsymbol{x};\phi,\theta) \cong \frac{1}{\Omega}\sum_{i=1}^{\Omega}\Bigg\{&\frac{1}{J_2}\sum_{j=1}^{J_2}[-\log q_\psi(\boldsymbol{a}_i|\boldsymbol{y}_j)] \\ &+\hat{\beta}_1 D_{\text{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}_i)\|q(\hat{\boldsymbol{z}})) \\ &+\frac{\hat{\beta}_2}{J_1}\sum_{j=1}^{J_1}[-\log q_\delta(\boldsymbol{a}_i|\hat{\boldsymbol{z}}_j)]\Bigg\}. \end{aligned} \tag{4.18}$$

4.4 JSCC Modulation

In existing communication standards, symbols are transmitted with specific constellation orders and designs. In this section, we develop a JSCC modulation scheme that can map arbitrary complex-valued JSCC symbols to a predefined constellation diagram with finite points, as shown in Fig. 4.2. In addition, we introduce a learning method to adjust the optimal constellation parameter according to the quantization loss. For clarity, we use QAM as an example. Note that our method can be easily extended to other modulation schemes.

4.4.1 Quantization and Normalization

To enable the quantization of arbitrary complex-valued JSCC symbols into a predefined constellation diagram, the following rule is applied to each symbol:

$$\bar{z}_i = Q(z_i) = \arg\min_{e_j} \|z_i - e_j\|_2^2, \tag{4.19}$$

where $z_i \in \mathbb{C}$ represents the original symbol, $\bar{z}_i \in \mathbb{C}$ represents the quantized symbol, $i \in \{1,\cdots,k\}$, $Q(\cdot)$ denotes the quantization function, and $\|\cdot\|_2$ denote the l_2-norm. $e_j \in \{e_1,\cdots,e_u\}$ represents the predefined constellation points, where $e_j \in \mathbb{C}$, and u denote the number of constellation points. This quantization operation can be extended to a vector as follows,

$$\bar{\boldsymbol{z}} = Q(\boldsymbol{z}) = [Q(z_1),\cdots,Q(z_k)]. \tag{4.20}$$

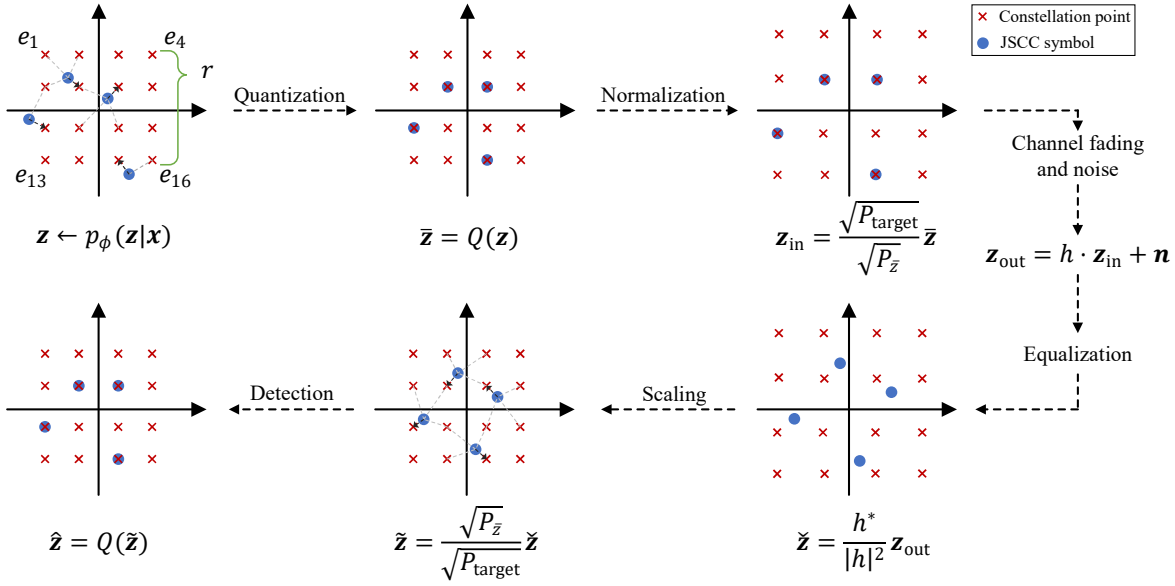


Figure 4.2: An example of the JSCC modulation and signal transmission procedure for $\boldsymbol{z} \in \mathbb{C}^4$ using 16-QAM.

Since the transmitted symbols should satisfy the average power constraint:

$$\frac{1}{k}\sum_{i=1}^{k}|\bar{z}_i|^2 \leq P_{\text{target}}, \tag{4.21}$$

the channel input (normalized symbols) are given by:

$$\boldsymbol{z}_{\text{in}} = \frac{\sqrt{P_{\text{target}}}}{\sqrt{P_{\bar{\boldsymbol{z}}}}} \cdot \bar{\boldsymbol{z}}, \tag{4.22}$$

where $P_{\bar{\boldsymbol{z}}} = \frac{1}{k}\sum_{i=1}^{k}|\bar{z}_i|^2$ denotes the power of quantized symbols $\bar{\boldsymbol{z}}$.

The channel input $\boldsymbol{z}_{\text{in}}$ is transmitted through the channel $\boldsymbol{z}_{\text{out}} = h \cdot \boldsymbol{z}_{\text{in}} + \boldsymbol{n}$. Assume that the receiver has the full CSI knowledge and knows $P_{\bar{\boldsymbol{z}}}$, in the case of the static channel, it can perform channel equalization:

$$\check{\boldsymbol{z}} = \frac{h^*}{|h|^2}\boldsymbol{z}_{\text{out}}, \tag{4.23}$$

where h^* denotes the conjugate of channel coefficient h and $\check{\boldsymbol{z}}$ denotes the equalized symbols. After equalization, the equalized symbols should be scaled as

$$\tilde{\boldsymbol{z}} = \frac{\sqrt{P_{\bar{\boldsymbol{z}}}}}{\sqrt{P_{\text{target}}}} \cdot \check{\boldsymbol{z}}, \tag{4.24}$$

where $\tilde{\boldsymbol{z}}$ denotes the scaled symbols. Then the reconstructed symbols $\hat{\boldsymbol{z}} = Q(\tilde{\boldsymbol{z}})$ can be obtained by Eq. (4.20).

4.4.2 Learnable Constellation Diagram and Fine-Tuning

Traditional modulation techniques, such as QAM, employ a lookup table that maps bits to constellation points. In contrast, the complex-valued channel symbols produced by the JSCC encoder are continuous, necessitating a different approach for their mapping.

Equation (4.19) demonstrates that the coordinates of each constellation point e_j directly affect the quantization outcome. We propose a learnable constellation diagram that adapts to the observed space of JSCC symbols, minimizes quantization loss, and improves performance with the JSCC encoder and the information reshaper. Taking u-QAM as an example, where u denotes the number of constellation points, the coordinates of each constellation point can be derived by the parameter r. This parameter specifies the distance between two constellation points located at the corners of one side, as illustrated in Fig. 4.2. Then, the real part and imaginary part of the constellation point e_j are given as follows:

$$\mathrm{Re}(e_j) = -\frac{r}{2} + \frac{(j \bmod \sqrt{u}) \cdot r}{\sqrt{u}-1}, \tag{4.25}$$

$$\mathrm{Im}(e_j) = \frac{r}{2} - \frac{\lfloor j/\sqrt{u} \rfloor \cdot r}{\sqrt{u}-1}, \tag{4.26}$$

where " mod " denotes the modulo operation and $\lfloor \cdot \rfloor$ denotes the rounding down function.

The quantization loss is defined as

$$\mathcal{L}_Q(\boldsymbol{z}; r) = \frac{1}{k} \sum_{i=1}^{k} \min_{e_j} \|z_i - e_j\|_2. \tag{4.27}$$

The training process for the learnable constellation diagram begins with the initialization of the constellation parameter r to a predefined value r_{init}, along with loading a pre-trained JSCC encoder. Using an image dataset $\mathcal{X}$ with corresponding ground truth actions $\mathcal{A}$, mini-batches are sampled iteratively during training. For each mini-batch, images are encoded into JSCC symbols, and the average batch loss is computed based on the quantization error. The constellation parameter r is then updated by backpropagation until convergence. The output of this process is the optimal constellation parameter r^*. The detailed constellation parameter training process is provided in Algorithm 2. Once the optimal r^* is obtained, the JSCC encoder and the information reshaper are jointly fine-tuned using the extended loss function:

$$\mathcal{L}_{\text{VIB-}Q}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) = \mathcal{L}_{\text{VIB}}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) + \beta_Q \mathcal{L}_Q(\boldsymbol{z}; r^*), \tag{4.28}$$

where β_Q is a hyperparameter that balances the quantization loss with the original VIB loss.

This method enhances the practical applicability of JSCC modulation by integrating it with established digital communication systems while preserving the benefits of customized encoding and decoding strategies.

The previous work [84] explores two quantization approaches for JSCC: (1) a fixed soft-to-hard quantizer, where the constellation points are predefined, and (2) a fully learnable

Algorithm 2 Training Learnable Constellation Diagram

Initialization: Initialize the constellation parameter $r \to r_{\text{init}}$, and load pre-trained JSCC encoder $p_\phi(\boldsymbol{z}|\boldsymbol{x})$.

Input: Image dataset $\mathcal{X}$ with corresponding ground truth action $\mathcal{A}$.
while not converged **do**
  Sample mini-batch $\{(\boldsymbol{a}_i, \boldsymbol{x}_i)\}_{i=1}^{\Omega}$ from $\mathcal{X}$ and $\mathcal{A}$.
  Encode image $\{\boldsymbol{x}_i\}_{i=1}^{\Omega}$ to symbols $\{\boldsymbol{z}_i\}_{i=1}^{\Omega}$.
  Compute the average batch loss $\frac{1}{\Omega}\sum_{i=1}^{\Omega} \mathcal{L}_Q(\boldsymbol{z}_i; r)$.
  Update parameter r through backpropagation.
end while
Output: Optimal constellation parameter r^*.

soft-to-hard quantizer, where each constellation point can freely adapt its position during training. Although the second method offers flexible constellation optimization, it typically results in irregular constellation arrangements that deviate from the standard square lattice structures used in practical QAM modulation schemes. Consequently, this irregularity could potentially limit compatibility with existing digital communication infrastructure, which primarily relies on standard constellations.

In contrast, the proposed method adopts a learnable constellation method while explicitly maintaining a square lattice arrangement. By constraining constellation points to remain uniformly distributed within a square lattice on the complex plane, the proposed method achieves adaptability to the data distribution while preserving compatibility with widely-deployed digital communication standards (e.g., standard QAM modulation schemes). Thus, the proposed method offers a beneficial balance between optimized, task-oriented communication performance and practical applicability within existing communication systems.

4.5 Extended VIB for Edge-based Autonomous Driving

Trajectory-Guided Control Prediction (TGCP)[1] is the state-of-the-art E2E self-driving framework that combines trajectory planning and multi-stage control prediction into a unified neural network [125]. This framework, notable for using only a monocular camera, ranks third on the CARLA leaderboard[2]. We extend VIB to TGCP to examine its applicability in an edge-based autonomous driving system.

4.5.1 Background of TGCP

TGCP on the edge server processes task-specific data $\boldsymbol{y} \in \mathbb{R}^l$ and additional state information $\boldsymbol{m}$ to make driving decisions. Note that the task-specific data $\boldsymbol{y} \in \mathbb{R}^l$ shares the same structure as the input data $\boldsymbol{x} \in \mathbb{R}^l$. In this case, $\boldsymbol{x}$ and $\boldsymbol{y}$ are RGB images. The state information $\boldsymbol{m}$ includes variables such as speed, destination coordinates, and current driving guidance (e.g.,

[1] To avoid confusion with the Transmission Control Protocol (TCP), we denote Trajectory-guided Control Prediction as TGCP in this chapter.

[2] `https://leaderboard.carla.org/leaderboard/`

"turn left" or "follow the lane"). For this study, we assume that $\boldsymbol{m}$ can be transmitted losslessly to the edge server.

The autonomous driving agent is modeled as $q_\psi(\boldsymbol{a}|\boldsymbol{y})$, which generates the inferred action $\hat{\boldsymbol{a}}$ from task-specific data $\boldsymbol{y}$. In particular, the individual components of the inferred action $\hat{\boldsymbol{a}} = (\hat{v}, \hat{s}, \hat{\boldsymbol{w}}, \hat{\boldsymbol{f}}^{\text{traj}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{f}}^{\text{ctrl}})$ are defined as follows:

- $\hat{v}$: estimated target speed.
- $\hat{s}$: value of the extracted features estimated by the expert [118].
- $\hat{\boldsymbol{w}}$: predicted waypoints from the trajectory branch.
- $\hat{\boldsymbol{f}}^{\text{traj}}$: estimated extracted features for trajectory planning.
- $\hat{\boldsymbol{b}} = [\hat{\boldsymbol{b}}_0, \hat{\boldsymbol{b}}_1, \ldots, \hat{\boldsymbol{b}}_\Gamma]$: estimated control actions from the beta distribution in the control prediction branch, where Γ denotes the prediction length.
- $\hat{\boldsymbol{f}}^{\text{ctrl}} = [\hat{\boldsymbol{f}}_0^{\text{ctrl}}, \hat{\boldsymbol{f}}_1^{\text{ctrl}}, \ldots, \hat{\boldsymbol{f}}_\Gamma^{\text{ctrl}}]$: predicted informative features of the control prediction branch.

4.5.2 Control and Trajectory Prediction Loss Functions

The designed controller, based on [125], computes control commands such as throttle, steer, and brake using the output of the trajectory and control prediction branches. The corresponding loss functions are defined as:

$$\mathcal{L}_{\text{traj}} = \|\boldsymbol{w} - \hat{\boldsymbol{w}}\|_1 + \lambda_{\text{feat}} \|\boldsymbol{f}^{\text{traj}} - \hat{\boldsymbol{f}}^{\text{traj}}\|_2, \tag{4.29}$$

$$\begin{aligned}\mathcal{L}_{\text{ctrl}} =& D_{\text{KL}}(\mathcal{B}e(\boldsymbol{b}_0) \| \mathcal{B}e(\hat{\boldsymbol{b}}_0)) \\ &+ \frac{1}{\Gamma} \sum_{i=1}^{\Gamma} D_{\text{KL}}(\mathcal{B}e(\boldsymbol{b}_i) \| \mathcal{B}e(\hat{\boldsymbol{b}}_i)) \\ &+ \lambda_{\text{feat}} \| \boldsymbol{f}_0^{\text{ctrl}} - \hat{\boldsymbol{f}}_0^{\text{ctrl}} \|_2 + \frac{1}{\Gamma} \sum_{i=1}^{\Gamma} \| \boldsymbol{f}_i^{\text{ctrl}} - \hat{\boldsymbol{f}}_i^{\text{ctrl}} \|_2,\end{aligned} \tag{4.30}$$

where λ_{feat} is a hyperparameter, $\boldsymbol{w}$, $\boldsymbol{f}^{\text{traj}}$, $\boldsymbol{b}_i$, and $\boldsymbol{f}_i^{\text{ctrl}}$ are from the ground truth action $\boldsymbol{a}$, $\|\cdot\|_1$ denotes the l_1-norm, and $\mathcal{B}e(\cdot)$ denotes the beta distribution.

Furthermore, the auxiliary loss function is defined as:

$$\mathcal{L}_{\text{aux}} = \|v - \hat{v}\|_1 + \|s - \hat{s}\|_2, \tag{4.31}$$

where speed v and value of features s are from the ground truth action $\boldsymbol{a}$. Combining these terms, the overall loss function $\mathcal{L}_{\text{TCGP}}$ becomes:

$$\mathcal{L}_{\text{TCGP}} = \lambda_{\text{traj}} \mathcal{L}_{\text{traj}} + \lambda_{\text{ctrl}} \mathcal{L}_{\text{ctrl}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}, \tag{4.32}$$

where λ_{traj}, λ_{ctrl}, and λ_{aux} are hyperparameters.

4.5.3 Task-Oriented End-to-End Training

Typically, we assume that the posterior $q_\psi(\boldsymbol{a}|\boldsymbol{y})$ follows a Gaussian distribution

$$\mathcal{N}(\boldsymbol{\mu}_\psi(\boldsymbol{y}), \boldsymbol{\Sigma}_\psi(\boldsymbol{y})), \tag{4.33}$$

where $\boldsymbol{\mu}_\psi(\boldsymbol{y}) \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_\psi(\boldsymbol{y}) = \sigma_c^2 I_d$ (σ_c is a constant). Thus, we can derive

$$\begin{aligned}
&-\log q_\psi(\boldsymbol{a}|\boldsymbol{y}) \\
=&-\log \mathcal{N}\left(\boldsymbol{\mu}_\psi(\boldsymbol{y}), \boldsymbol{\Sigma}_\psi(\boldsymbol{y})\right) \\
=&-\log\left[\exp\left(-\frac{1}{2}(\boldsymbol{a}-\boldsymbol{\mu}_\psi(\boldsymbol{y}))^T \boldsymbol{\Sigma}_\psi^{-1}(\boldsymbol{y})(\boldsymbol{a}-\boldsymbol{\mu}_\psi(\boldsymbol{y}))\right)\right] + \log\left[(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_\psi(\boldsymbol{y})|^{\frac{1}{2}}\right] \\
=&\frac{\left\|\boldsymbol{a}-\boldsymbol{\mu}_\psi(\boldsymbol{y})\right\|_2^2}{2\sigma_c^2} + d\log\sigma_c + \frac{d}{2}\log 2\pi.
\end{aligned}$$

Since σ_c is a constant, we have

$$-\log q_\psi(\boldsymbol{a}|\boldsymbol{y}) \sim \frac{1}{2\sigma_c^2}\|\boldsymbol{a}-\boldsymbol{\mu}_\psi(\boldsymbol{y})\|_2^2, \tag{4.34}$$

where $\boldsymbol{\mu}_\psi(\boldsymbol{y}) = \hat{\boldsymbol{a}}$. Eq. (4.34) shows that $-\log q_\psi(\boldsymbol{a}|\boldsymbol{y})$ can serve as a distance metric, like the l_2-norm. Since $\mathcal{L}_{\text{TCGP}}$ is a combination of distance metric of action $\boldsymbol{a}$ (l_1-norm, l_2-norm, and KL divergence), we heuristically propose substituting the first term in Eq. (4.17) with $\mathcal{L}_{\text{TCGP}}$ to adapt the objective function as

$$\begin{aligned}
\mathcal{L}'_{\text{VIB}}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\Big\{&\mathcal{L}_{\text{TCGP}} \\
&+ \hat{\beta}_1 D_{\text{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|q(\hat{\boldsymbol{z}})) \\
&+ \hat{\beta}_2 \mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi,\theta}[-\log q_\delta(\boldsymbol{a}|\hat{\boldsymbol{z}})]\Big\}.
\end{aligned} \tag{4.35}$$

In addition, the Eq. (4.28) can be modified as:

$$\mathcal{L}'_{\text{VIB-}Q}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) = \mathcal{L}'_{\text{VIB}}(\boldsymbol{a}, \boldsymbol{x}; \phi, \theta) + \beta_Q \mathcal{L}_Q(\boldsymbol{z}; r^*). \tag{4.36}$$

Training of JSCC encoder and information reshaper consists of two stages: pre-training and fine-tuning. In pre-training, the neural network parameters (ϕ and θ) are initialized, and images from the dataset are encoded into JSCC symbols, transmitted through a channel without a fixed constellation, and reshaped into task-specific data. The TGCP model, with frozen parameters, generates inferred actions $\hat{\boldsymbol{a}}$, and the loss $\mathcal{L}'_{\text{VIB}}$ is computed to update the network parameters. Fine-tuning follows a similar process, but the symbols are transmitted with JSCC modulation, and the loss $\mathcal{L}'_{\text{VIB-}Q}$ is used for parameter updates. Finally, the optimized parameters ϕ and θ are output. The training process of the proposed aligned task- and reconstruction-oriented JSCC encoder and information reshaper is shown in Algorithm 3. Here, CH$(\cdot)$ denotes the function of a JSCC modulation and communication channel, while TGCP$(\cdot)$ denotes the function of TGCP. Specifically, during the fine-tuning process, both

the JSCC encoder and the information reshaper are actively adjusted, which means that neither component is frozen. This fine-tuning process reduces the quantization loss of the encoder's output and preserves task-critical information, showing the potential for real-world applications.

Algorithm 3 Training JSCC Encoder and Information Reshaper.

Initialization: Initialize the neural network parameters ϕ and θ.

