Deep Learning Enabled Semantic Communication Systems

Huiqiang Xie

Doctor of Philosophy

School of Electronic Engineering and Computer Science

Queen Mary University of London

April 21, 2023

Abstract

In the past decades, communications primarily focus on how to accurately and effectively transmit symbols (measured by bits) from the transmitter to the receiver. Recently, various new applications appear, such as autonomous transportation, consumer robotics, environmental monitoring, and tele-health. The interconnection of these applications will generate a staggering amount of data in the order of zetta-bytes and require massive connectivity over limited spectrum resources but with lower latency, which poses critical challenges to conventional communication systems. Semantic communication has been proposed to overcome the challenges by extracting the meanings of data and filtering out the useless, irrelevant, and unessential information, which is expected to be robust to terrible channel environments and reduce the size of transmitted data. While semantic communications have been proposed decades ago, their applications to the wireless communication scenario remain limited. Deep learning (DL) based neural networks can effectively extract semantic information and can be optimized in an end-to-end (E2E) manner. The inborn characteristics of DL are suitable for semantic communications, which motivates us to exploit DL-enabled semantic communication.

Inspired by the above, this thesis focus on exploring the semantic communication theory and designing semantic communication systems. First, a basic DL based semantic communication system, named DeepSC, is proposed for text transmission. In addition, DL based multi-user semantic communication systems are investigated for transmitting single-modal data