Input: Image dataset $\mathcal{X}$ with corresponding ground-truth agent output $\mathcal{A}$. Well-trained TGCP model with frozen parameters. Learning rate η.
while not converged **do**
  Sample mini-batch $\{(\boldsymbol{a}_i, \boldsymbol{x}_i)\}_{i=1}^{\Omega}$ from $\mathcal{A}$ and $\mathcal{X}$.
  for sample $i = 1, \ldots, \Omega$ **do**
    Encode image $\boldsymbol{x}_i$ to JSCC symbols $\boldsymbol{z}_i$.
    Transmit JSCC symbols through a channel without JSCC modulation: $\hat{\boldsymbol{z}}_i \leftarrow \mathrm{CH}(\boldsymbol{z}_i)$.
    Reshape the reconstructed JSCC symbols $\hat{\boldsymbol{z}}_i$ to task-specific data $\boldsymbol{y}_i$.
    Generate inferred action: $\hat{\boldsymbol{a}}_i \leftarrow \mathrm{TGCP}(\boldsymbol{y}_i)$.
    Compute loss $\mathcal{L}'_{\text{VIB}}$ based on Eq. (4.35).
  end for
  Update parameters (pre-training): $\phi \overset{+}{\leftarrow} -\eta \cdot \nabla_{\phi} \mathcal{L}'_{\text{VIB}}$, $\theta \overset{+}{\leftarrow} -\eta \cdot \nabla_{\theta} \mathcal{L}'_{\text{VIB}}$.
end while
Find optimal constellation parameter r^* according to Algorithm 2.
while not converged **do**
  Sample mini-batch $\{(\boldsymbol{a}_i, \boldsymbol{x}_i)\}_{i=1}^{\Omega}$ from $\mathcal{A}$ and $\mathcal{X}$.
  for sample $i = 1, \ldots, \Omega$ **do**
    Encode image $\boldsymbol{x}_i$ to JSCC symbols $\boldsymbol{z}_i$.
    Transmit JSCC symbols through a channel with JSCC modulation: $\hat{\boldsymbol{z}}_i \leftarrow \mathrm{CH}(\boldsymbol{z}_i)$.
    Reshape the reconstructed JSCC symbols $\hat{\boldsymbol{z}}_i$ to task-specific data $\boldsymbol{y}_i$.
    Generate inferred action: $\hat{\boldsymbol{a}}_i \leftarrow \mathrm{TGCP}(\boldsymbol{y}_i)$.
    Compute loss $\mathcal{L}'_{\text{VIB-}Q}$ based on Eq. (4.36).
  end for
  Update parameters (fine-tuning): $\phi \overset{+}{\leftarrow} -\eta \cdot \nabla_{\phi} \mathcal{L}'_{\text{VIB-}Q}$, $\theta \overset{+}{\leftarrow} -\eta \cdot \nabla_{\theta} \mathcal{L}'_{\text{VIB-}Q}$.
end while
Output: Neural network parameters: ϕ and θ.

4.6 Performance Evaluation

4.6.1 Experiment Setup

Dataset

We utilize the Car Learning to Act (CARLA) simulator, an open-source platform designed for autonomous driving research [124], which provides a variety of urban environments that simulate real-world traffic scenarios. The image dataset from [125], comprising images from various urban maps, serves as the input data $\boldsymbol{x}$ for our training. In our experiments, the training dataset consists of 189,524 images from four maps: Town01, Town03, Town04, and Town06. The test dataset includes 27,201 images from another four maps: Town02, Town05, Town07, and Town10.

Evaluation Metrics

Our evaluation focuses on comparing the driving performance of our ATROC framework against various baselines within the CARLA simulator. We use the commonly adopted driving score[3] to assess the vehicle's ability to navigate according to predetermined waypoints, destinations, and comply with traffic regulations. Each test is conducted three times in Town05 under four different weather scenarios: clear noon, cloudy sunset, soft rain at dawn, and hard rain at night.

Basic Settings

In our proposed framework, we configure the JSCC symbols dimension k to 1024, enabling us to achieve significantly low bits per service of 6144 for 64-QAM. For training of the JSCC encoder and the information reshaper, we set the Lagrange multipliers $\hat{\beta}_1 = 1$, $\hat{\beta}_2 = 8192$, and the quantization loss hyperparameter $\beta_Q = 10$, which is a good balance between fidelity and compression. Furthermore, we set the learning rate η to 4×10^{-5}, and impose a power constraint with $P_{\text{target}} = 1$.

The TGCP model is trained following the instructions of [125]. The parameters ψ of the TGCP model are kept fixed throughout all training phases to ensure full compatibility with the existing edge intelligence infrastructure. Specifically, our goal is for the edge server's inference network to effectively handle both conventional (normal) data $\boldsymbol{x}$ and the task-specific data $\boldsymbol{y}$, without requiring retraining or fine-tuning of the inference model itself. By fixing the TGCP parameters, we explicitly demonstrate that our proposed method (e.g., the JSCC encoder and information reshaper) does not change the edge inference architecture, highlighting its compatibility with existing systems. This pre-trained TGCP model serves as the AI agent in the following experiments.

[3] `https://leaderboard.carla.org/`

For simplicity, a deterministic information reshaper is used, allowing us to approximate $q_\delta(\boldsymbol{a}|\hat{\boldsymbol{z}})$ by $q_\psi(\boldsymbol{a}|\boldsymbol{y})$. The architecture and detailed parameters of the proposed JSCC encoder and the information reshaper are shown in Fig. 4.3.

In addition, we introduce a performance metric *bits per service* to measure communication efficiency, which is defined as $k \cdot c$, where c represents bits per symbol.

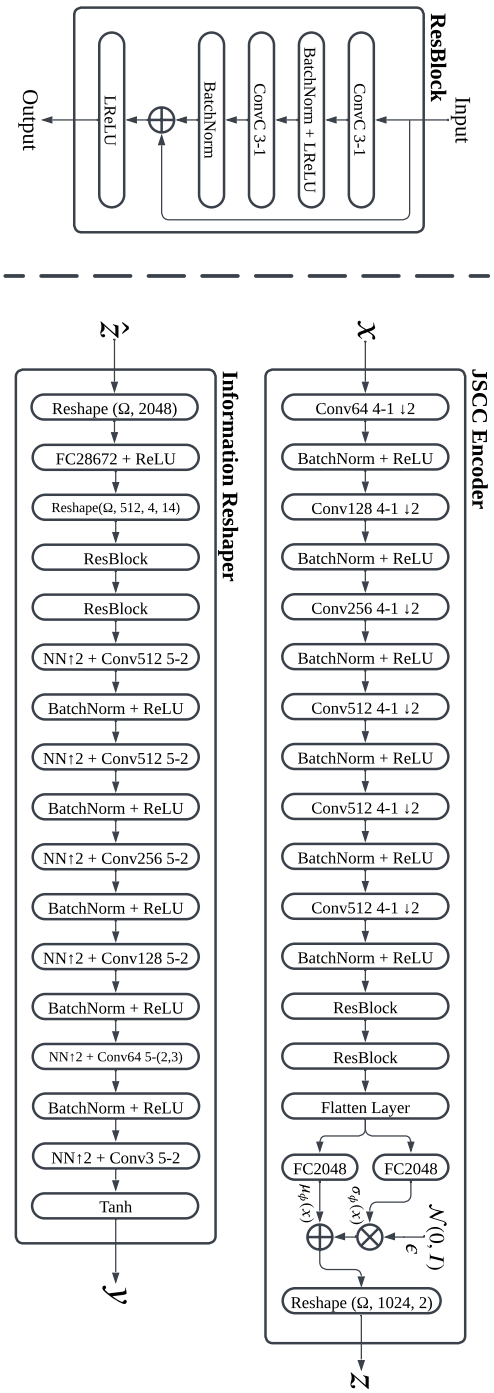


Figure 4.3: Architecture of the proposed JSCC encoder and information reshaper. For example, *ConvC 3-1* represents a convolutional layer with C channels, a 3×3 kernel size, and padding of 1 on both sides. $\downarrow 2$ denotes the strided down convolutions, while NN$\uparrow$2 denotes the nearest neighbor upsampling. *FC2048* refers to a fully connected layer with an output size of 2048. *BatchNorm* denotes batch normalization, *LReLU* represents the leaky ReLU activation with $\alpha = 0.2$, and Ω represents the batch size. The dimensions (number of channels) of the inputs and outputs for the *ResBlock* remain unchanged.

Baseline Methods

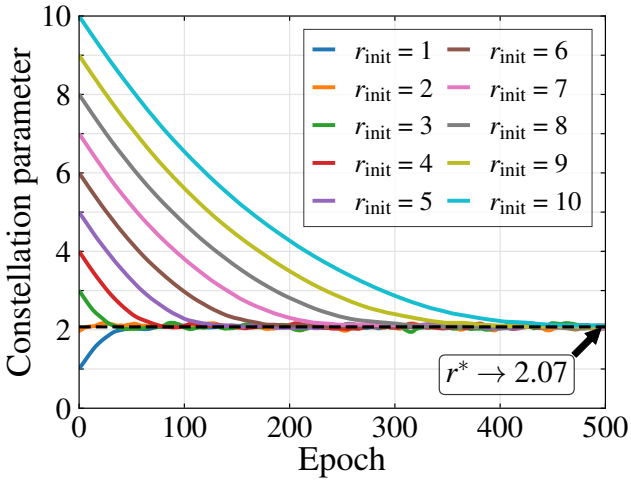
Three traditional image coding methods are included as baseline methods for comparison: (1) JPEG [3]; (2) JPEG2000 [4]; (3) and BPG [131]. Each traditional image coding method is followed by (2048, 6144) LDPC codes combined with a 64-QAM digital modulation scheme. The average bits per service for these methods range from 36,844 to 1,041,758.

In addition, baseline methods also include two state-of-the-art reconstruction-oriented JSCC designs, with the legends "ROC-AE" [14] and "ROC-VAE" [15], which represent traditional autoencoder and variational autoencoder approaches. Note that the training dataset for the ROC-AE, ROC-VAE, ATROC, and pre-trained TGCP is identical. For a fair comparison, ROC-AE, ROC-VAE, and ATROC use the same network structure, resulting in the same bits per service (i.e., 6144). In particular, ROC-VAE and ROC-AE are also fine-tuned by our proposed JSCC modulation scheme for 64-QAM, where the optimal constellation parameters r^* are 4 and 50.4, respectively.

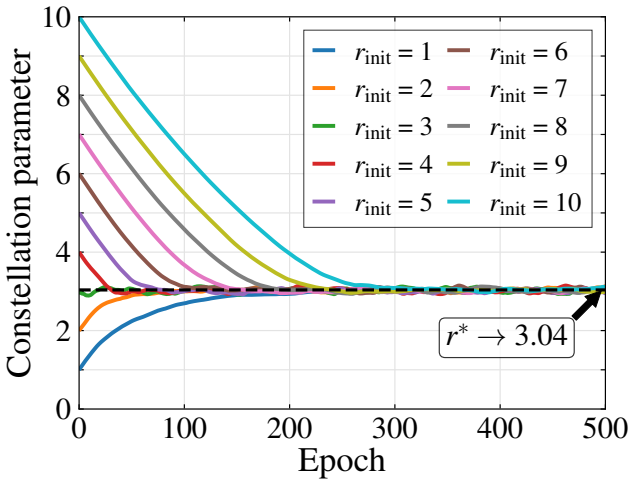
4.6.2 Results of JSCC Modulation

The constellation parameter r is trained using a pre-trained JSCC encoder, as described in Algorithm 2. Figure 4.4 demonstrates that regardless of the initial value of the constellation parameter, $r_{\text{init}} \in \{1, \cdots, 10\}$, the optimal constellation parameter r^* consistently converges, validating the effectiveness of the proposed modulation approach.

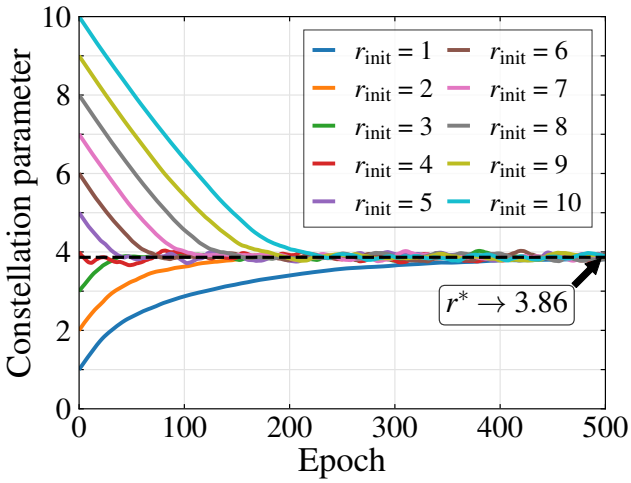
Driving scores from different fine-tuned models across various constellation parameters based on 64-QAM are presented in Fig. 4.5. The model fine-tuned with the optimal constellation parameter r^* outperforms other models under the AWGN channel with SNRs range from -10 dB to 10 dB, showcasing the superiority of our proposed JSCC modulation scheme.



(a) Training of 16-QAM



(b) Training of 64-QAM.



(c) Training of 256-QAM.

Figure 4.4: Training of the constellation parameter for 16-QAM, 64-QAM, and 256-QAM. Regardless of the initial value of the constellation parameter, the optimal value consistently converges.

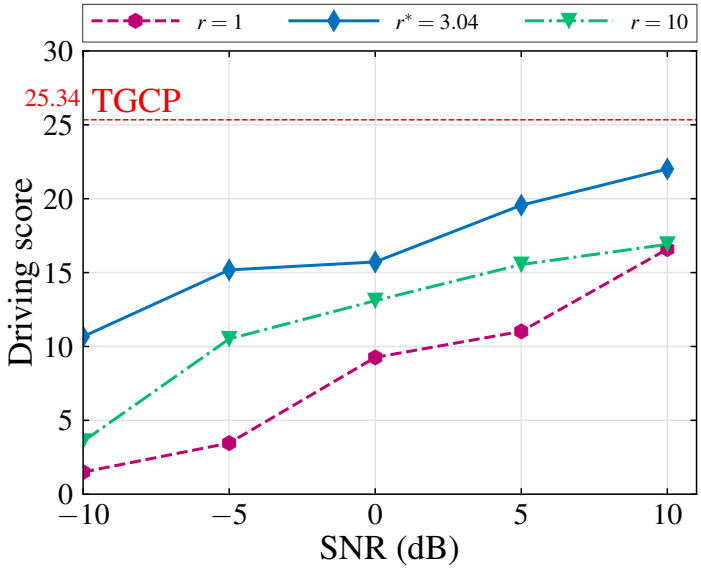


Figure 4.5: Driving score of fine-tuned models based on 64-QAM with different constellation parameters ($r \in \{1, r^*, 10\}$, where $r^* = 3.04$) under the AWGN channel with SNR range from -10 dB to 10 dB.

4.6.3 Evaluation on CARLA

The impact of bits per service on the driving score is illustrated in Fig. 4.6, where the driving score of TGCP using raw images without communication is 25.34. Notably, our proposed method achieves significant bandwidth savings (99.19% compared to existing methods) while maintaining a required driving score of 20 under both the AWGN and Rayleigh fading channels. This substantial reduction in bits per service not only illustrates the efficiency of our approach but also underscores its capability to operate effectively under stringent bandwidth constraints.

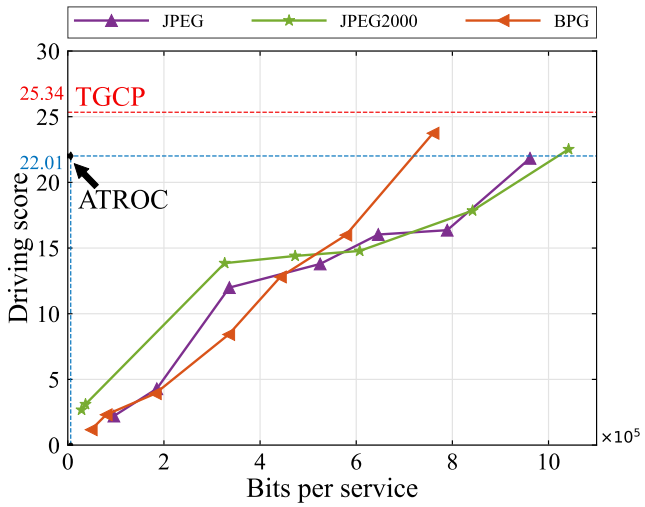
Detailed results for specific channel conditions are shown in Fig. 4.7a and Fig. 4.7b, demonstrating the dependency of driving scores on SNR. For traditional image coding methods, such as JPEG, JPEG2000, and BPG, we apply two configurations for comparison: (1) the average bits per service are set to 961,484 for JPEG*, 1,041,758 for JPEG2000*, and 759,683 for BPG*; (2) the average bits per service are reduced to 524,996 for JPEG$^-$, 472,958 for JPEG2000$^-$, and 442,152 for BPG$^-$. The first configuration highlights the optimal performance of traditional image coding methods, as shown in Fig. 4.6a and Fig. 4.6b. In contrast, the second configuration approximately halves the bits per service from the first, as a basis for further comparison. In contrast, our method requires only 6144 bits per service, highlighting its superior compression and transmission efficiency. In addition, we compare the proposed method ATROC with the state-of-the-art reconstruction-oriented JSCC designs using the same neural network structure, with the legends “ROC-AE” [14] and “ROC-VAE” [15].

Under AWGN channel conditions, our method significantly outperforms reconstruction-oriented communication methods with driving scores of 15.72 at SNR = 0 dB, 15.18 at SNR = -5 dB, and 10.67 at SNR = -10 dB, as shown in Fig. 4.7a. Traditional methods (JPEG*,

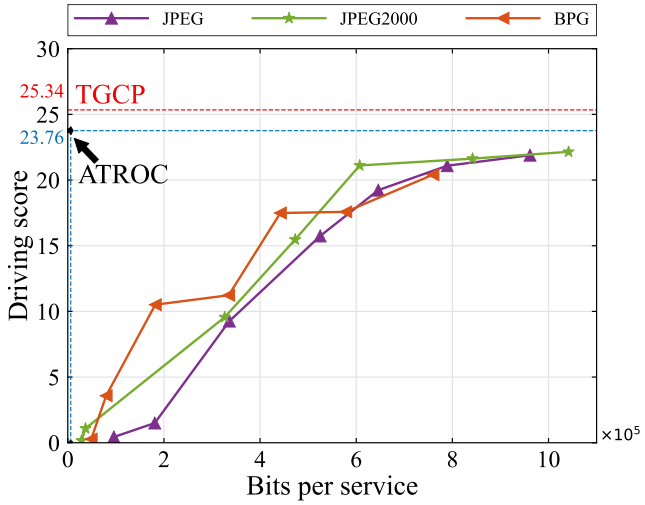
JPEG^{-}, JPEG2000*, JPEG2000^{-}, BPG*, and BPG^{-},) show a dramatic decline in the driving score as SNR decreases (below 0 dB), emphasizing the robustness of our ATROC framework under challenging conditions.

Similarly, in Rayleigh fading channel scenarios (Fig. 4.7b), our proposed method continues to demonstrate superior performance with driving scores of 18.57 at SNR = 10 dB, 13.1 at SNR = 5 dB, and 12.73 at SNR = 0 dB. However, traditional methods experience significant performance degradation when SNR is below 10 dB.

Moreover, JSCC-based reconstruction-oriented communication methods perform poorly under this extremely limited communication bandwidth, as these methods fail to preserve task-specific information.

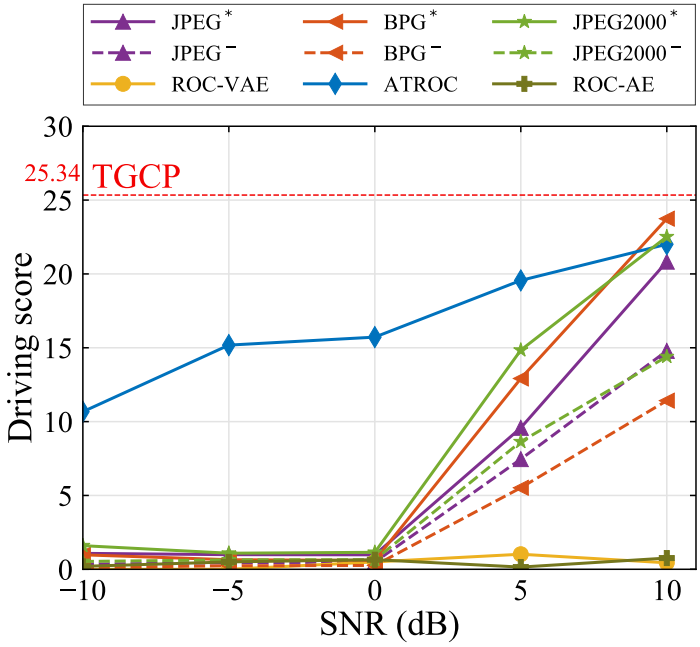


(a) AWGN channel with SNR = 10 dB.

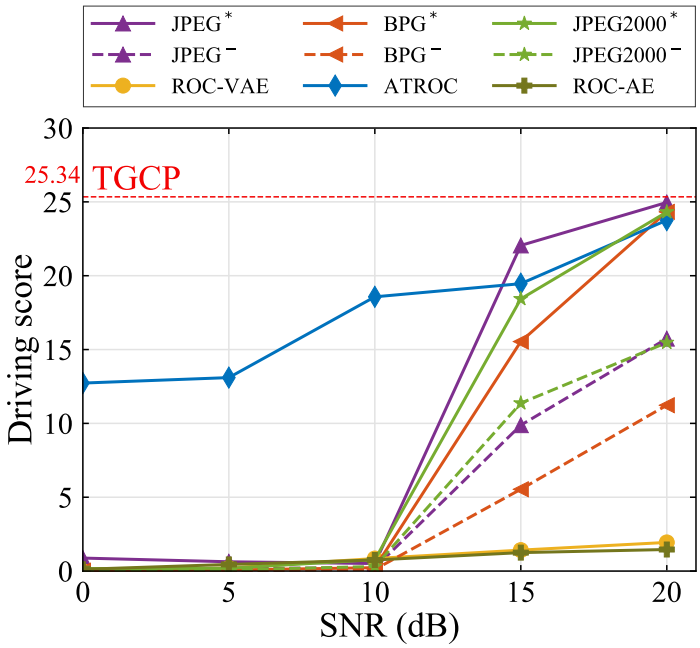


(b) Rayleigh fading channel with SNR = 20 dB.

Figure 4.6: Driving score of traditional reconstruction-oriented communication with varied bits per service under AWGN channel and Rayleigh channel. The ATROC with 6144 bits per service serves as a baseline for comparison across both channel conditions. In addition, the TCGP using raw RGB images (5.5296×10^6 bits per service) for autonomous driving is also included as a baseline.



(a) AWGN channel with SNRs range from -10 dB to 10 dB.



(b) Rayleigh channel with SNRs range from 0 dB to 20 dB.

Figure 4.7: Driving score with varied SNRs under AWGN channel and Rayleigh channel.

These findings are further supported by qualitative analysis, as illustrated in Fig. 4.8. JSCC-based reconstruction-oriented communication methods, while capable of producing high-quality image reconstructions suitable for human vision, often fail to retain crucial task-specific information, such as vehicles, cyclists, road markers, and traffic lights. This deficiency leads to poor performance in edge-based autonomous driving applications, where precise detection of such elements is critical for safety and efficiency. In contrast, our proposed method can effectively preserve task-specific information, shown in the blue, red, and purple boxes of Fig. 4.8. To reduce the required bits per service, it ignores task-agnostic information, shown in the green box of Fig. 4.8. Moreover, our proposed method demonstrates remarkable

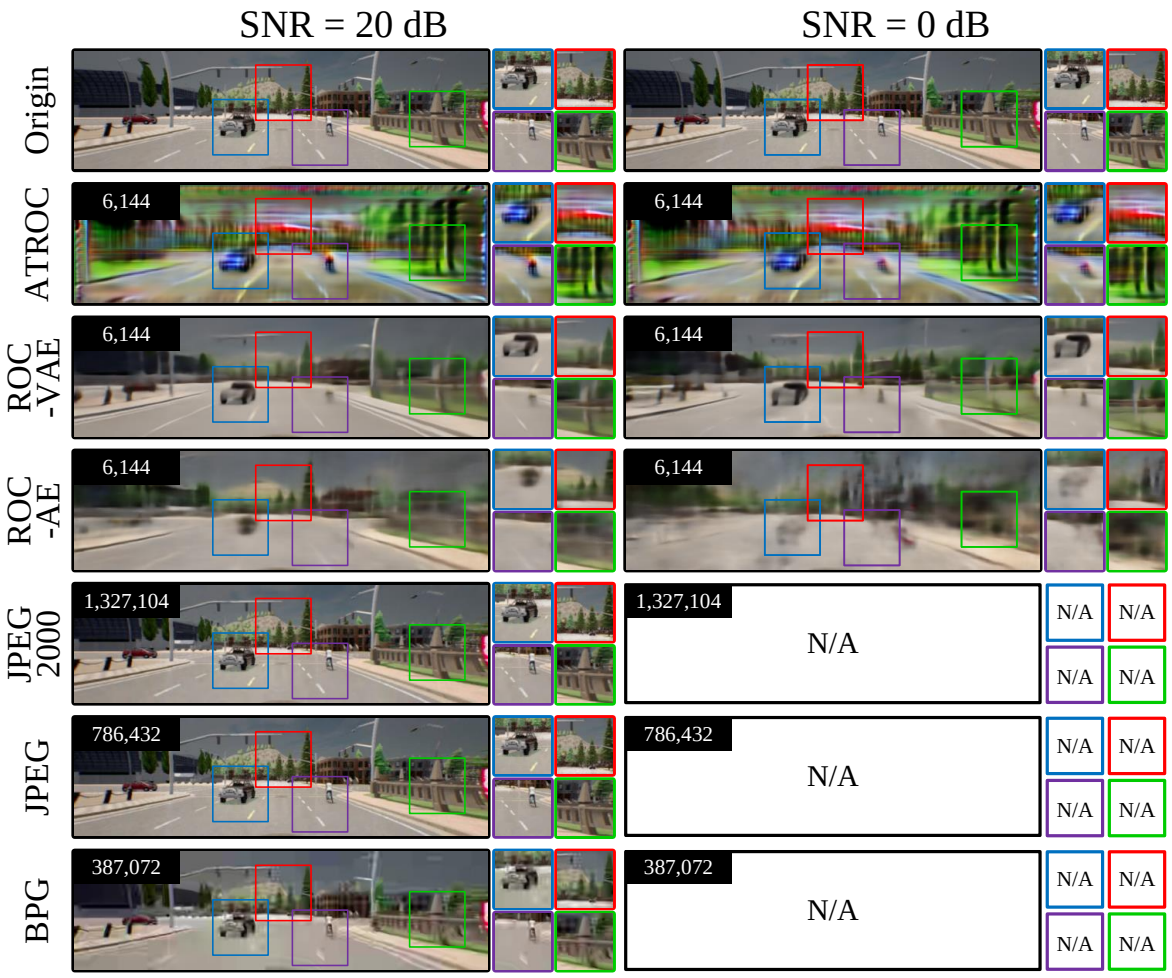


Figure 4.8: A qualitative example of our proposed method and baseline methods under Rayleigh fading channel with SNR = 20 dB and SNR = 0 dB. The bits per service of each image are provided in the upper left corner. The details in the reconstructed image are highlighted on the right side of the image. 1) blue box: vehicle and road marks; 2) red box: traffic lights; 3) purple box: cyclist and road marks; 4) green box: fence in the distance. Since traditional reconstruction-oriented communication methods (JPEG, JPEG2000, and BPG) fail to reconstruct images when SNR = 0 dB, we use "N/A" (Not Applicable) to represent the corrupted images.

noise resistance under low SNR conditions. It effectively preserves task-specific information, maintaining its completeness even in challenging communication environments.

Furthermore, in Table 4.2, we evaluate additional performance metrics such as PSNR, MS-SSIM, and FID, which are typically used to assess image quality from a human perspective. The divergence in performance metrics between traditional reconstruction-oriented methods and our proposed method highlights the necessity of a communication design that prioritizes machine vision, particularly in applications where decision-making accuracy is critical.

Table 4.2: Human Perceptional Metrics

Method	$k \cdot c$	PSNR(dB) ↑	MS-SSIM ↑	FID↓
JPEG	961484	34.56	0.99	5.83
JPEG2000	1041758	37.54	0.99	7.17
BPG	759683	34.93	0.98	6.68
ROC-AE	6144	17.24	0.41	200.59
ROC-VAE	6144	21.75	0.72	135.68
TOC	6144	11.43	0.27	268.14

4.7 Conclusion

This chapter has investigated an ATROC framework for edge intelligence, aimed at improving the integration of AI technologies within existing communication infrastructures. By extending the IB theory and incorporating JSCC modulation, our framework shifts the focus from traditional signal reconstruction fidelity to task relevance, thus optimizing the performance of AI-driven applications in bandwidth-constrained and noise-interference environments.

Our evaluations conducted within the CARLA simulator highlight the robustness of the proposed ATROC framework. Particularly in low SNR conditions, our framework demonstrated significant superiority over traditional reconstruction-oriented communication methods by achieving a reduction of up to 99.19% bits per service without sacrificing the effectiveness of task execution.

The qualitative analysis revealed that while reconstruction-oriented communication methods are effective for human visual perception, they often fail to satisfy the specific requirements of machine vision. This observation emphasizes the need for communication designs that align more closely with the specific information needs of AI systems rather than human interpretation.

Chapter 5

Toward Holistic Systems: Task-Oriented Co-design of Communication, Computing, and Control for Cyber-Physical Systems

5.1 Introduction

In industrial CPS, ensuring URLLCs is crucial for achieving reliable real-time performance [44]. Applications such as automated transportation, material handling, and inspection increasingly rely on autonomous vehicles and robots within factories, warehouses, and hazardous environments. Autonomous driving plays an important role in these systems, enabling the automation of essential tasks, optimizing workflow efficiency, and improving safety [132].

To meet the stringent requirements of URLLC (e.g., the E2E delay should be less than 1 ms and the packet loss probability should be less than 10^{-5} [133]), edge inference has emerged as a promising solution [134]. By minimizing the physical distance between data generation and processing, edge inference significantly reduces latency, which is vital for autonomous systems that must respond timely to dynamic environmental changes, such as navigating unpredictable factory layouts or reacting to sudden obstacles. The primary motivation for using edge computing (also called off-board computing) lies in its ability to enhance system flexibility, particularly given the differing innovation cycles of the automotive and semiconductor industries. While a vehicle's lifespan typically ranges from 10 to 20 years, advancements in computing capabilities can be significant within this period. By leveraging edge computing within shared telecommunication infrastructure, the vehicle can enjoy much better flexibility in upgrading the computing power and software throughout their entire lifetime. However, the transmission of massive amounts of sensor and video data presents a challenge to the edge's ability to handle real-time processing while maintaining the reliability and low-latency communications. In that case, edge inference, often powered by DNNs, is still affected by nontrivial communication latency and bandwidth constraints, particularly under the demands of URLLC in industrial CPS [135].

Recent developments in deep learning have introduced JSCC as a promising solution to the limited communication bandwidth and significant noise interference [14]. Unlike traditional separate coding designs, JSCC integrates source and channel coding, improving data transmission efficiency. Despite these advantages, conventional JSCC approaches typically focus on accurate signal reconstruction at the receiver, potentially wasting communication resources on task-agnostic information that does not directly contribute to the decision-making process [79]. This inefficiency has attracted significant research interest in task-oriented communication, a technology designed to prioritize the transmission of task-specific information, thus reducing data rates and improving efficiency, especially for AI-driven applications [65].

Furthermore, data sent to the edge server become outdated due to uplink delays, including processing, transmission, propagation, and queueing delay, negatively impacting the timeliness of edge inference results. This issue is further exacerbated by downlink, computing, and control delays. The E2E delay (round-trip delay) degrades system performance and makes it difficult to meet the URLLC requirements in industrial CPS. Prediction-based methods can mitigate perceived E2E delay [136], but longer prediction horizons increase the risk of inaccuracy, creating a trade-off between minimizing delay and ensuring reliability.

Given the limitations of traditional approaches that design communication, computing, and control components separately, an integrated task-oriented co-design framework becomes essential [11], [137], [138]. For example, the work in [139] introduces a joint learning and communication framework in which agents learn both actions and communication strategies over noisy channels, specifically designed to enable effective coordination and control in Multi-Agent Reinforcement Learning (MARL) environments. Similarly, [140] provides a forward-looking perspective on communication in 6G systems, highlighting the importance of timely and task-aware information exchange in intelligent networked systems. These contributions align closely with the motivation of this chapter, which explores E2E co-design of task-oriented communication and control under real-world constraints such as communication delays and bandwidth limitations.

In this chapter, our objective is to address three fundamental questions for edge-enabled mission-critical industrial CPS: 1) How can data transmission be optimized for bandwidth-constrained and latency-sensitive applications to ensure that task-specific information is prioritized? 2) How can predictive models be utilized to ensure that edge inference systems make decisions that reduce perceived E2E delay? 3) How can communication, computing, and control be jointly designed and optimized to meet the demands of URLLC in mission-critical applications? The key contributions of this chapter are summarized as follows:

- We develop a comprehensive task-oriented co-design framework that jointly optimizes communication, computing, and control. This framework seamlessly integrates task-oriented JSCC with a delay-aware autonomous driving agent, addressing the critical challenges of bandwidth constraints, noise interference, and E2E delay to maximize performance for edge-enabled autonomous driving.

- We formulate the problem of task-oriented communication using the IB approach and employ a variational approximation to derive a tractable upper bound, resulting in the VIB method. Additionally, we extend the standard VIB framework to incorporate conditional information, such as vehicle and channel state information, ensuring better alignment with mission-critical applications. Furthermore, we handle the KL-divergence term using a concise approach inspired by [121]. Our formulation improves communication efficiency in dynamic and noisy environments, which is essential for the reliable operation of industrial CPS.

- We establish the DTCP strategy for autonomous driving, which uniquely combines two dominant autonomous driving paradigms: trajectory planning and control prediction. The DTCP processes JSCC symbols, state information, and channel state to predict optimal driving actions that reduce perceived E2E delay. In addition, DTCP is specially co-designed with the task-oriented JSCC and is jointly trained for machine-to-machine communication.

The rest of this chapter is organized as follows. We detail the fundamentals of predictions in URLLC applications in Section 5.2. In Section 5.3, we introduce the system model and formulate the variational problem. Section 5.5 presents the details of DTCP and the proposed co-design with task-oriented JSCC. The numerical and experimental results are provided in Section 5.6. Finally, conclusions are drawn in Section 5.7.

The main notations used throughout the chapter are summarized in Table 5.1. To improve readability and manage the complexity of the joint design of communication, computation, and control, the temporal subscript of the notation is omitted in Section 5.3.

5.2 Predictions in URLLC Applications

Ultra-Reliable and Low-Latency Communications (URLLC) form the cornerstone of mission-critical services in modern and future wireless networks, including autonomous driving, telesurgery, industrial automation, and the Tactile Internet. These applications demand unprecedented E2E delay limits (often below 1 ms) and ultra-high reliability with packet loss probabilities in the range of 10^{-5} to 10^{-7} [44]. Achieving such stringent requirements, especially under the unpredictable and dynamic conditions of wireless environments, requires the use of accurate and real-time prediction mechanisms. Predictions in URLLC systems play a dual role: first, they proactively mitigate the impact of variable conditions such as channel fading, network congestion, and mobility; second, they enable anticipatory resource allocation and control strategies that prevent latency violations and packet losses before they occur [141].

Traditional model-driven optimization techniques often fall short in meeting URLLC demands due to their reliance on idealized assumptions and limited tractability in real-time scenarios. For example, queueing delays, access delays, and processing delays are stochastic and highly sensitive to network load and topology [60]. Moreover, conventional methods

Table 5.1: Summary of Main Symbols

$\boldsymbol{x}$	Input image (data)	K_b	Size of mini-batch
X	Random variable of $\boldsymbol{x}$	$\boldsymbol{x}_t$	Image captured by a camera at time t
$\mathcal{X}$	Space of $\boldsymbol{x}$	$\boldsymbol{z}_t^{l(t)}, \boldsymbol{z}_t^l$	JSCC symbols with length l transmitted at time t
$\boldsymbol{z}$	Transmitted JSCC symbols	$\hat{\boldsymbol{z}}_{t-\delta_{eu}}^l$	Reconstructed JSCC symbols with length l received at time t
Z	Random variable of $\boldsymbol{z}$	$\boldsymbol{m}_t$	State information transmitted at time t
$\hat{\boldsymbol{z}}$	Reconstructed JSCC symbols	τ	Duration of time slot
$\mathcal{Z}$	Space of JSCC symbols	D_t	Index set for $\boldsymbol{z}_t^l$
$\check{\boldsymbol{z}}$	Received JSCC symbols	δ_e	Computing delay of JSCC encoding
h	Channel state	δ_u	Uplink communication delay
$\boldsymbol{n}$	Gaussian noise	δ_a	Computing delay of agent
H	Random variable of h	δ_d	Downlink communication delay
$\mathcal{H}$	Space of h	δ_c	Control delay
$\boldsymbol{m}$	State information	δ_{eu}	Combined delay of δ_e and δ_u
M	Random variable of $\boldsymbol{m}$	δ	End-to-end delay
$\mathcal{M}$	Space of $\boldsymbol{m}$	δ_T	Delay threshold
$\boldsymbol{a}$	Ground-truth action	l_p	Prediction horizon
A	Random variable of $\boldsymbol{a}$	l_w	Extra Prediction Horizon
$\hat{\boldsymbol{a}}$	Estimated action	$\boldsymbol{r}_{t-\delta_{eu}}^{\text{traj}}$	Trajectory feature corresponding to $x_{t-\delta_{eu}}$
$\hat{A}$	Random variable of $\hat{\boldsymbol{a}}$	$\mathcal{R}_{\text{traj}}$	Space of trajectory features
$\mathcal{A}$	Space of action	$\boldsymbol{h}_{t-\delta_{eu}}^{\text{traj}}$	Trajectory hidden state corresponding to $x_{t-\delta_{eu}}$
$f_e(\cdot)$	Function of JSCC encoder	$\boldsymbol{w}_{t-\delta_{eu}}$	Planned waypoint at time $t-\delta_{eu}$
$f_h(\cdot)$	Function of noisy fading channel	$\boldsymbol{c}_{t-\delta_{eu}+l_p}^{\text{traj}}$	Predicted trajectory command with horizon l_p for $x_{t-\delta_{eu}}$
$f_a(\cdot)$	Function of autonomous driving agent	$\boldsymbol{r}_{t-\delta_{eu}}^{\text{ctrl}}$	Control feature corresponding to $x_{t-\delta_{eu}}$
P_{target}	Average power constraint of $\boldsymbol{z}$	$\mathcal{R}_{\text{ctrl}}$	Space of control features
l_x	Dimension of $\boldsymbol{x}$	$\boldsymbol{h}_{t-\delta_{eu}}^{\text{ctrl}}$	Control hidden state corresponding to $x_{t-\delta_{eu}}$
l_z	Dimension of $\boldsymbol{z}$	$\boldsymbol{c}_{t-\delta_{eu}+l_p}^{\text{ctrl}}$	Predicted control command with horizon l_p for $x_{t-\delta_{eu}}$
$g(\cdot,\cdot)$	Distortion measuring function	$\boldsymbol{c}_{t-\delta_{eu}+l_p}^{\text{comb}}$	Predicted combined command with horizon l_p for $x_{t-\delta_{eu}}$
ζ_{ratio}	Constraint of bandwidth compression ratio	$\mathcal{C}_{\text{cmd}}$	Space of commands
ζ_{rate}	Constraint of rate	$f_{\text{feat-t}}(\cdot,\cdot,\cdot)$	Function of trajectory feature extractor
ϕ	Parameters of JSCC encoder	$f_{\text{traj}}(\cdot)$	Function of trajactory branch
ψ	Parameters of autonomous driving agent	$f_{\text{feat-c}}(\cdot,\cdot,\cdot)$	Function of control feature extractor
β	Lagrange multiplier	$f_{\text{ctrl}}(\cdot,\cdot)$	Function of control branch
$f_{\boldsymbol{\mu}}(\cdot)$	Function for estimating the mean of reconstructed JSCC symbols	$f_{\text{comb}}(\cdot,\cdot)$	Function of command combination
$f_{\boldsymbol{\sigma}}(\cdot)$	Function for estimating the standard deviation of reconstructed JSCC symbols	$\lambda_{[\cdot]}$	Hyperparameters of agent
K_a	Size of dataset	i	General index depended on the context

struggle with scalability and adaptation in non-stationary environments, a critical limitation in high-mobility contexts such as vehicular networks or telesurgical systems. In contrast, data-driven approaches, particularly deep learning, offer a promising solution by approximating complex policies and predicting future system states based on historical and contextual data [142].

However, generic deep learning models require large amounts of training data and often suffer from poor generalization when deployed in environments with different statistical properties than those encountered during training. To address this, [44] proposes integrating domain knowledge, such as information-theoretic bounds, queueing models, and cross-layer dependencies, into the learning process. This hybrid model- and data-driven approach improves learning efficiency, convergence rate, and interpretability, enabling URLLC systems to make accurate predictions under strict Quality of Service (QoS) constraints.

5.2.1 Analytical Foundations for Predictive URLLC

The analytical foundations of predictive URLLC aim to characterize the performance limits and behaviors of communication systems under stringent latency and reliability constraints. These foundations provide a critical backbone for prediction mechanisms, offering tractable models to anticipate performance metrics and guide learning algorithms in low-latency environments. Central to this analytical core are tools from short blocklength information theory, queuing theory, and stochastic geometry, each capturing essential dynamics in URLLC systems.

Short Blocklength Information Theory

Short blocklength information theory revises classical Shannon capacity, which assumes infinite coding blocklength and vanishing error probability, to better reflect the realities of URLLC where packets are short and must be delivered with finite delay. [60] introduced an approximation for the maximum coding rate achievable over AWGN channels given a fixed blocklength and error probability, highlighting a trade-off between rate, reliability, and latency. This finite blocklength regime is essential for predicting achievable throughput and required bandwidth in URLLC applications, especially when channel conditions vary rapidly. Moreover, these models allow systems to proactively allocate resources based on predicted reliability performance under different coding and modulation schemes.

Queuing Theory

Queuing theory contributes to predictive URLLC through models that estimate queuing delay and buffer occupancy probabilities. Although average delay models such as Little's Law offer baseline estimates, URLLC requires a focus on statistical delay bounds, particularly the violation probability of a given latency threshold. Tools such as effective bandwidth and effective capacity translate arrival and service processes into exponential tail bounds on queuing delay distributions, offering a predictive mechanism for resource provisioning and

traffic shaping [143]. Additionally, the AoI metric captures the freshness of received data and is critical in predictive control systems, where outdated information can undermine decision accuracy in real-time operations [144].

Stochastic Geometry

Stochastic geometry extends predictive analytics to large-scale networks with random spatial distributions of users and base stations. This framework models wireless networks as spatial point processes, allowing statistical predictions of link availability, interference levels, and access delays under varying user densities. Although most stochastic geometry analyses focus on average behaviors, recent extensions consider delay distributions and coverage probabilities tailored for URLLC [145]. For example, modeling the probability of delay outage under different network topologies enables predictive scheduling and handover strategies that prevent latency violations.

Cross-layer optimization serves as a unifying framework, integrating insights from these theories to capture dependencies across physical, link, and network layers. For predictive purposes, cross-layer models enable the estimation of end-to-end performance metrics based on contextual parameters such as channel state, queueing status, and mobility patterns. However, these models are often analytically intractable because of nonconvexity and high dimensionality. Thus, recent work focuses on using these analytical tools to inform the design and initialization of learning-based predictors, guide them toward feasible policy spaces, and reduce training time and error rates [44].

5.2.2 Deep Learning for Predictive Modeling

Deep learning has emerged as a promising paradigm for predictive modeling in URLLC systems, offering flexible, data-driven methods to approximate complex mappings between high-dimensional input states and decision outputs. Unlike traditional optimization algorithms, which often require explicit models and are computationally intensive for real-time use, deep learning methods can infer near-optimal solutions with low latency once trained, making them suitable for the sub-millisecond response times demanded by URLLC applications [44]. Predictive modeling using deep learning enables proactive resource allocation, early detection of latency violations, mobility prediction, and anticipatory handover, thereby improving both reliability and responsiveness in dynamic wireless environments.

Three primary categories of deep learning techniques are leveraged for predictive tasks in URLLC: supervised learning, unsupervised learning, and Deep Reinforcement Learning (DRL).

Supervised Learning

Supervised learning is particularly effective when labeled datasets are available, such as historical traces of user mobility or network traffic. Models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), including their gated variants like

LSTM and GRU, are used to predict time series behavior, such as future channel conditions, delay violations, or user trajectory. These models enable anticipatory mechanisms, for example, scheduling resources based on predicted user movement or traffic spikes, reducing handover failures and packet collisions.

Unsupervised Learning

Unsupervised learning, on the contrary, is used when labeled data are unavailable or impractical to collect. Techniques such as autoencoders and GAN are used to learn compact representations or generate synthetic training data for scenarios with limited observations [146]. For example, [147] proposed the use of unsupervised deep learning to optimize resource allocation and network scheduling by learning the structure of the optimization landscape directly from observed data. This method enables efficient function approximation in systems with intractable models or unknown distributions.

Deep Reinforcement Learning

Deep reinforcement learning (DRL) combines the predictive power of deep learning with decision-making under uncertainty, allowing URLLC systems to learn optimal policies through interaction with the environment. DRL algorithms such as Deep Q-Networks (DQN), Actor-Critic methods, and Proximal Policy Optimization (PPO) are used to explore and exploit predictive policies for resource scheduling, access control, and task offloading in real-time [148]. A key advantage of DRL is its model-free nature, which allows it to learn from observed feedback without requiring an explicit system model. However, exploration safety is a critical concern, especially in URLLC, where poor actions can lead to QoS violations or system instability. This has motivated research on safe DRL, where the search for policies is restricted to ensure compliance with the reliability and latency limits during both the learning and inference phases [149].

Despite their potential, deep learning-based prediction systems in URLLC face several challenges, including non-stationarity of wireless environments, the scarcity of labeled training data, and the need for fast convergence. To address these, hybrid approaches that integrate domain knowledge into the learning process are rising, such as initializing models with output from theoretical frameworks or designing custom loss functions that reflect delay/reliability trade-offs. These hybrid models improve generalizability, reduce training time, and ensure compliance with URLLC constraints, making deep learning a viable component of predictive intelligence in 6G networks.

5.2.3 Recent Advancement

In the context of URLLC, prediction-based methods have been explored to mitigate delays. For example, [150] proposed a technique to predict movements or force feedback to reduce perceived delay in tactile Internet applications. Similarly, [151] presented a co-design approach for packetized predictive control (PPC) in real-time CPS, addressing the delay in the

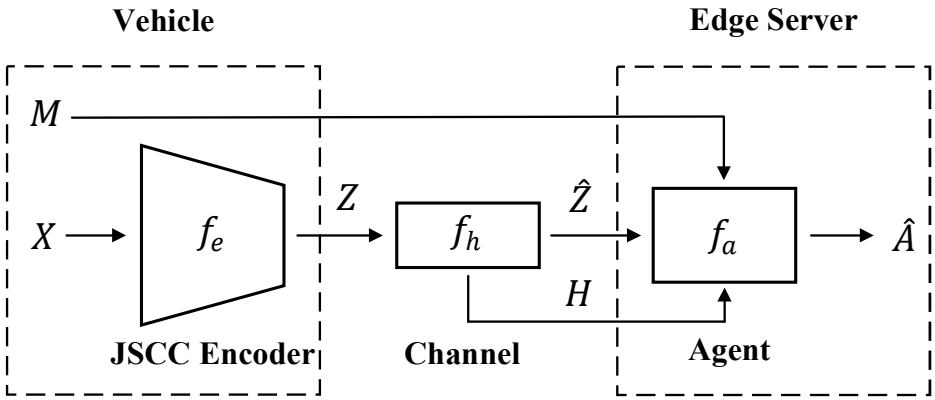


Figure 5.1: General framework of edge-enabled autonomous driving.

tight interaction between wireless communication and control systems. For visual content, [152] investigated how predictive displays can mitigate communication delays in telesurgery using AR technology. The proposed system provided real-time visual feedback to surgeons by predicting movements of robotic tools, significantly improving task completion times under latency without increasing error rates. Likewise, [153] introduced edge intelligence to predict user motion, enabling pre-rendering and caching of VR content, thus significantly reducing the latency in VR streaming. However, in these studies [150]–[153], the trade-off between the prediction horizon and the reliability of the system was not adequately addressed. To bridge this gap, [136] focused on the challenges of delay and reliability in URLLC by co-designing prediction and communication systems. The proposed framework enables mobile devices to predict future states and send these predictions to a data center in advance, thus reducing perceived delays. The study also analyzed the trade-off between prediction accuracy and system reliability, demonstrating that longer prediction horizons increase the likelihood of errors.

5.3 System Model and Problem Formulation

As shown in Fig. 5.1, we consider an edge server that provides computing service for a single vehicle (device). The vehicle transmits JSCC symbols and encoded state information to the edge server. After processing the received data, the edge server sends the drive commands back to the vehicle.

The on-vehicle JSCC encoder f_e is defined as:

$$f_e : \mathcal{X} \to \mathcal{Z} : \boldsymbol{x} \mapsto \boldsymbol{z}, \tag{5.1}$$

where $\boldsymbol{x} \in \mathbb{R}^{l_x}$ denotes the input image, and $\boldsymbol{z} = [z_1, \ldots, z_{l_z}] \in \mathbb{C}^{l_z}$ denotes the transmitted JSCC symbols. Here, l_x denotes the source bandwidth, which is the product of the height, width, and number of color channels of the image $\boldsymbol{x}$. The parameter l_z denotes the channel bandwidth. We define l_z/l_x as the *bandwidth compression ratio* [14]. In particular, the

transmitted JSCC symbols should satisfy the average power constraint P_{target}:

$$\frac{1}{l_z}\sum_{i=1}^{l_z}|z_i|^2 \leq P_{\text{target}}. \tag{5.2}$$

Then $\boldsymbol{z}$ are transmitted to the edge server via communication channels, which can be mathematically represented by the function:

$$f_h : \mathcal{Z} \to \mathcal{Z} : \boldsymbol{z} \mapsto \hat{\boldsymbol{z}}. \tag{5.3}$$

In this chapter, we model the communication channel as a frequency-selective channel implemented through Orthogonal Frequency-Division Multiplexing (OFDM) to mitigate multipath fading, as detailed in [85]:

$$\check{\boldsymbol{z}} = f_h(\boldsymbol{z}) = \boldsymbol{h}\cdot\boldsymbol{z} + \boldsymbol{n}, \tag{5.4}$$

where $\check{\boldsymbol{z}}$ denotes the received JSCC symbols, and $\boldsymbol{n} \sim \mathcal{CN}(0, \boldsymbol{\sigma}_n^2 I)$ represents complex Gaussian noise with zero mean and standard deviation $\boldsymbol{\sigma}_n$, where I denotes the identity matrix and $\boldsymbol{\sigma}_n$ is a diagonal matrix. The channel frequency response $\boldsymbol{h} \in \mathbb{C}^{l_z}$ captures the characteristics of multipath fading. A comprehensive modeling of the OFDM channel is provided in Appendix A.

In this chapter, we assume that the perfect CSI is available at the receiver, while the transmitter has no knowledge of it. After receiving, the JSCC symbols are equalized:

$$\hat{\boldsymbol{z}} = \frac{h^*}{|h|^2}\check{\boldsymbol{z}}, \tag{5.5}$$

where h^* denotes the conjugate of channel coefficient h and $\hat{\boldsymbol{z}}$ denotes the reconstructed JSCC symbols.

Following transmission, the reconstructed JSCC symbols $\hat{\boldsymbol{z}}$ are processed by the autonomous driving agent f_a with state information and channel state, which is defined as:

$$f_a : \mathcal{Z}\times\mathcal{M}\times\mathcal{H} \to \mathcal{A} : (\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h}) \mapsto \hat{\boldsymbol{a}}, \tag{5.6}$$

where $\hat{\boldsymbol{a}} \sim p(\hat{\boldsymbol{a}})$ denotes the estimated action, which approximates the ground-truth action $\boldsymbol{a} \sim p(\boldsymbol{a})$. In addition, $\boldsymbol{m}$ denotes state information consisting of vehicle speed, discrete navigation command, destination coordinates, and timestamp. The agent incorporates vehicle state information $\boldsymbol{m}$ and channel state information $\boldsymbol{h}$ as conditional inputs, establishing a direct link between communication and control to improve decision-making. Since the state information $\boldsymbol{m}$ typically consumes negligible bandwidth (in this chapter, $\boldsymbol{m}$ consists of four floating-point numbers and one integer), we assume that it is received losslessly by the edge server, provided that the corresponding image is successfully received and decoded.

The task-oriented objective of edge-enabled autonomous driving is to minimize the expected distortion between the ground-truth action $\boldsymbol{a}$ and the estimated action $\hat{\boldsymbol{a}}$, which

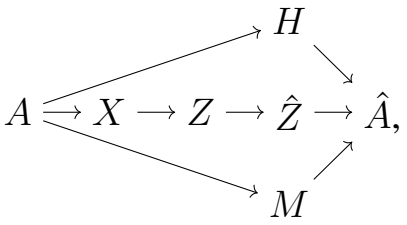


Figure 5.2: The DPGM for edge-enabled autonomous driving.

is defined as $g : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$. Consequently, the problem of the proposed task-oriented co-design is defined in Problem 2.

Problem 2:

$$\min_{f_e, f_a} \mathbb{E}_{\boldsymbol{a} \sim p(\boldsymbol{a}|\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})} \left\{ \mathbb{E}_{\hat{\boldsymbol{a}} \sim p(\hat{\boldsymbol{a}}|\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})} \left[g(\boldsymbol{a}, \hat{\boldsymbol{a}}) \right] \right\} \tag{5.7}$$

$$\text{s.t.} \quad l_z / l_x - \zeta_{\text{ratio}} \leq 0, \tag{5.8}$$

$$(5.1), (5.2), (5.3), (5.4), (5.6), \tag{5.9}$$

where ζ_{ratio} denotes an upper bound of the bandwidth compression ratio.

Problem 2 is an abstract formulation that describes the overarching goal of optimizing task performance through appropriate parameter selection. Note that the dynamic model of the vehicle is not explicitly included because it is implicitly captured by the autonomous driving agent f_a. Specifically, the dynamic model of the vehicle is learned by the agent during the E2E training process, in which the agent is optimized to predict effective control actions directly from observed data and feedback. Thus, while vehicle dynamics are not explicitly modeled or parameterized, the learned representation within the agent ensures that the necessary knowledge of the system dynamics is implicitly embedded within the model parameters.

However, directly solving Problem 2 poses significant computational challenges, particularly in the evaluation of the expectation over random variables, which involves integration that can be computationally prohibitive. In addition, finding a proper objective function $g(\cdot, \cdot)$ is also difficult. In the following section, we introduce the VIB approach combined with DNNs to effectively address Problem 2.

5.4 Variational Information Bottleneck Approach

Based on the discussion in Section 5.3, the DPGM of the proposed framework can be depicted as shown in Fig. 5.2, where A, X, Z, $\hat{Z}$, M, H and $\hat{A}$ denotes the random variables of the ground-truth action $\boldsymbol{a}$, input images $\boldsymbol{x}$, transmitted JSCC symbols $\boldsymbol{z}$, reconstructed JSCC symbols $\hat{\boldsymbol{z}}$, state information $\boldsymbol{m}$, channel state $\boldsymbol{h}$, and estimated action $\hat{\boldsymbol{a}}$, respectively. With

this DPGM, the IB can be formulated as an optimization problem:

$$\begin{aligned} \min_{\phi,\psi} \quad & -I(A;\hat{Z},M,H) \\ \text{s.t.} \quad & I(X;\hat{Z})-\zeta_{\text{rate}} \leq 0, \end{aligned} \tag{5.10}$$

where ϕ and ψ are the parameters of JSCC encoder f_e and autonomous driving agent f_a, respectively. ζ_{rate} denotes the upper bound of the rate. Eq. (5.10) is derived as a practical and task-oriented reformulation of Problem 2. The objective function in Eq. (5.10), $I(A;\hat{Z},M,H)$, measures the information shared between the target variable A and the combined set of variables $(\hat{Z},M,H)$. By minimizing $-I(A;\hat{Z},M,H)$, we ensure that the information about A retained in $(\hat{Z},M,H)$ is maximized, aligning the optimization with the task-specific objectives of Problem 2. This is supported by the DPGM structure where $\hat{A}$ depends on $(\hat{Z},M,H)$, highlighting that preserving information about A in these variables is key to achieving optimal task performance. Moreover, the constraints in Eq. (5.10) explicitly incorporate communication limitations through the term $I(X,\hat{Z})$, which limits the rate of transmitted information. These constraints parallel the bandwidth restrictions in Problem 2, thereby ensuring consistency between the two formulations.

By introducing the Lagrange multiplier β, IB can be further formulated to minimize the following objective function:

$$\mathcal{L}_{\text{IB}} := \underbrace{-I(A;\hat{Z},M,H)}_{\text{Distortion}} + \beta \underbrace{I(X;\hat{Z})}_{\text{Rate}}. \tag{5.11}$$

Building on the foundational works of [11] and [10], we develop a VIB approach to approximate each term in Eq. (5.11), addressing the intractability of mutual information. The first term $-I(A;\hat{Z},M,H)$ can be expressed as:

$$-I(A;\hat{Z},M,H) = -\int p(\boldsymbol{a},\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h}) \log p(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h}) \,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\hat{\boldsymbol{z}}\,\mathrm{d}\boldsymbol{m}\,\mathrm{d}\boldsymbol{h} - H(A), \tag{5.12}$$

where $H(A)$ denotes the entropy of random variable A, which is independent of the optimization and thus can be ignored. In addition, $p(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})$ is the posterior probability, which can be derived from the DPGM [10], [11] as:

$$p(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h}) = \int \frac{p(\boldsymbol{a})p(\boldsymbol{x}|\boldsymbol{a})p_\phi(\boldsymbol{z}|\boldsymbol{x})p(\hat{\boldsymbol{z}}|\boldsymbol{z})p(\boldsymbol{m}|\boldsymbol{a})p(\boldsymbol{h}|\boldsymbol{a})}{p(\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}. \tag{5.13}$$

Since this integration is intractable in our case, we use $q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})$ as a variational approximation of $p(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})$. Based on the fact that KL divergence is always non-negative, the following inequality can be obtained:

$$-I(A;\hat{Z},M,H) + H(A) \leq \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\Big[\mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi}\big[\mathbb{E}_{\boldsymbol{m},\boldsymbol{h}}[-\log q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})]\big]\Big]. \tag{5.14}$$

The second term $I(X;\hat{Z})$ can be formulated as:

$$I(X;\hat{Z}) = \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[D_{\mathrm{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|p(\hat{\boldsymbol{z}}))\right], \tag{5.15}$$

where $p(\hat{\boldsymbol{z}})$ is the intractable prior probability of $\hat{\boldsymbol{z}}$. Instead of using approximation proposed in [154], we adopt a predefined Gaussian distribution $q(\hat{\boldsymbol{z}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\hat{\boldsymbol{z}}}, \boldsymbol{\sigma}^2_{\hat{\boldsymbol{z}}} I)$ as the approximation of $p(\hat{\boldsymbol{z}})$ [121], where $\boldsymbol{\mu}_{\hat{\boldsymbol{z}}}$ and $\boldsymbol{\sigma}_{\hat{\boldsymbol{z}}}$ represent the mean and standard deviation of the Gaussian distribution, respectively.

In addition, we model the JSCC encoder f_e as a probability model $p_\phi(\boldsymbol{z}|\boldsymbol{x})$. Considering that $p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}) = \int p_\phi(\boldsymbol{z}|\boldsymbol{x}) p(\hat{\boldsymbol{z}}|\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$, where $p(\hat{\boldsymbol{z}}|\boldsymbol{z})$ represents the probabilistic nature of the channel function f_h, and assuming perfect CSI, we define $p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}) \sim \mathcal{N}(f_{\boldsymbol{\mu}}(\hat{\boldsymbol{z}}), f^2_{\boldsymbol{\sigma}}(\hat{\boldsymbol{z}}) I)$. Here, $f_{\boldsymbol{\mu}}(\cdot)$ and $f_{\boldsymbol{\sigma}}(\cdot)$ are functions that estimate the mean and standard deviation of this Gaussian distribution, respectively.

Meanwhile, the mutual information $I(X;\hat{Z})$ can also be written as:

$$\begin{aligned} I(X;\hat{Z}) &= \int p(\boldsymbol{x},\hat{\boldsymbol{z}}) \log \frac{p(\boldsymbol{x},\hat{\boldsymbol{z}})}{p(\boldsymbol{x})p(\hat{\boldsymbol{z}})}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\hat{\boldsymbol{z}} \\ &= \int p(\boldsymbol{x},\hat{\boldsymbol{z}}) \log \frac{p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})}{p(\hat{\boldsymbol{z}})}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\hat{\boldsymbol{z}}. \end{aligned}$$

Since $D_{\mathrm{KL}}(p(\hat{\boldsymbol{z}}) \parallel q(\hat{\boldsymbol{z}})) \geq 0$, we have

$$\int p(\hat{\boldsymbol{z}}) \log p(\hat{\boldsymbol{z}})\,\mathrm{d}\hat{\boldsymbol{z}} \geq \int p(\hat{\boldsymbol{z}}) \log q(\hat{\boldsymbol{z}})\,\mathrm{d}\hat{\boldsymbol{z}}.$$

So that

$$\begin{aligned} I(X;\hat{Z}) &\leq \int p(\boldsymbol{x},\hat{\boldsymbol{z}}) \log \frac{p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})}{q(\hat{\boldsymbol{z}})}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\hat{\boldsymbol{z}} \\ &= \int p(\boldsymbol{a},\boldsymbol{x}) p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}) \log \frac{p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})}{q(\hat{\boldsymbol{z}})}\,\mathrm{d}\boldsymbol{a}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\hat{\boldsymbol{z}} \\ &= \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left[D_{\mathrm{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|q(\hat{\boldsymbol{z}}))\right]. \end{aligned} \tag{5.16}$$

Therefore, we derive the following corollary as an approximation of Eq. (5.11).

Corollary 1: Assume the DPGM shown in Fig. 5.2, let $q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})$ be a variational approximation of $p(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})$, let $q(\hat{\boldsymbol{z}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\hat{\boldsymbol{z}}}, \boldsymbol{\sigma}^2_{\hat{\boldsymbol{z}}} I)$ be a variational approximation of $p(\hat{\boldsymbol{z}})$, and let $p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}) \sim \mathcal{N}(f_{\boldsymbol{\mu}}(\hat{\boldsymbol{z}}), f^2_{\boldsymbol{\sigma}}(\hat{\boldsymbol{z}}) I)$ be a variational approximation of $p(\hat{\boldsymbol{z}}|\boldsymbol{x})$, the upper bound of Eq. (5.11) is given by

$$\begin{aligned} \mathcal{L}_{\mathrm{VIB}} :=\ & \mathbb{E}_{\boldsymbol{a},\boldsymbol{x}}\left\{\mathbb{E}_{\hat{\boldsymbol{z}}|\boldsymbol{x};\phi}\left[\mathbb{E}_{\boldsymbol{m},\boldsymbol{h}}\left[-\log q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}},\boldsymbol{m},\boldsymbol{h})\right]\right] + \beta D_{\mathrm{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x})\|q(\hat{\boldsymbol{z}}))\right\} \\ \geq\ & \mathcal{L}_{\mathrm{IB}} + H(A). \end{aligned} \tag{5.17}$$

This corollary can be optimized using stochastic gradient descent through Monte Carlo sampling, providing a practical framework for empirical estimation and subsequent optimization. Given a dataset with size K_a, a mini-batch $\{(\boldsymbol{a}_i,\boldsymbol{x}_i)\}_{i=1}^{K_b}$ of size K_b is randomly drawn

without overlap in the same epoch to compute the gradient of loss $\mathcal{L}_{\text{VIB}}$. In particular, the number of samples of $-\log q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})$ can be set to 1 as long as the size of the dataset K_a is large enough [121]. Thus, we have the following estimation:

$$\mathcal{L}_{\text{VIB}} \approx \frac{1}{K_b} \sum_{i=1}^{K_b} \Big\{ -\log q_\psi(\boldsymbol{a}_i|\hat{\boldsymbol{z}}_i, \boldsymbol{m}_i, \boldsymbol{h}_i) + \beta D_{\text{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}_i) \| q(\hat{\boldsymbol{z}})) \Big\}. \tag{5.18}$$

Note that the dataset $\{(\boldsymbol{a}_i, \boldsymbol{x}_i)\}_{i=1}^{K_a}$ can be collected from expert agents or human drivers.

5.5 Delay-Aware Trajectory-Guided Control Prediction for Autonomous Driving

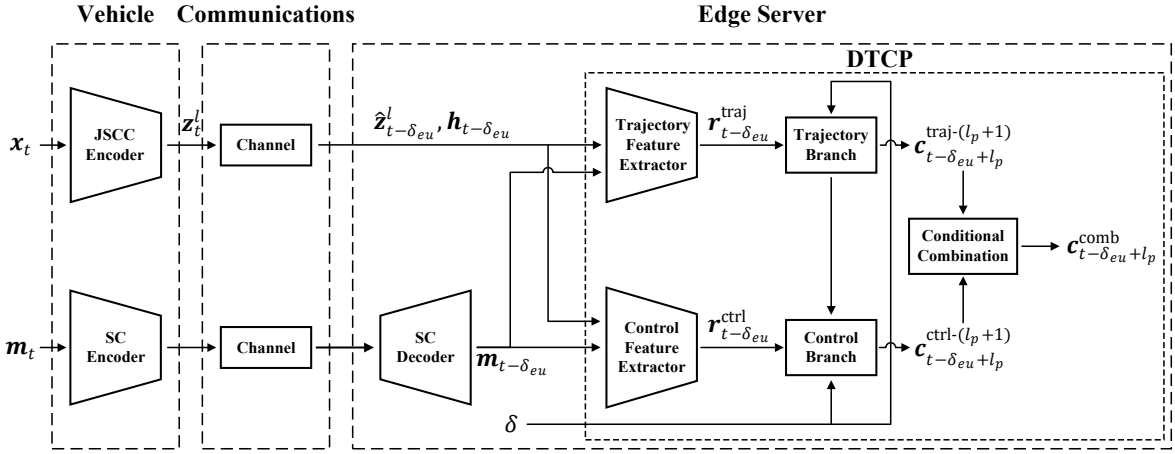


Figure 5.3: The proposed task-oriented co-design framework based on JSCC and DTCP.

TGCP is one of the state-of-the-art frameworks of E2E autonomous driving, integrating trajectory planning and multistage control prediction together [125]. This advanced framework, which uses only a monocular camera, currently ranks third on the Car Learning to Act (CARLA) leaderboard[1]. However, the original TGCP framework relies on a stream of raw images for decision making, resulting in high bandwidth usage. In addition, it does not account for the impact of communication latency on decision-making processes.

To overcome these limitations, we have developed a Delay-aware Trajectory-guided Control Prediction (DTCP) strategy that integrates the trajectory and control branches while considering the delay, as shown in Fig. 5.3. This integration ensures that predicted drive commands reduce the perceived E2E delay, leading to safer and more efficient autonomous driving.

We assume the system is time-slotted and initiates at the time $t = 0$. The duration of each time slot is denoted as τ. Let $\boldsymbol{z}_t^{l(t)}$ denote the transmitted JSCC symbols with length $l(t)$, and $\boldsymbol{m}_t$ represents the state information, both corresponding to the image $\boldsymbol{x}_t$ captured

[1] `https://leaderboard.carla.org/leaderboard/`

by the onboard camera at time t. The term $1 \leq l(t) \leq l_z$ denotes a function that decides the number of selected JSCC symbols of $\boldsymbol{z}_t$. For simplicity of notation, we denote $l(t)$ by l in the following formulation.

Define the index set $D_t \subset \{1, 2, \ldots, l_z\}$ such that:

$$D_t = \{i \in \{1, \ldots, l_z\} \mid \\ |z_i^t|^2 \text{ is one of the } l \text{ largest numbers in } |\boldsymbol{z}_t|^2\}. \tag{5.19}$$

In particular, $\boldsymbol{z}_t^l = \{z_i^t \mid i \in D_t\}$ are JSCC symbols selected from $\boldsymbol{z}_t = [z_1^t, \ldots, z_{l_z}^t]$ based on energy significance. The selected JSCC symbols are kept for transmission, while the missing JSCC symbols are filled with 0 on the edge server. Note that this selection process can be integrated into the JSCC encoder f_e. If only selected JSCC symbols are transmitted, the receiver must be made aware of the indices of the selected or abandoned JSCC symbols, which may increase the communication load. In this chapter, we design the selective JSCC symbols to provide flexibility within this task-oriented communication co-designed paradigm and demonstrate their potential for dimensionality reduction. To address the additional communication load introduced by index transmission, methods such as Variable-Length Variational Feature Encoding (VL-VFE) [11] offer promising directions for further exploration.

5.5.1 Prediction for End-to-End Delay

In edge-enabled autonomous systems, drive commands are often outdated due to E2E delay, including communication, computation, and control delays. Fig. 5.4 shows a complete cycle of the communication, computing, and control process, along with the prediction structure. Assume that an image $\boldsymbol{x}_{t_0}$ is captured by the camera at time $t = t_0$. After encoding and selecting, the JSCC symbols $\boldsymbol{z}_{t_0}^l$ are generated with a computation delay δ_e. The JSCC symbols reconstructed by the edge server are denoted as $\hat{\boldsymbol{z}}_{t_0}^l$ arriving with an uplink delay δ_u. The agent on the edge server takes δ_a time slots to generate the command $\boldsymbol{c}_{t_0+l_p}^{\text{comb}}$, where $l_p \geq 0$ denotes the prediction horizon. The command is then sent back to the vehicle with a downlink delay δ_d. Upon receiving the command, the vehicle takes δ_c time slots to execute the command. Thus, the E2E delay is expressed as $\delta = \delta_e + \delta_u + \delta_a + \delta_d + \delta_c$. Consequently, the perceived E2E delay is given by $\delta - l_p$. Since the command $\boldsymbol{c}_{t_0+l_p}^{\text{comb}}$ consumes negligible bandwidth, it is assumed to be transmitted losslessly to the vehicle in this chapter. It is worth noting that while the uplink delay can be further decomposed into components such as transmission delay, queuing delay, propagation delay, and processing delay, a detailed analysis of each individual component lies outside the primary scope of this chapter. Instead, our focus is on the E2E delay and addressing it through predictive mechanisms.

It is crucial to recognize that in the process described above, when the onboard camera captures an image $\boldsymbol{x}_t$ at any time t, reconstructed JSCC symbols $\hat{\boldsymbol{z}}_{t-\delta_{eu}}^l$ (corresponding to image $\boldsymbol{x}_{t-\delta_{eu}}$) on the edge server are outdated of $\delta_{eu} = \delta_e + \delta_u$ time slots. The combined delay δ_{eu} can be calculated from the timestamp in the state information $\boldsymbol{m}_{t-\delta_{eu}}$, if it is transmitted in sync with $\hat{\boldsymbol{z}}_{t-\delta_{eu}}^l$. In addition, we assume that the agent's computation delay δ_a and the

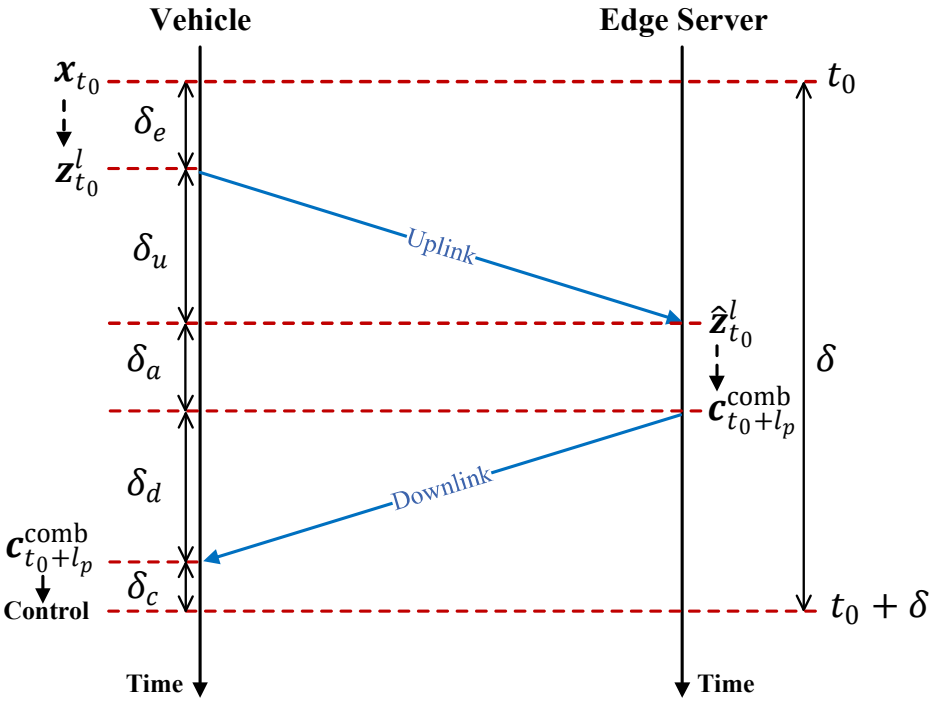


Figure 5.4: The illustration of a completed cycle of the communication, computing, and control process, along with the prediction structure.

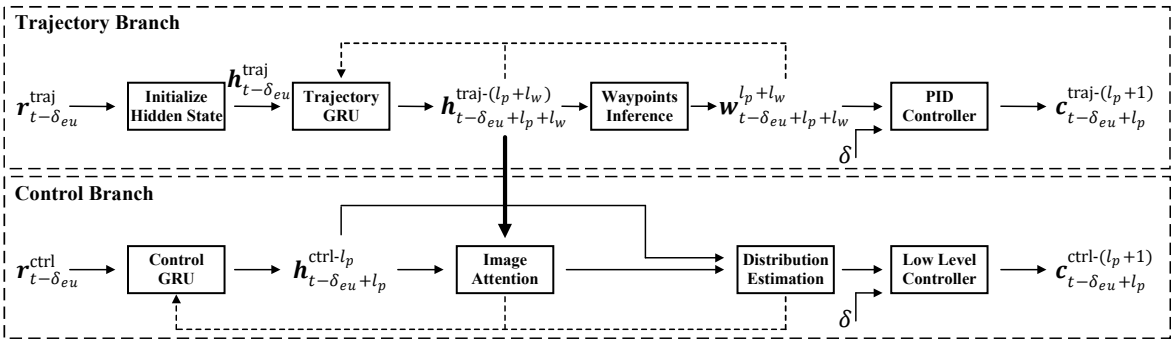


Figure 5.5: The framework of the trajectory and control branch of DTCP.

control execution delay δ_c are known constants, and the edge server continuously measures the downlink communication delay in real time. With this information, the agent remains aware of the E2E delay δ, allowing it to dynamically adjust the prediction horizon.

Note that the amount of transmitted information mainly influences the uplink delay δ_u and downlink delay δ_d. In this work, the dimensionality l of the transmitted JSCC symbols is predefined. As a result, delays caused by transmission can be treated as known and constant. Although we do not explicitly optimize the communication strategy for these delays during training, they are inherently controlled through the predefined dimension of the JSCC symbols. This design choice simplifies the modeling of E2E delay and ensures predictable communication latency.

5.5.2 Trajectory Branch

The trajectory branch first generates planned waypoints, and then a low-level PID controller generates trajectory commands based on them. We define the function of the trajectory feature extractor as:

$$f_{\text{feat-t}} : \mathcal{Z} \times \mathcal{M} \times \mathcal{H} \to \mathcal{R}_{\text{traj}} : (\hat{\boldsymbol{z}}^{l}_{t-\delta_{eu}}, \boldsymbol{m}_{t-\delta_{eu}}, \boldsymbol{h}_{t-\delta_{eu}}) \mapsto \boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}}, \tag{5.20}$$

where $\hat{\boldsymbol{z}}^{l}_{t-\delta_{eu}}, \boldsymbol{m}_{t-\delta_{eu}}$, and $\boldsymbol{h}_{t-\delta_{eu}}$ represent the reconstructed JSCC symbols, state information, and channel state, respectively, corresponding to image $\boldsymbol{x}_{t-\delta_{eu}}$ captured by the camera δ_{eu} time slots ago. At time t, the trajectory feature on the edge server is denoted as $\boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}}$, as shown in Fig. 5.5. The trajectory hidden state $\boldsymbol{h}^{\text{traj}}_{t-\delta_{eu}}$ of a Gated Recurrent Unit (GRU) [155] is initialized with the trajectory feature. Then it auto-regressively generates the sequence of trajectory hidden states $\boldsymbol{h}^{\text{traj-}(l_p+l_w)}_{t-\delta_{eu}+l_p+l_w} = (\boldsymbol{h}^{\text{traj}}_{t-\delta_{eu}+l_p+l_w}, \ldots, \boldsymbol{h}^{\text{traj}}_{t-\delta_{eu}+1})$, where l_w denotes an extra prediction horizon for planned waypoints and $l_p + l_w$ in the superscript represents the length of the sequence. Using a waypoint inference network, the planned $l_p + l_w$ waypoints can be obtained from the sequence of trajectory hidden states, denoted as $\boldsymbol{w}^{l_p+l_w}_{t-\delta_{eu}+l_p+l_w} = (\boldsymbol{w}_{t-\delta_{eu}+l_p+l_w}, \ldots, \boldsymbol{w}_{t-\delta_{eu}+1})$. The initial waypoint $\boldsymbol{w}_{t-\delta_{eu}}$ is defined as the origin.

Each trajectory $\boldsymbol{w}^{l_w+1}_{t-i+l_w} = (\boldsymbol{w}_{t-i+l_w}, \ldots, \boldsymbol{w}_{t-i})$ with length $l_w + 1$ is processed by a PID controller to generate the predicted trajectory command $\boldsymbol{c}^{\text{traj}}_{t-i}$, where $i = \delta_{eu}, \ldots, \delta_{eu} - l_p$. Thus, the sequence of predicted trajectory commands branch is denoted as $\boldsymbol{c}^{\text{traj-}(l_p+1)}_{t-\delta_{eu}+l_p} = (\boldsymbol{c}^{\text{traj}}_{t-\delta_{eu}+l_p}, \ldots, \boldsymbol{c}^{\text{traj}}_{t-\delta_{eu}})$. We defined the function of the trajectory branch as:

$$f_{\text{traj}} : \mathcal{R}_{\text{traj}} \to \mathcal{C}^{l_p+1}_{\text{cmd}} : \boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}} \mapsto \boldsymbol{c}^{\text{traj-}(l_p+1)}_{t-\delta_{eu}+l_p}. \tag{5.21}$$

5.5.3 Control Branch

As outlined in [125], a control model that predicts current actions based solely on current inputs typically employs supervised training similar to behavior cloning, which assumes that the data is i.i.d. However, for autonomous driving, future states and commands are under the influence of historical commands. To address this problem and deal with latency, the control branch predicts control commands in multiple steps in the future and obtains the desired commands based on the E2E delay δ.

We defined the mapping of the reconstructed JSCC symbols to the control features as:

$$f_{\text{feat-c}} : \mathcal{Z} \times \mathcal{M} \times \mathcal{H} \to \mathcal{R}_{\text{ctrl}} : (\hat{\boldsymbol{z}}^{l}_{t-\delta_{eu}}, \boldsymbol{m}_{t-\delta_{eu}}, \boldsymbol{h}_{t-\delta_{eu}}) \mapsto \boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}}. \tag{5.22}$$

At time t, the control hidden state $\boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}}$ is initialized with zero value and enters control GRU with the control feature $\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}}$ to generate the next hidden state $\boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}+1}$. The hidden state of the control branch $\boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}+1}$ and the hidden state of the trajectory branch $\boldsymbol{h}^{\text{traj}}_{t-\delta_{eu}+1}$ are used to estimate the important regions of the image by generating a binary mask that matches the shape of the image feature map from the middle layer of the control feature

extractor [125]. This mask is then applied through element-wise multiplication with the feature map. The results of image attention are then used to generate the predicted control feature $\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}+1}$ and the control command $\boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}+1}$. The next control GRU hidden state $\boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}+2}$ is obtained from the previous hidden state $\boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}+1}$ and the predicted control feature $\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}+1}$. This process auto-regressively generates the sequence of control hidden states $\boldsymbol{h}^{\text{ctrl-}l_p}_{t-\delta_{eu}+l_p} = (\boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}+l_p}, \ldots, \boldsymbol{h}^{\text{ctrl}}_{t-\delta_{eu}+1})$, which is used to generate the sequence of predicted control features $\boldsymbol{r}^{\text{ctrl-}l_p}_{t-\delta_{eu}+l_p} = (\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}+l_p}, \ldots, \boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}+1})$. Based on that, the sequence of predicted control commands $\boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p} = (\boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}+l_p}, \ldots, \boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}})$ is derived from a low-level controller, where $\boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}}$ is directly generated from the initial control feature $\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}}$. The function of the trajectory branch is defined as:

$$f_{\text{ctrl}} : \mathcal{R}_{\text{ctrl}} \times \mathcal{H}^{l_p}_{\text{traj}} \to \mathcal{C}^{l_p+1}_{\text{cmd}} : (\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}}, \boldsymbol{h}^{\text{traj-}l_p}_{t-\delta_{eu}+l_p}) \mapsto \boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p}. \tag{5.23}$$

5.5.4 Two Branch Combination

To minimize the perceived E2E delay δ_r, $l_p \geq \delta$ must be satisfied, i.e., $l_p - \delta_{eu} \geq \delta_a + \delta_d + \delta_c$. Because the trajectory branch and the control branch specialize in different driving scenarios, commands from the two branches are conditionally fused to obtain the combined command $\boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p}$. This fusion depends on the driving situation – whether the vehicle is turning or not. In addition, considering the trade-off between the prediction horizon and the reliability of the system, the predicted control is applied when the delay exceeds a certain threshold δ_T for the turning situation. Otherwise, the robustness of the system can deal with the delay better than applying predicted commands. We define this combination function as:

$$f_{\text{comb}} : \mathcal{C}^{l_p+1}_{\text{cmd}} \times \mathcal{C}^{l_p+1}_{\text{cmd}} \to \mathcal{C}_{\text{cmd}} : (\boldsymbol{c}^{\text{traj-}(l_p+1)}_{t-\delta_{eu}+l_p}, \boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p}) \mapsto \boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p}. \tag{5.24}$$

The combined command $\boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p}$ is denoted as:

$$\boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p} = \begin{cases} \lambda_c \cdot \boldsymbol{c}^{\text{traj}}_{t-\delta_{eu}+l_p} + (1-\lambda_c) \cdot \boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}+l_p}, & \text{if turning and } \delta \geq \delta_T, \\ \lambda_c \cdot \boldsymbol{c}^{\text{traj}}_{t-\delta_{eu}} + (1-\lambda_c) \cdot \boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}}, & \text{if turning and } \delta < \delta_T, \\ \lambda_c \cdot \boldsymbol{c}^{\text{ctrl}}_{t-\delta_{eu}} + (1-\lambda_c) \cdot \boldsymbol{c}^{\text{traj}}_{t-\delta_{eu}}, & \text{otherwise}, \end{cases} \tag{5.25}$$

where $\lambda_c \in [0.5, 1]$ is a hyperparameter. The details of a complete cycle of communication, computing, and control of DTCP and task-oriented JSCC are illustrated in Algorithm 4.

Algorithm 4 Communication, Computing, and Control of DTCP and Task-Oriented JSCC.

Initialization: Load the pre-trained parameter ϕ for JSCC encoder (f_e) and parameter ψ for DTCP ($f_{\text{feat-t}}, f_{\text{feat-c}}, f_{\text{traj}}$, and f_{ctrl}).
Vehicle:
    At time $t-\delta_{eu}$, capture image $\boldsymbol{x}_{t-\delta_{eu}}$ and generate state information $\boldsymbol{m}_{t-\delta_{eu}}$.
    At time $t-\delta_u$, generate selected JSCC symbols: $\boldsymbol{z}^l_{t-\delta_{eu}} \leftarrow f_e(\boldsymbol{x}_{t-\delta_{eu}})$.
Edge Server:
    At time t, receive reconstructed JSCC symbols $\hat{\boldsymbol{z}}^l_{t-\delta_{eu}} \leftarrow f_h(\boldsymbol{z}^l_{t-\delta_{eu}})$ and state information $\boldsymbol{m}_{t-\delta_{eu}}$. Measure corresponding channel state $\boldsymbol{h}_{t-\delta_{eu}}$.
    Generate trajectory feature: $\boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}} \leftarrow f_{\text{feat-t}}(\hat{\boldsymbol{z}}^l_{t-\delta_{eu}}, \boldsymbol{m}_{t-\delta_{eu}}, \boldsymbol{h}_{t-\delta_{eu}})$.
    Generate sequence of trajectory command: $\boldsymbol{c}^{\text{traj-}(l_p+1)}_{t-\delta_{eu}+l_p} \leftarrow f_{\text{traj}}(\boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}})$, and sequence of hidden state $\boldsymbol{h}^{\text{traj-}l_p}_{t-\delta_{eu}+l_p}$.
    Generate control feature: $\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}} \leftarrow f_{\text{feat-c}}(\hat{\boldsymbol{z}}^l_{t-\delta_{eu}}, \boldsymbol{m}_{t-\delta_{eu}}, \boldsymbol{h}_{t-\delta_{eu}})$.
    Generate sequence of control command: $\boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p} \leftarrow f_{\text{ctrl}}(\boldsymbol{r}^{\text{ctrl}}_{t-\delta_{eu}}, \boldsymbol{h}^{\text{traj-}l_p}_{t-\delta_{eu}+l_p})$.
    At time $t+\delta_a$, generate combined command: $\boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p} \leftarrow f_{\text{comb}}(\boldsymbol{c}^{\text{traj-}(l_p+1)}_{t-\delta_{eu}+l_p}, \boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p})$.
Vehicle:
    At time $t+\delta_a+\delta_d$, receive command $\boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p}$.
    At time $t+\delta_a+\delta_d+\delta_c$, vehicle is controlled by the command $\boldsymbol{c}^{\text{comb}}_{t-\delta_{eu}+l_p}$.

5.5.5 Loss function

We denote the estimated action corresponding to image $\boldsymbol{x}_{t-\delta_{eu}}$ by

$$\hat{\boldsymbol{a}}_{t-\delta_{eu}} = (v_{t-\delta_{eu}}, s_{t-\delta_{eu}}, \boldsymbol{w}^{l_p+l_w}_{t-\delta_{eu}+l_p+l_w}, \boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}}, \boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p}, \boldsymbol{r}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p}), \tag{5.26}$$

which consists of task-critical variables, where $v_{t-\delta_{eu}}$ denotes the estimated target velocity and $s_{t-\delta_{eu}}$ denotes the value of the extracted features. The corresponding ground-truth action is defined as:

$$\boldsymbol{a}_{t-\delta_{eu}} = ({}^{\text{ex}}v_{t-\delta_{eu}}, {}^{\text{ex}}s_{t-\delta_{eu}}, {}^{\text{ex}}\boldsymbol{w}^{l_p+l_w}_{t-\delta_{eu}+l_p+l_w}, {}^{\text{ex}}\boldsymbol{r}^{\text{traj}}_{t-\delta_{eu}}, {}^{\text{ex}}\boldsymbol{c}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p}, {}^{\text{ex}}\boldsymbol{r}^{\text{ctrl-}(l_p+1)}_{t-\delta_{eu}+l_p}), \tag{5.27}$$

which is collected from expert agents or human drivers.

The loss function of the trajectory branch is defined as follows:

$$\mathcal{L}_{\text{traj}} = \|\boldsymbol{w}_{t-\delta_{eu}+l_p+l_w}^{l_p+l_w} - {}^{\text{ex}}\boldsymbol{w}_{t-\delta_{eu}+l_p+l_w}^{l_p+l_w}\|_1 + \lambda_{\text{feat}}\|\boldsymbol{r}_{t-\delta_{eu}}^{\text{traj}} - {}^{\text{ex}}\boldsymbol{r}_{t-\delta_{eu}}^{\text{traj}}\|_2, \tag{5.28}$$

where λ_{feat} is a hyperparameter, $\|\cdot\|_1$ denotes the l_1-norm, $\|\cdot\|_2$ denotes the Euclidean distance (l_2-norm).

For the control branch, the distribution of the control action is modeled as a beta distribution [125]. The loss function of the control branch is defined as follows:

$$\mathcal{L}_{\text{ctrl}} = \frac{1}{l_p+1}\sum_{i=t-\delta_{eu}}^{t-\delta_{eu}+l_p} D_{\text{KL}}(\mathcal{B}e(\boldsymbol{c}_i^{\text{ctrl}})\|\mathcal{B}e({}^{\text{ex}}\boldsymbol{c}_i^{\text{ctrl}})) + \lambda_{\text{feat}}\|\boldsymbol{r}_{t-\delta_{eu}+l_p}^{\text{ctrl-}(l_p+1)} - {}^{\text{ex}}\boldsymbol{r}_{t-\delta_{eu}+l_p}^{\text{ctrl-}(l_p+1)}\|_2, \tag{5.29}$$

where $\mathcal{B}e(\cdot)$ denotes the beta distribution. Furthermore, an auxiliary function is used to measure the accuracy of the estimated current speed and value that is obtained from the speed head and the value head, respectively, to help the agent make decisions [125]. The auxiliary function is defined as:

$$\mathcal{L}_{\text{aux}} = \lambda_{\text{value}}\|v_{t-\delta_{eu}} - {}^{\text{ex}}v_{t-\delta_{eu}}\|_1 + \lambda_{\text{speed}}\|s_{t-\delta_{eu}} - {}^{\text{ex}}s_{t-\delta_{eu}}\|_2, \tag{5.30}$$

where λ_{value} and λ_{speed} are hyperparameters. Thus, the overall loss function of the DTCP is defined as:

$$\mathcal{L}_{\text{DTCP}} = \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{ctrl}}\mathcal{L}_{\text{ctrl}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}, \tag{5.31}$$

where λ_{traj}, λ_{ctrl}, and λ_{aux} are hyperparameters. The design of the loss functions in Eq. (5.28), Eq. (5.29), and Eq. (5.30) follows a consistent principle: combining an output loss and a feature loss through a weighted summation. We consider these weights essential because the two types of variables (e.g., waypoints and trajectory features) typically have different scales and ranges, requiring proper balancing to ensure meaningful contributions from each term. For the overall loss function in Eq. (5.31), the weights (λ_{traj}, λ_{ctrl}, λ_{aux}) are carefully chosen to ensure that each component contributes appropriately to the task objective, aligning with the goal of achieving better system performance.

5.5.6 Joint Training

To jointly train the DTCP and task-oriented JSCC, we employ imitation learning, specifically through behavior cloning. In this approach, the agent learns to perform tasks by replicating the actions of experts based on a dataset of expert demonstrations. Behavior cloning works by directly mapping observed states to corresponding actions, allowing the agent to learn a policy that mirrors the expert's behavior. This approach is particularly effective in scenarios where a large amount of labeled data is available, allowing the agent to generalize from the expert's actions to similar situations encountered during autonomous driving.

Algorithm 5 Joint Training of DTCP and Task-Oriented JSCC.

Initialization: Initialize the neural network parameters ϕ and ψ.
Input: Image dataset $\mathcal{X}$ with corresponding ground-truth agent output $\mathcal{A}$ and state information $\mathcal{M}$.
while not converged **do**
  Sample mini-batch $\{(\boldsymbol{a}_i, \boldsymbol{x}_i)\}_{i=1}^{K_b}$ from $\mathcal{A}$ and $\mathcal{X}$ with corresponding state information $\{\boldsymbol{m}_i\}_{i=1}^{K_b}$ from $\mathcal{M}$.
  for sample $i = 1, \ldots, K_b$ **do**
    Encode image to JSCC symbols: $\boldsymbol{z}_i \leftarrow f_e(\boldsymbol{x_i})$.
    Transmit JSCC symbols through channel and apply equalization: $\hat{\boldsymbol{z}}_i \leftarrow f_h(\boldsymbol{z}_i)$.
    Estimate mean and standard deviation: $\boldsymbol{\mu}_i \leftarrow f_{\boldsymbol{\mu}}(\hat{\boldsymbol{z}}_i)$, $\boldsymbol{\sigma}_i \leftarrow f_{\boldsymbol{\sigma}}(\hat{\boldsymbol{z}}_i)$.
    Compute KL divergence $D_{\text{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}_i)\|q(\hat{\boldsymbol{z}}))$.
    Generate estimated action: $\hat{\boldsymbol{a}}_i \leftarrow f_a(\hat{\boldsymbol{z}}_i, \boldsymbol{m}_i, \boldsymbol{h}_i)$.
    Compute DTCP loss based on Eq. (5.31).
  end for
  Compute joint loss $\mathcal{L}'_{\text{VIB}}$ of this mini-batch based on Eq. (5.34).
  Update neural network parameters: $\phi \overset{+}{\leftarrow} -\nabla_\phi \mathcal{L}'_{\text{VIB}}$, $\psi \overset{+}{\leftarrow} -\nabla_\psi \mathcal{L}'_{\text{VIB}}$.
end while

Assuming the posterior $q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})$ follows a Gaussian distribution

$$\mathcal{N}(\mu_\psi(\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h}), \sigma^2_{\text{const}} I), \tag{5.32}$$

where $\mu_\psi(\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})$ maps reconstructed JSCC symbols $\hat{\boldsymbol{z}}$, state information $\boldsymbol{m}$, and channel state $\boldsymbol{h}$ to the mean of a Gaussian distribution and σ_{const} is a constant, we can derive the following expression:

$$-\log q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h}) \sim \frac{1}{2\sigma^2_{\text{const}}} \|\boldsymbol{a} - \mu_\psi(\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})\|_2^2, \tag{5.33}$$

where $\mu_\psi(\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h}) = \hat{\boldsymbol{a}}$. Eq. (5.33) shows that $-\log q_\psi(\boldsymbol{a}|\hat{\boldsymbol{z}}, \boldsymbol{m}, \boldsymbol{h})$ can serve as a distance metric, analogous to the square of the l_2-norm. From this perspective, we heuristically regard the loss function of DTCP as an extension of the first term in Eq. (5.18), thus we can jointly optimize DTCP with task-oriented communication as follows:

$$\mathcal{L}'_{\text{VIB}} := \frac{1}{K_b} \sum_{i=1}^{K_b} \Big\{ \mathcal{L}_{\text{DTCP}} + \beta D_{\text{KL}}(p_\phi(\hat{\boldsymbol{z}}|\boldsymbol{x}_i)\|q(\hat{\boldsymbol{z}})) \Big\}. \tag{5.34}$$

The joint training process of proposed task-oriented co-design is shown in Algorithm 5.

5.6 Performance Evaluation

In this section, we present a case study of our proposed task-oriented co-design framework. The evaluation is carried out within the simulator CARLA, which offers a variety of urban environments that closely mimic real-world traffic scenarios.

5.6.1 Experimental Setup

Dataset We utilize the well-structured dataset provided by [125], which consists of images (height = 256, width = 900, channels = 3) captured from various urban environments, along with the corresponding vehicle state information. Specifically, the dataset contains $K_a = 189524$ images from four maps (Town01, Town03, Town04, and Town06) for training, and 27201 images from four different maps (Town02, Town05, Town07, and Town10) for testing. This well-structured dataset allows us to effectively train and validate the proposed framework across a range of real-world-like scenarios.

To train DTCP using behavior cloning, we use Roach [118] as the expert agent in our experiments. Roach is a highly capable autonomous driving agent that relies on Bird's-eye View (BEV) as input. Since BEV data are challenging to collect in real time for real-world autonomous driving, this highlights the importance of training autonomous driving agents using data from standard sensors. Our proposed DTCP, equipped with only one camera, demonstrates strong potential for practical deployment in real-world scenarios.

Evaluation The experiment is designed to evaluate the driving performance of the proposed task-oriented co-design framework against established baselines under varying communication conditions. These conditions include significant communication latency, constrained bandwidth, and the presence of noisy fading channels. The baselines for comparison include three widely recognized image coding techniques: 1) JPEG [3]; 2) JPEG2000 [4]; and 3) BPG [131]. Each coding method is followed by (2048, 6144) LDPC codes with a 64-QAM digital modulation scheme.

In addition, two JSCC-based methods, referred to as "JSCC-AE" [14] and "JSCC-VAE" [15], are also included as baselines. JSCC-AE is a seminal work that introduced the concept of joint source-channel coding without relying on explicit separate codes for compression or error correction, making it a foundational approach in this area. Based on this, JSCC-VAE offers robustness against variations in channel conditions, further enhancing its practical applicability. These methods focus on accurately reconstructing the image at the edge server, but do not co-design with the autonomous driving agent (DTCP). Furthermore, the baseline includes [125], which performs driving tasks using uncompressed images, denoted as "TGCP."

Driving performance is quantified using the established driving score metric[2] of CARLA, which evaluates the vehicle's ability to follow predefined waypoints, reach target destinations, and comply with traffic regulations. To ensure robustness, each experiment is repeated three

[2] `https://leaderboard.carla.org/`

times on a selected route in Town05, under four distinct weather conditions: clear noon, cloudy sunset, soft rain at dawn, and heavy rain at night.

In Chapter 4, the driving scores for all methods are below 26, despite the theoretical upper bound being 100. This indicates that all methods, including TGCP (i.e., the agent driving with raw images), perform relatively poorly in those specific test scenarios, making performance comparisons less informative. To enable a more meaningful and intuitive comparison in Chapter 5, we selected road sections where TGCP achieves a perfect driving score of 100. This ensures that the baseline performance is strong, and any performance degradation observed in the compressed or task-oriented communication cases can be more clearly attributed to the proposed design, rather than to the difficulty of the scenario itself.

Parameters Settings For the task-oriented JSCC encoder, we configure the dimension of the JSCC symbols to $l_z = 1024$, achieving a significant low bandwidth compression ratio of $l_z/l_x \approx 0.0015$. The average power constraint P_{target} for JSCC symbols is fixed at 1. The predefined Gaussian distribution is assumed to be $q(\hat{\boldsymbol{z}}) \sim \mathcal{N}(0, I)$. In addition, "JSCC-AE" and "JSCC-VAE" use the same network structure as the proposed task-oriented JSCC for fair comparisons.

For DTCP, the parameters are configured as follows: $\lambda_c = 0.7$, $\lambda_{\text{feat}} = 0.05$, $\lambda_{\text{value}} = 0.001$, $\lambda_{\text{speed}} = 0.05$, and $\lambda_{\text{traj}} = \lambda_{\text{ctrl}} = \lambda_{\text{aux}} = 1$. The values of $\lambda_{[\cdot]}$ were selected based on our preliminary tests, which ensure a stable training process and achieve relatively optimal driving performance. In our preliminary tests, we observed that excessively large ($\beta > 0.01$) or small ($\beta < 0.000001$) values of β can disrupt the balance between the IB terms, leading to training instability and crashes. To address this, we set $\beta = 0.0001$ for jointly training JSCC encoder and DTCP under the IB objective. This β value creates a reasonable balance between the two IB terms, ensuring stable training and achieving good overall performance. For the mini-batch, we set $K_b = 32$. Moreover, the duration of each time slot τ is synchronized with the simulation time step of CARLA, which is 0.05 seconds. The neural network architectures of the proposed JSCC encoder and DTCP are shown in Fig. 5.6.

For the OFDM system, the parameters are set to: $N_{\text{sub}} = 12$, $N_{\text{path}} = 8$, $\gamma = 4$, and the length of the cyclic prefix (CP) is 3.

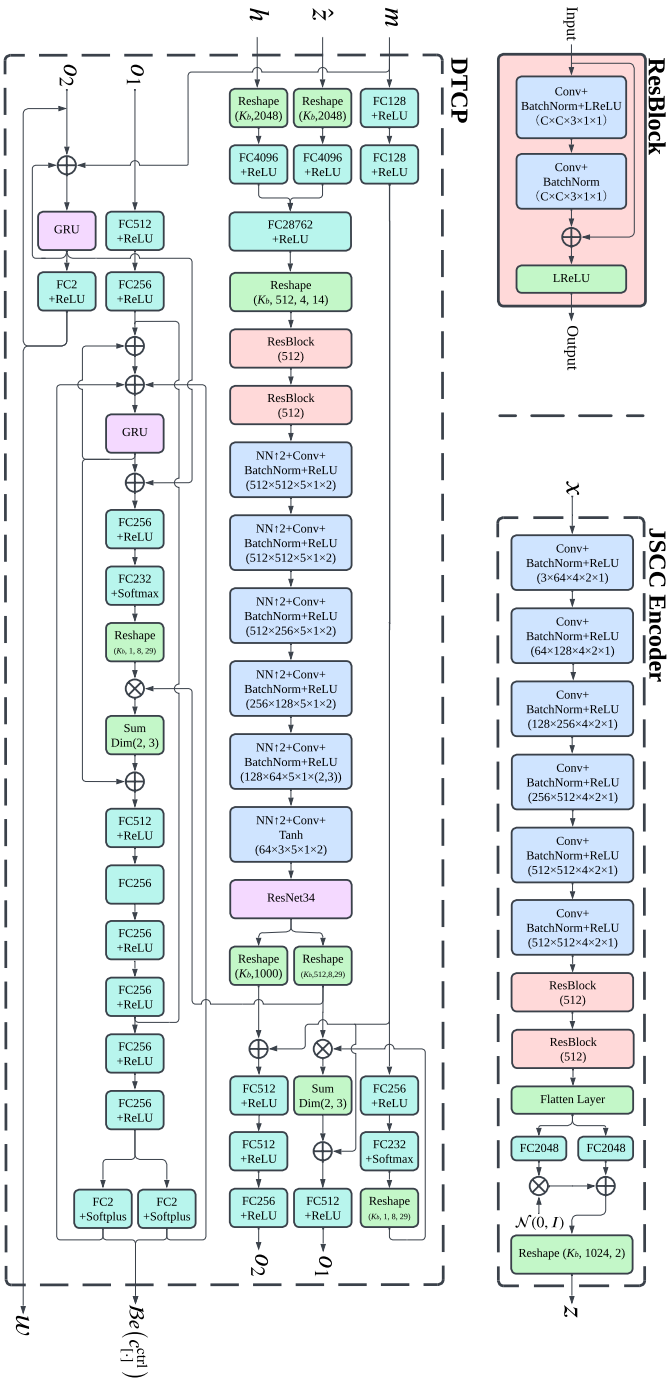


Figure 5.6: Neural network architecture of the proposed JSCC encoder and DTCP. The main components are annotated as follows: **Conv**: Convolutional layer, with parameters specified as (*input channel size* × *output channel size* × *kernel size* × *stride* × *padding*). **FC**: Fully-connected layer, where the following number indicates the output dimensions. **NN↑2**: Nearest neighbor upsampling. **ResBlock**: Residual block, with parameters specifying the input and output channel sizes. **Reshape**: Reshaping layer, with parameters specifying the target dimensions. **LReLU**: Leaky ReLU activation function with $\alpha = 0.2$. **Softplus**: Softplus activation function. **Sum Dim(2,3)**: Summation operation performed along dimensions 2 and 3 [125]. **GRU**: Gated Recurrent Unit (GRU) [155]. Connection points o_1 and o_2 represent linked points, specifically, all instances of o_1 are interconnected, as are all instances of o_2.

5.6.2 Evaluation on CARLA

Constrained Bandwidth Compression Ratio

The effect of the bandwidth compression ratio on the driving score is illustrated in Fig. 5.7. When the required driving score is 90, the proposed DTCP secures substantial reductions in bandwidth usage by at least 99.19% compared to traditional coding methods. In contrast, if bandwidth compression ratios are drastically reduced for traditional coding methods (i.e., less than 0.05), the corresponding reduction in image quality leads to a severe degradation in driving performance, with driving scores struggling to exceed 20. This comparison shows the limits of conventional approaches under extreme bandwidth constraints and showcases the superior adaptability of our task-oriented co-design framework in such challenging scenarios.

Noisy Fading Channel

In Fig. 5.8, we analyze variations in driving performance as a function of SNR under an OFDM channel. Drawing from the findings in the constrained bandwidth compression ratio experiment, we set the bandwidth compression ratios to 0.232 for JPEG, 0.251 for JPEG2000, and 0.183 for BPG, where traditional methods perform comparably to DTCP, with driving scores consistently higher than 90, ensuring a fair comparison.

When SNR ≥ 15 dB, the proposed DTCP framework performs similarly to traditional coding methods. When SNR $= 10$ dB, JPEG, JPEG2000, and BPG occasionally encounter decoding errors, leading to driving scores below 72, while DTCP still maintains a driving score above 89. As SNR drops below 5 dB, the DTCP framework continues to maintain robust driving performance, with scores remaining above 49. Specifically, the DTCP framework achieves driving scores of 59.78 at SNR = 5 dB and 49.29 at SNR = 0 dB. In contrast, severe noise significantly hampers the performance of systems utilizing traditional coding methods when SNR is lower than 5 dB, causing frequent decoding failures and dramatically low driving scores (below 21 when SNR $= 5$ dB and below 2 when SNR $= 0$ dB). Additionally, both the JSCC-AE and JSCC-VAE methods consistently produce driving scores below 20 across all SNR levels, highlighting the importance of task-oriented co-design in transmitting task-critical information.

These results underscore the resilience of our task-oriented co-design framework under adverse noise conditions, demonstrating its ability to maintain effective performance even in highly challenging environments. Moreover, the framework achieves excellent scores under regular channel conditions while simultaneously achieving bandwidth savings of at least 99.19%. This highlights the efficiency and robustness of the DTCP approach, making it a viable solution for real-world scenarios where communication channels are unreliable and constrained.

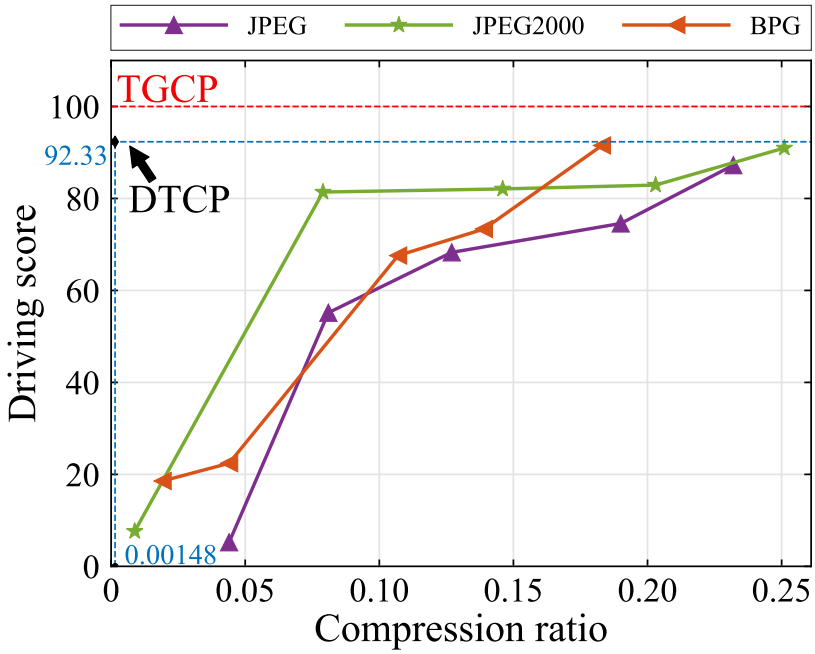


Figure 5.7: Driving scores of traditional coding methods with varied bandwidth compression ratios under OFDM channel with SNR = 20 dB.

Selection of JSCC symbols

Given the characteristics of JSCC, symbols with relatively low energy are particularly vulnerable to noise. To optimize the trade-off between bandwidth compression and driving performance, we explore the selection of generated JSCC symbols, aiming to further reduce the bandwidth while maintaining the required driving performance.

As shown in Fig. 5.9, the number of selected JSCC symbols varies from 168 to 1008, in increments of 168, while the corresponding bandwidth compression ratio varies from 0.00024 to 0.00146. The choice of 168 as the incremental step size is based on the structure of a 5G resource block, which consists of 12 subcarriers and 14 OFDM symbols per slot, totaling 168 resource elements [156]. This value is a natural fit for our simulation setup, as it aligns with the granularity of resource allocation in modern cellular networks, making the results more relevant for real-world applications.

The driving score exhibits a gradual decline (from 89.28 to 80.81) as the number of selected JSCC symbols decreases from 1008 to 504. However, the driving score drops sharply (from 80.81 to 52.15) when the number of selected JSCC symbols is reduced further from 504 to 168. This significant drop suggests that high-energy JSCC symbols are more critical to task performance, as they carry essential information required for accurate decision-making in autonomous driving tasks.

Our proposed DTCP framework demonstrates the ability to maintain a driving score above 80 by transmitting only the top 504 high-energy JSCC symbols. This selective transmission strategy demonstrates the potential to reduce communication overhead by 50.78% compared to transmitting 1024 JSCC symbols. Achieving this reduction depends on adequately mitigating index transmission overhead, which could be addressed by applying techniques such as

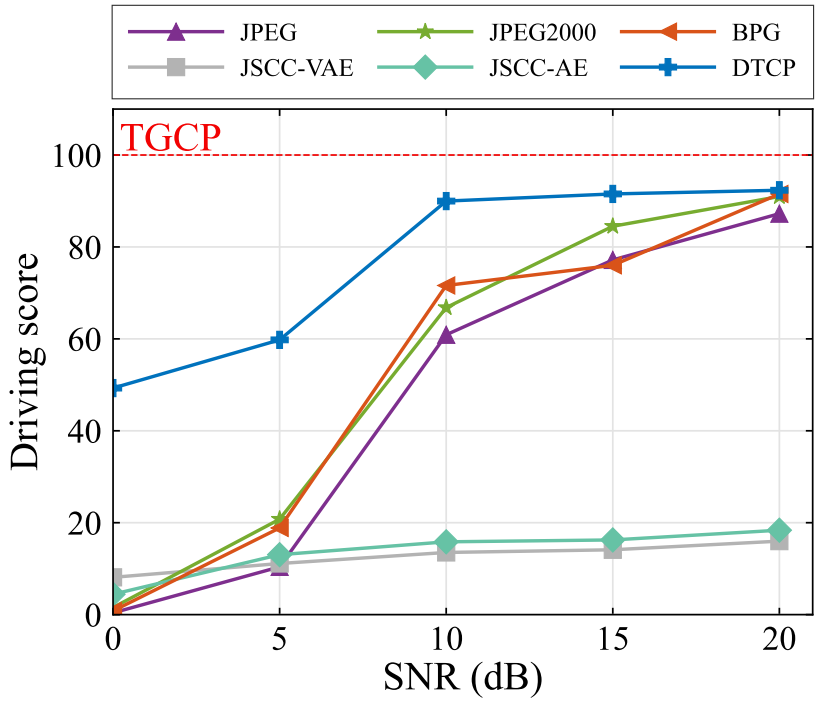


Figure 5.8: Driving scores with varied SNRs under OFDM channel.

VL-VFE [11]. This approach not only optimizes bandwidth usage but also ensures robust driving performance.

Compensate Perceived E2E Delay

The impact of communication delays on driving performance using the DTCP framework is presented in Fig. 5.10. The delay ranges from 0 to 20 time slots, increasing by increments of 2 time slots. We evaluate five distinct configurations within the DTCP framework:

- **DTCP-1**: Transmits all JSCC symbols and generates commands based on Eq. (5.25) with parameters $l = 1024$, $l_p = \delta$, and $\delta_T = 10$. This is also the default configuration of DTCP in previous experiments.
- **DTCP-2**: Selects 504 JSCC symbols for transmission, generating commands according to Eq. (5.25) with $l = 504$, $l_p = \delta$, and $\delta_T = 10$.
- **DTCP-3**: Transmits all JSCC symbols and always generates predicted commands for the turning situation ($l = 1024$, $l_p = \delta$, and $\delta_T = 0$).
- **DTCP-4**: Transmits all JSCC symbols but generates commands without prediction ($l = 1024$, $l_p = 0$, and $\delta_T \to \infty$).
- **DTCP-5**: Transmits all JSCC symbols and always generates predicted commands for all situations ($l = 1024$, $l_p = \delta$, and $\delta_T = 0$).

In this experiment, BPG with a bandwidth compression ratio of 0.183 is used as a representative baseline.

The results show that DTCP-5 experiences a steep decline in driving scores, falling below 61 even with a delay of just 2 time slots, and continues to decrease with increasing delay. In addition, DTCP-5 also performs worse than BPG in the presence of delays. These results indicate that relying exclusively on predicted commands is unreliable, particularly in non-turning scenarios.

When the delay is less than 10 time slots, DTCP-4 manages to maintain a driving score greater than 80 without relying on predicted commands. However, beyond 10 time slots, driving performance sharply declines, highlighting the limitations of unpredicted commands in high-latency conditions. In particular, when the delay is less than 8 time slots, the superior performance of DTCP-4 compared to DTCP-3 demonstrates that receiving accurate commands, even with some latency, is more critical than receiving inaccurate predicted commands under low latency. In contrast, when the delay exceeds 10 time slots, DTCP-3 outperforms DTCP-4, showing that adopting predicted commands becomes more effective in high-latency environments.

DTCP-1 combines the advantages of DTCP-3 and DTCP-4, offering the most balanced performance by dynamically switching between unpredicted and predicted commands based on E2E delay. It outperforms DTCP-4 and BPG significantly by 20.39 and 21.38 points at $\delta = 16$, 35.78 and 35.69 points at $\delta = 18$, and 26.95 and 31.59 points at $\delta = 20$, respectively. Furthermore, DTCP-1 outperforms DTCP-3 when the delay is less than 10 time slots.

DTCP-2, while leading to an average reduction of 17 points compared to DTCP-1, shows the potential to preserve 50.78% of communication resources. Despite the decrease in driving performance, DTCP-2 still maintains a driving score above 50, even with delays of up to 12 time slots. Furthermore, the driving scores of DTCP-2 exceed BPG and DTCP-4 when the delay is greater than 8 and 16 time slots, respectively, highlighting the efficiency of DTCP-2 in managing significant delays and offering a viable trade-off between communication bandwidth and driving performance.

The delay range from 0.05 s to 1 s (i.e., 1 to 20 time slots with each slot being 0.05 s) is intentionally chosen to evaluate the impact of E2E delay on driving performance and to demonstrate the effectiveness of the proposed prediction mechanism in compensating for such delays. While a 1-second E2E delay may seem large, it is not uncommon in practical deployments where the system involves high-resolution sensor data transmission, edge server queuing, computation, and control signal feedback, particularly in congested or lossy wireless networks or under adverse channel conditions. Including a delay of up to 1 s allows us to stress-test the system and analyze its robustness under extreme but plausible conditions. In this experiment, a 1-second delay leads to a substantial performance drop (from a perfect score of 100 to around 20 for the DTCP-4 baseline) that highlights the severity of delayed perception and action. In contrast, our proposed prediction mechanism (DTCP-1) mitigates the impact of delay and maintains a driving score of approximately 50, clearly demonstrating its practical value and robustness under realistic high-latency conditions.

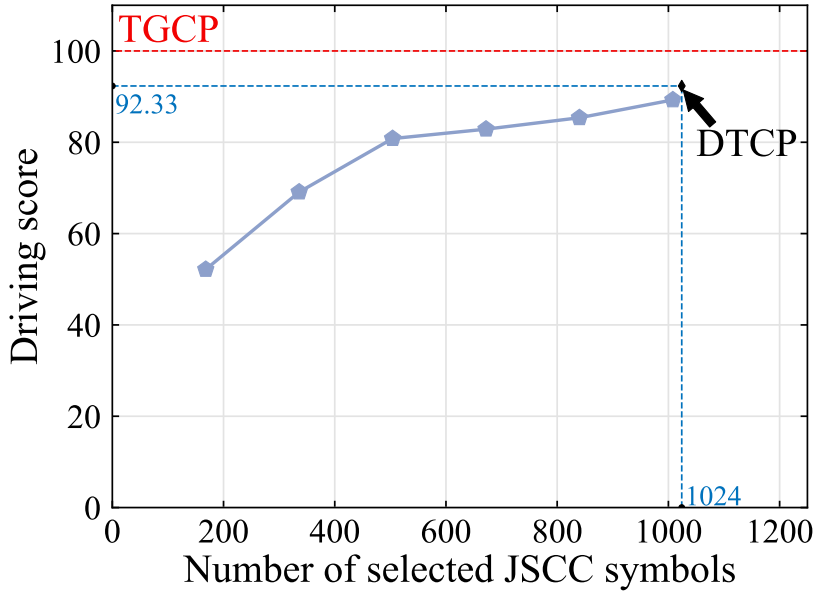


Figure 5.9: Driving scores with varied selected JSCC symbols under OFDM channel with SNR = 20 dB.

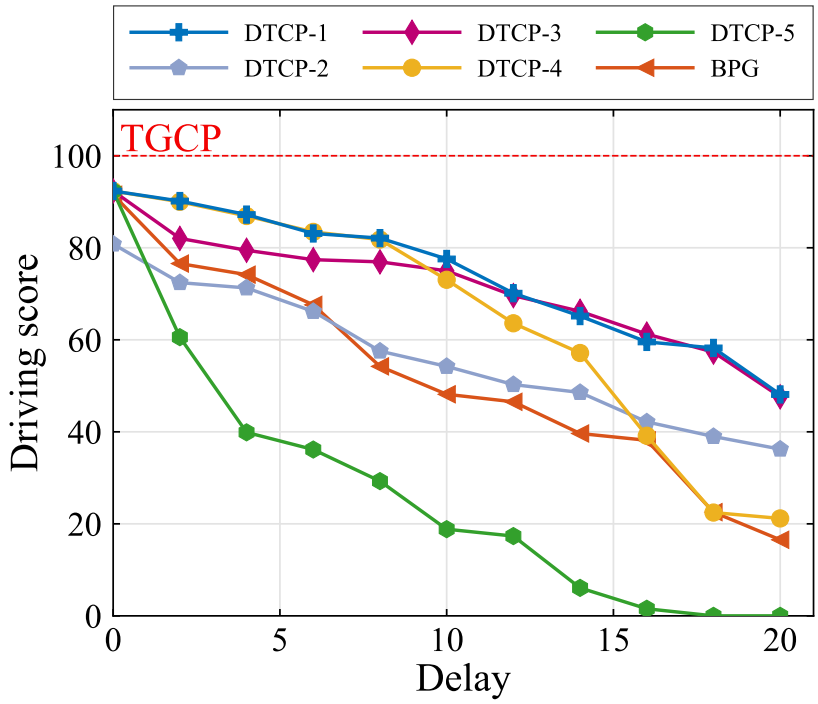


Figure 5.10: Driving scores with varied delays under OFDM channel with SNR = 20 dB.

5.7 Conclusion

In this chapter, we introduced a novel task-oriented co-design framework that integrates communication, computing, and control, specifically tailored for edge-enabled industrial CPS. By leveraging task-oriented JSCC through VIB theory, our approach effectively dis-

cards task-agnostic information, resulting in significant savings in communication bandwidth. Furthermore, with the incorporation of delay awareness into the trajectory-guided control prediction framework, the proposed DTCP framework adaptively generates predicted commands based on real-time delay, thereby maintaining driving performance even with significant latency.

Extensive evaluations using the CARLA simulator demonstrate that the task-oriented co-design framework significantly improves driving performance under conditions of constrained bandwidth, noise interference, and varying communication delays. The proposed DTCP consistently outperforms traditional methods across multiple scenarios. In particular, with an E2E delay of 1 second (equivalent to 20 time slots in CARLA), our framework achieves a driving score of 48.12, which is 31.59 points higher than when using BPG, while also reducing bandwidth usage by 99.19%. Moreover, our analysis of compensating for perceived E2E delay highlights the inherent unreliability of prediction under certain conditions, underscoring the need to balance predicted and unpredicted commands for optimal system performance. There are several promising directions for future research based on this chapter, such as extending the framework to more realistic wireless environments, including Urban Micro (UMi) and Urban Macro (UMa), and dynamically optimizing coding rates and modulation schemes based on channel conditions and SNR, leveraging 5G Modulation and Coding Scheme (MCS).

Chapter 6

Conclusions and Future Directions

6.1 Conclusions and Discussion

6.1.1 Summary

In this thesis, we explore task-oriented communication for edge intelligence enabled connected robotics systems, focusing on optimizing data transmission, processing, and decision-making for mission-critical applications.

Key contributions of this thesis include:

1. **Task-Oriented Source-Channel Coding (TSCC):** We proposed a novel TSCC framework to optimize data transmission for edge-enabled autonomous driving, which significantly reduces communication bandwidth usage while preserving task-critical information. We leverage an autonomous driving agent to guide source-channel coding based on a modified CVAE. We test the proposed framework on a well-known autonomous driving platform (CARLA) with different communication channel conditions. The experimental results show that compared to traditional communication and state-of-the-art deep JSCC, TSCC achieves superior performance by saving 98.36% communication overhead and maintains an 83.24% driving score even at 0 dB SNR.

2. **Aligned Task- and Reconstruction-Oriented Communication (ATROC):** We proposed integrating information bottleneck principles with deep JSCC to align task- and reconstruction-oriented communication. The idea is to extend the Information Bottleneck (IB) theory to optimize data transmission by minimizing task-relevant loss function, while maintaining the structure of the original data by an information reshaper. We also introduce a JSCC modulation scheme compatible with classical modulation techniques, which enables the deployment within existing infrastructures. Our evaluation in the CARLA simulator demonstrates that the proposed framework significantly reduces bits per service by 99.19% compared to existing methods, such as JPEG, JPEG2000, and BPG, without compromising the effectiveness of task execution.

3. **Task-Oriented Co-Design of Communication, Computing, and Control:** We developed a Delay-Aware Trajectory-Guided Control Prediction (DTCP) framework for

real-time decision-making in industrial CPS. In addition, the DTCP is co-designed with task-oriented JSCC, focusing on transmitting task-specific information for timely and reliable autonomous driving. Experimental results in the CARLA simulator demonstrate that, under an E2E delay of 1 second (20 time slots), the proposed framework achieves a driving score of 48.12, which is 31.59 points higher than using Better Portable Graphics (BPG) while reducing bandwidth usage by 99.19%.

Our comprehensive evaluations demonstrated the effectiveness of the proposed frameworks in enhancing communication efficiency, system robustness, and real-time decision-making capabilities.

6.1.2 Generalization Capabilities of the Proposed Method

The proposed task-oriented communication framework and associated methods demonstrate a promising degree of generalization due to their reliance on deep learning architectures. The generalization capability of the proposed methods stems from the use of the VIB framework. VIB inherently encourages the model to extract and preserve mission-critical information while discarding redundant task-agnostic data. This selective preservation of information leads to learned representations that are more robust to variations in channel conditions, sensor inputs, and operational contexts. As a result, the proposed framework can generalize well across different noise scenarios and SNR conditions without the need for retraining.

In addition, the integration of JSCC modulation with traditional constellation diagrams (e.g., QAM) ensures compatibility with existing digital communication infrastructures. This compatibility facilitates deployment within diverse, real-world network infrastructures that vary widely in communication standards and protocols.

Moreover, the deep learning architectures used in this thesis have been carefully designed and trained with varied environmental scenarios. This approach helps the system achieve good performance across different operational environments. For instance, although trained primarily on the CARLA simulator for autonomous driving tasks, the proposed method's design principles could generalize to other robotics or CPS applications, such as robotic manipulation, drone navigation, and remote monitoring, by retraining on the relevant task-specific datasets.

However, it is essential to acknowledge that the generalization capability of deep learning-based methods inherently depends on the diversity and quality of the training data. If training data do not adequately represent the real-world variations (e.g., varying lighting conditions, dynamic obstacles, and multiple weather scenarios), the generalization performance may be limited when deployed in environments significantly different from the training conditions.

Although extensive simulation experiments and careful model design suggest promising generalization capabilities, additional real-world validation is necessary. Future work should include evaluating the proposed methods across multiple real-world scenarios and different sensor modalities to rigorously test and further improve their generalization capabilities.

6.1.3 Comprehensive Reflections and Research Outlook

At the early stage, I explored the potential of online training for JSCC neural networks within a closed-loop autonomous driving simulator (CARLA). However, the high computational cost and slow simulation speed made real-time or online training infeasible. This limitation led to a pragmatic shift toward using an offline dataset for training, effectively framing the learning process as an imitation learning or knowledge distillation problem. Although this approach provided a solid foundation, it also revealed the limitations of static training when applied to dynamic environments.

This experience highlights a critical future direction: integrating reinforcement learning or online adaptation mechanisms with JSCC models, provided that simulation platforms or real-world deployment environments can support faster or more efficient data generation and collection. Such integration would enable communication strategies to adapt on-the-fly to environmental changes such as varying channel conditions, sensor configurations, or task requirements.

One important fact is that the effectiveness of a task-oriented JSCC model is linked to the capability of the downstream task agent. The performance ceiling of the agent limits the potential benefit of communication optimization. Therefore, advances in autonomous driving models (e.g., more accurate, robust, and generalizable perception and control agents) can directly translate into improvements in JSCC design. Future work may benefit from co-optimizing the communication model and the task agent.

Additionally, this thesis focuses on visual input in the form of RGB images. Although this is a practical and common choice, real-world autonomous systems typically rely on multimodal sensory inputs, including depth images, LiDAR point clouds, and voxel-based representations. A promising research direction lies in extending the task-oriented JSCC framework to handle multimodal fusion, which would require careful design to balance modality-specific compression with shared task relevance across input types.

Finally, this work remains at the simulation level. Validating the proposed methods on real-world hardware, such as embedded communication modules and autonomous robots, would offer critical insights into latency, noise, interoperability, and deployment feasibility. In particular, implementing and evaluating the proposed ATROC framework in a hardware testbed would provide practical validation of its task-oriented benefits under real-time constraints.

Overall, this thesis lays the groundwork for a new generation of intelligent and efficient systems where perception, transmission, and action are co-designed for the task at hand. It also opens several rich avenues for future exploration, combining theoretical rigor with practical relevance across simulation, learning, and deployment.

6.2 Future Directions

While this thesis has addressed several key challenges, there remain numerous directions for further research and development:

1. **Extension to More Realistic Wireless Environments**: Future studies could explore the deployment of the proposed frameworks in complex real-world communication environments, such as Urban Micro (UMi) and Urban Macro (UMa) scenarios, to further assess robustness against channel variations and interference.

2. **Adaptive and Dynamic Task-Oriented Communication**: The integration of adaptive learning mechanisms to dynamically optimize coding rates, modulation schemes, and inference strategies based on real-time channel conditions and system requirements remains an open challenge.

3. **Integration with 5G and Beyond Networks**: Leveraging advanced wireless technologies, such as 5G Modulation and Coding Scheme (MCS) and future 6G paradigms, could provide new opportunities to improve communication efficiency and reliability in edge intelligence systems.

4. **Security and Privacy Enhancements**: Further research is needed to address privacy concerns related to task-oriented communication, such as developing a privacy-preserving task-oriented JSCC.

5. **Human-in-the-Loop Edge Intelligence**: Incorporating human feedback into task-oriented communication and decision-making frameworks could improve adaptability and robustness, particularly in dynamic and unpredictable environments.

By addressing these challenges and advancing task-oriented communication frameworks, future research can contribute to the continued evolution of intelligent connected autonomous systems, fostering their deployment in safety-critical and resource-constrained applications across various scenarios.

Appendix A

Modeling Frequency-Selective Channel

We consider a multipath fading channel described by a discrete channel transfer function:

$$\check{\boldsymbol{z}}_{\text{time}} = \boldsymbol{h}_{\text{time}} * \boldsymbol{z}_{\text{time}} + \boldsymbol{n}_{\text{time}}, \tag{A.1}$$

where $*$ denotes the convolution operation. Here, $\check{\boldsymbol{z}}_{\text{time}}$ and $\boldsymbol{z}_{\text{time}}$ are the received and transmitted signals in the time domain, respectively, while $\boldsymbol{n}_{\text{time}}$ represents the additive Gaussian noise. The impulse response $\boldsymbol{h}_{\text{time}} = [h_{\text{time-}0}, \cdots, h_{\text{time-}(N_{\text{path}}-1)}]$ captures the multipath effect, where $h_{\text{time-}i} \sim \mathcal{CN}(0, \sigma_i^2)$ for $i = 0, 1, \cdots, N_{\text{path}} - 1$. We assume that path power decays exponentially as $\sigma_i^2 = \alpha_i e^{-\frac{i}{\gamma}}$, with α_i ensuring power normalization $\sum_{i=0}^{N_{\text{path}}-1} \sigma_i^2 = 1$. Here, γ is a delay spread constant.

To simplify, we assume synchronized transmission/reception without carrier frequency offset, and perfectly estimated channel state information by block-type pilot symbols. First, JSCC symbols $\boldsymbol{z} \in \mathbb{C}^{l_z}$ are padded with $N_{\text{sub}} - (l_z \bmod N_{\text{sub}})$ zeros and reshaped to $\boldsymbol{z}_r \in \mathbb{C}^{N_{\text{sym}} \times N_{\text{sub}}}$, where $N_{\text{sym}} = \lceil l_z / N_{\text{sub}} \rceil$ denotes the number of OFDM symbols and N_{sub} represents the number of subcarriers per OFDM symbol. When l_z / N_{sub} is not an integer, the subcarriers in the final OFDM symbol are not fully utilized for driving, but can be used for other tasks as needed.

Next, the Inverse Discrete Fourier Transform (IDFT) and cyclic prefix (CP) are applied, followed by transmission through the multipath channel as described in Eq. (A.1). The receiver removes the CP and applies the Discrete Fourier Transform (DFT) to yield the received JSCC symbols $\check{\boldsymbol{z}}_r \in \mathbb{C}^{N_{\text{sym}} \times N_{\text{sub}}}$. Therefore, we have the following equation:

$$\check{\boldsymbol{z}}_r[j,k] = \boldsymbol{h}_r[j,k]\boldsymbol{z}_r[j,k] + \boldsymbol{n}_r[j,k], \tag{A.2}$$

where k denotes the k_{th} subcarrier, j denotes the j_{th} OFDM symbol, and $\boldsymbol{h}_r[j,k] = \boldsymbol{h}_r[j',k]$, $\forall j, j' \in \{1, \cdots, N_{\text{sym}}\}$ represents the subcarrier-specific channel response. Flatting each term and removing the dimensions of driving-irrelevant subcarriers lead to the simplified expression:

$$\check{\boldsymbol{z}} = \boldsymbol{h} \cdot \boldsymbol{z} + \boldsymbol{n},$$

where $\check{\boldsymbol{z}} \in \mathbb{C}^{l_z}$, $\boldsymbol{h} \in \mathbb{C}^{l_z}$, and $\boldsymbol{n} \in \mathbb{C}^{l_z}$.

Bibliography

[1] W. J. Baker, *A History of the Marconi Company 1874-1965*. 2013.

[2] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.

[3] G. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, 1992.

[4] D. S. Taubman, M. W. Marcellin, and M. Rabbani, "JPEG2000: Image compression fundamentals, standards and practice," *J. Electron. Imag.*, vol. 11, no. 2, pp. 286–287, 2002.

[5] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[6] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, 1999, pp. 368–377.

[7] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 14–20, 1972.

[8] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, 1972.

[9] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop*, 2015, pp. 1–5.

[10] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[11] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 197–211, 2022.

[12] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multidevice cooperative edge inference," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 73–87, 2023.

[13] Q. Liao and T.-Y. Tung, "AdaSem: Adaptive goal-oriented semantic communications for end-to-end camera relocalization," in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 1111–1120.

[14] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, 2019.

[15] Y. M. Saidutta, A. Abdi, and F. Fekri, "Joint source-channel coding over additive noise analog channels using mixture of variational autoencoders," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2000–2013, 2021.

[16] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[17] K. Brandenburg, "MP3 and aac explained," *J. Audio Eng. Soc.*, no. 17-009, 1999.

[18] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, 1950.

[19] A. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 751–772, 1971.

[20] V. H. M. Donald, "Advanced mobile phone service: The cellular concept," *Bell Syst. Tech. J.*, vol. 58, no. 1, pp. 15–41, 1979.

[21] J. Lehenkari and R. Miettinen, "Standardisation in the construction of a large technological system—the case of the nordic mobile telephone system," *Telecommun. Policy*, vol. 26, no. 3-4, pp. 109–127, 2002.

[22] D. Barnes, "The introduction of cellular radio in the united kingdom," in *IEEE Veh. Technol. Conf.*, vol. 35, 1985, pp. 147–152.

[23] S. Faruque and S. Faruque, "Frequency Division Multiple Access (fdma)," *Radio Freq. Mult. Access Tech. Made Easy*, pp. 21–33, 2019.

[24] M. Mouly and M.-B. Pautet, *The GSM system for mobile communications*. 1992.

[25] K. S. Zigangirov, *Theory of code division multiple access communication*. 2004.

[26] C. Peersman, S. Cvetkovic, P. Griffiths, and H. Spear, "The global system for mobile communications short message service," *IEEE Pers. Commun.*, vol. 7, no. 3, pp. 15–23, 2000.

[27] K. Murota and K. Hirade, "GMSK modulation for digital mobile radio telephony," *IEEE Trans. Commun.*, vol. 29, no. 7, pp. 1044–1050, 1981.

[28] J. Cai and D. Goodman, "General packet radio service in GSM," *IEEE Commun. Mag.*, vol. 35, no. 10, pp. 122–131, 1997.

[29] A. Furuskar, S. Mazur, F. Muller, and H. Olofsson, "EDGE: enhanced data rates for GSM and TDMA/136 evolution," *IEEE Pers. Commun.*, vol. 6, no. 3, pp. 56–66, 1999.

[30] F. Hillebrand, *GSM and UMTS: the creation of global mobile communication*. 2002.

[31] A. Samukic, "UMTS universal mobile telecommunications system: Development of standards for the third generation," *IEEE Trans. Veh. Technol.*, vol. 47, no. 4, pp. 1099–1104, 1998.

[32] L. Milstein, "Wideband code division multiple access," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 8, pp. 1344–1354, 2000.

[33] D. Saha and T. Birdsall, "Quadrature-quadrature phase-shift keying," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 437–448, 1989.

[34] H. Holma, A. Toskala, K. Ranta-aho, and J. Pirskanen, "High-speed packet access evolution in 3GPP release 7 [topics in radio communications]," *IEEE Commun. Mag.*, vol. 45, no. 12, pp. 29–35, 2007.

[35] H. Holma and A. Toskala, *WCDMA for UMTS: Radio access for third generation mobile communications*. 2005.

[36] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[37] *ETSI - long term evolution*, https://www.etsi.org/technologies/mobile/4g, 2015.

[38] M. Morelli, C.-C. J. Kuo, and M.-O. Pun, "Synchronization techniques for orthogonal frequency division multiple access (OFDMA): A tutorial review," *Proc. IEEE*, vol. 95, no. 7, pp. 1394–1427, 2007.

[39] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier FDMA for uplink wireless transmission," *IEEE Veh. Technol. Mag.*, vol. 1, no. 3, pp. 30–38, 2006.

[40] D. Love, R. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2735–2747, 2003.

[41] S. Sesia, I. Toufik, and M. Baker, *LTE-the UMTS long term evolution: from theory to practice*. 2011.

[42] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. 2013.

[43] H. Kim, "Enhanced mobile broadband communication systems*," in *Design and Optimization for 5G Wireless Communications*. 2020, pp. 239–302.

[44] C. She, C. Sun, Z. Gu, *et al.*, "A tutorial on ultrareliable and low-latency communications in 6G: Integrating domain knowledge into deep learning," *Proc. IEEE*, vol. 109, no. 3, pp. 204–246, 2021.

[45] C. Bockelmann, N. Pratas, H. Nikopour, *et al.*, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, 2016.

[46] S. Dahmen-Lhuissier, *Multi-Access Edge Computing (MEC)*, `https://www.etsi.org/technologies/multi-access-edge-computing`.

[47] Z. Zhang, Y. Xiao, Z. Ma, *et al.*, "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, 2019.

[48] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2020.

[49] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nat. Electron.*, vol. 3, no. 1, pp. 20–29, 2020.

[50] M. Latva-Aho, K. Leppänen, *et al.*, "Key drivers and research challenges for 6G ubiquitous wireless intelligence," 2019.

[51] E. Calvanese Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, *et al.*, "6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 42–50, 2019.

[52] N. C. Luong, D. T. Hoang, S. Gong, *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[53] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, 2018.

[54] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[55] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning based semantic communications: An initial investigation," in *IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.

[56] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Second edition. 1991.

[57] H. Wei, W. Ni, W. Xu, F. Wang, D. Niyato, and P. Zhang, "Federated semantic learning driven by information bottleneck for task-oriented communications," *IEEE Commun. Lett.*, vol. 27, no. 10, pp. 2652–2656, 2023.

[58] L. Sun, Y. Yang, M. Chen, and C. Guo, "Disentangled information bottleneck guided privacy-protective joint source and channel coding for image transmission," *IEEE Trans. Commun.*, pp. 1–1, 2024.

[59] T. Richardson and R. Urbanke, *Modern coding theory*. 2008.

[60] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[61] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the internet of things (IoT), applications, opportunities, and challenges: A survey," *IEEE Access*, vol. 8, pp. 69 200–69 211, 2020.

[62] J. Hagenauer and T. Stockhammer, "Channel coding and transmission aspects for wireless multimedia," *Proc. IEEE*, vol. 87, no. 10, pp. 1764–1777, 1999.

[63] D. Gündüz, M. A. Wigger, T.-Y. Tung, P. Zhang, and Y. Xiao, "Joint source–channel coding: Fundamentals and recent progress in practical designs," *Proc. IEEE*, pp. 1–32, 2024.

[64] M. Gastpar, B. Rimoldi, and M. Vetterli, "To code, or not to code: Lossy source-channel communication revisited," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1147–1158, 2003.

[65] Y. Diao, Z. Meng, X. Xu, C. She, and P. G. Zhao, "Task-oriented source-channel coding enabled autonomous driving based on edge computing," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2024, pp. 1–6.

[66] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *Claude E. Shannon: Collected Papers*. 1993, pp. 325–350.

[67] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *Proc. Conf. Rec. Asilomar Conf. Signals Syst. Comput.*, 2018, pp. 298–303.

[68] T. Cover, A. Gamal, and M. Salehi, "Multiple access channels with arbitrarily correlated sources," *IEEE Trans. Inf. Theory*, vol. 26, no. 6, pp. 648–657, 1980.

[69] C. Tian, S. N. Diggavi, and S. Shamai, "The achievable distortion region of bivariate Gaussian source on Gaussian broadcast channel," in *Proc. IEEE Int. Symp. Inf. Theory*, 2010, pp. 146–150.

[70] M. Rao, N. Farsad, and A. Goldsmith, "Variable length joint source-channel coding of text using deep neural networks," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2018, pp. 1–5.

[71] S. Yao, K. Niu, S. Wang, and J. Dai, "Semantic coding for text transmission: An iterative design," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 4, pp. 1594–1603, 2022.

[72] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[73] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[74] S. Liu, Z. Gao, G. Chen, Y. Su, and L. Peng, "Transformer-based joint source channel coding for textual semantic communication," in *Proc. IEEE/CIC Int. Conf. Commun. China*, 2023, pp. 1–6.

[75] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, 2021.

[76] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, "Deep learning enabled semantic communications with speech recognition and synthesis," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 6227–6240, 2023.

[77] M. Bokaei, J. Jensen, S. Doclo, and J. Østergaard, "Deep joint source-channel analog coding for low-latency speech transmission over gaussian channels," in *Proc. Eur. Signal Process. Conf.*, 2023, pp. 426–430.

[78] S. Yao, J. Dai, X. Qin, *et al.*, "SoundSpring: Loss-resilient audio transceiver with dual-functional masked language modeling," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 4, pp. 1308–1322, 2025.

[79] D. B. Kurka and D. Gündüz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[80] T.-Y. Tung and D. Gündüz, "DeepWiVe: Deep-learning-aided wireless video transmission," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2570–2583, 2022.

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

[82] S. Wang, J. Dai, Z. Liang, *et al.*, "Wireless deep video semantic transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 214–229, 2023.

[83] D. B. Kurka and D. Gündüz, "Successive refinement of images with deep joint source-channel coding," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2019, pp. 1–5.

[84] T.-Y. Tung, D. B. Kurka, M. Jankowski, and D. Gündüz, "DeepJSCC-Q: Constellation constrained deep joint source-channel coding," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 4, pp. 720–731, 2022.

[85] M. Yang, C. Bian, and H.-S. Kim, "OFDM-guided deep joint source channel coding for wireless multipath fading channels," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 584–599, 2022.

[86] T.-Y. Tung and D. Gündüz, "Deep-learning-aided wireless video transmission," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2022, pp. 1–5.

[87] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, "Less data, more knowledge: Building next generation semantic communication networks," *IEEE Commun. Surv. Tut.*, pp. 1–1, 2024.

[88] W. Yang, H. Du, Z. Q. Liew, *et al.*, "Semantic communications for future internet: Fundamentals, applications, and challenges," *IEEE Commun. Surv. Tut.*, vol. 25, no. 1, pp. 213–250, 2023.

[89] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2021.

[90] E. C. Strinati, P. Di Lorenzo, V. Sciancalepore, *et al.*, "Goal-oriented and semantic communication in 6G AI-native networks: The 6G-GOALS approach," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit*, 2024, pp. 1–6.

[91] S. R. Pandey, V. P. Bui, and P. Popovski, "Goal-oriented communications in federated learning via feedback on risk-averse participation," in *Proc. IEEE Int. Symp. Pers. Indoor Mob. Radio Commun.*, 2023, pp. 1–6.

[92] J. Kang, H. Du, Z. Li, *et al.*, "Personalized saliency in task-oriented semantic communications: Image transmission and performance analysis," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 186–201, 2023.

[93] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 2326–2330.

[94] Q. Hu, G. Zhang, Z. Qin, Y. Cai, G. Yu, and G. Y. Li, "Robust semantic communications against semantic noise," in *Proc. IEEE Veh. Technol. Conf.*, 2022, pp. 1–6.

[95] Y. Diao, Y. Zhang, P. G. Zhao, and D. De Martini, "TAGIC: Task-guided image communication framework for seamless teleoperation," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2024, pp. 1–2.

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

[97] "IEEE standard for low-rate wireless networks," *IEEE Standard 802.15.4-2020*, pp. 1–800, 2020.

[98] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, "Neural joint source-channel coding," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 1182–1192.

[99] Y. Song, M. Xu, L. Yu, H. Zhou, S. Shao, and Y. Yu, "Infomax neural joint source-channel coding via adversarial bit flip," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 5834–5841.

[100] T.-Y. Tung, D. B. Kurka, M. Jankowski, and D. Gündüz, "DeepJSCC-Q: Channel input constrained deep joint source-channel coding," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 3880–3885.

[101] W. Hu, Y. Yang, Y. C. Eldar, C. Feng, and C. Guo, "Digital task-oriented communication with hardware-limited task-based quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2024, pp. 171–175.

[102] D. Xu, T. Li, Y. Li, *et al.*, "Edge intelligence: Empowering intelligence to the edge of network," *Proc. IEEE*, vol. 109, no. 11, pp. 1778–1837, 2021.

[103] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[104] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[105] M. Chen, Y. Hao, Y. Li, C.-F. Lai, and D. Wu, "On the computation offloading at ad hoc cloudlet: Architecture and service modes," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 18–24, 2015.

[106] 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," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[107] X. Huang and S. Zhou, "Dynamic compression ratio selection for edge inference systems with hard deadlines," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8800–8810, 2020.

[108] W. Shi, Y. Hou, S. Zhou, Z. Niu, Y. Zhang, and L. Geng, "Improving device-edge cooperative inference of deep learning via 2-step pruning," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2019, pp. 1–6.

[109] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, 2020.

[110] J. Shao and J. Zhang, "Communication-computation trade-off in resource-constrained edge inference," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 20–26, 2020.

[111] J. Shao and J. Zhang, "BottleNet++: An end-to-end approach for feature compression in device-edge co-inference systems," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2020, pp. 1–6.

[112] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Joint device-edge inference over wireless links with pruning," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun.*, 2020, pp. 1–5.

[113] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 89–100, 2021.

[114] Y. Dubois, B. Bloem-Reddy, K. Ullrich, and C. J. Maddison, "Lossy compression for lossless prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 14 014–14 028.

[115] J. Shao, H. Zhang, Y. Mao, and J. Zhang, "Branchy-GNN: A device-edge co-inference framework for efficient point cloud processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 8488–8492.

[116] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, 2019.

[117] H. A. Ameen, A. K. Mahamad, B. B. Zaidan, *et al.*, "A deep review and analysis of data exchange in vehicle-to-vehicle communications systems: Coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions," *IEEE Access*, vol. 7, pp. 158 349–158 378, 2019.

[118] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 15 222–15 232.

[119] G. Toderici, D. Vincent, N. Johnston, *et al.*, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5435–5443.

[120] E. Agustsson, F. Mentzer, M. Tschannen, *et al.*, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[121] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv: 1312.6114*, 2013.

[122] I. Higgins, L. Matthey, A. Pal, *et al.*, "β-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[123] A. Subramanian, *Pytorch-VAE*, `https://github.com/AntixK/PyTorch-VAE`, 2020.

[124] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Annu. Conf. Robot Learn.*, vol. 78, 2017, pp. 1–16.

[125] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 6119–6132.

[126] P. A. Stavrou and M. Kountouris, "A rate distortion approach to goal-oriented communication," in *Proc. IEEE Int. Symp. Inf. Theory*, 2022, pp. 590–595.

[127] Y. Shi, Y. Zhou, D. Wen, Y. Wu, C. Jiang, and K. B. Letaief, "Task-oriented communications for 6G: Vision, principles, and technologies," *IEEE Wirel. Commun.*, vol. 30, no. 3, pp. 78–85, 2023.

[128] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, 2020.

[129] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *Proc. Workshop Mobile Edge Commun.*, 2018, pp. 31–36.

[130] J. Duchi, "Derivations for linear algebra and optimization," 2007.

[131] F. Bellard. "BPG image format." (2014), [Online]. Available: `https://bellard.org/bpg/`.

[132] L. Chen, Y. Li, W. Silamu, Q. Li, S. Ge, and F.-Y. Wang, "Smart mining with autonomous driving in industry 5.0: Architectures, platforms, operating systems, foundation models, and applications," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 3, pp. 4383–4393, 2024.

[133] "*Study on scenarios and requirements for next generation access technologies*," document 3GPP, TSG RAN TR38.913 R14, 2017.

[134] C. She, Y. Duan, G. Zhao, T. Q. S. Quek, Y. Li, and B. Vucetic, "Cross-layer design for mission-critical IoT in mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9360–9374, 2019.

[135] B. S. Khan, S. Jangsher, A. Ahmed, and A. Al-Dweik, "URLLC and eMBB in 5G industrial IoT: A survey," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1134–1163, 2022.

[136] Z. Hou, C. She, Y. Li, L. Zhuo, and B. Vucetic, "Prediction and communication co-design for ultra-reliable and low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1196–1209, 2020.

[137] Z. Meng, C. She, G. Zhao, and D. De Martini, "Sampling, communication, and prediction co-design for synchronizing the real-world device and digital model in metaverse," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 288–300, 2023.

[138] Z. Meng, K. Chen, Y. Diao, *et al.*, "Task-oriented cross-system design for timely and accurate modeling in the metaverse," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 3, pp. 752–766, 2024.

[139] T.-Y. Tung, S. Kobus, J. P. Roig, and D. Gündüz, "Effective communications: A joint learning and communication framework for multi-agent reinforcement learning over noisy channels," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2590–2603, 2021.

[140] D. Gündüz, F. Chiariotti, K. Huang, A. E. Kalør, S. Kobus, and P. Popovski, "Timely and massive communication in 6G: Pragmatics, learning, and inference," *IEEE BITS Inform. Theory Mag.*, vol. 3, no. 1, pp. 27–40, 2023.

[141] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[142] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[143] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, 2003.

[144] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" In *Proc. IEEE Conf. Comput. Commun.*, 2012, pp. 2731–2735.

[145] J. G. Andrews, R. K. Ganti, M. Haenggi, N. Jindal, and S. Weber, "A primer on spatial modeling and analysis in wireless networks," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 156–163, 2010.

[146] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, 2018.

[147] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2775–2790, 2019.

[148] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016.

[149] J. García, Fern, and o Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 42, pp. 1437–1480, 2015.

[150] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5G-enabled tactile internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 460–473, 2016.

[151] X. Tong, G. Zhao, M. A. Imran, Z. Pang, and Z. Chen, "Minimizing wireless resource consumption for packetized predictive control in real-time cyber physical systems," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2018, pp. 1–6.

[152] F. Richter, Y. Zhang, Y. Zhi, R. K. Orosco, and M. C. Yip, "Augmented reality predictive displays to help mitigate the effects of delayed telesurgery," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 444–450.

[153] X. Hou and S. Dey, "Motion prediction and pre-rendering at the edge to enable ultra-low latency mobile 6DoF experiences," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1674–1690, 2020.

[154] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2498–2507.

[155] K. Cho, B. van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2014, pp. 1724–1734.

[156] 3rd Generation Partnership Project (3GPP), "5G; NR; Physical Channels and Modulation (3GPP TS 38.211 version 16.2.0 Release 16)," 3GPP, Tech. Rep., 2020, [Online]. Available: https://www.3gpp.org.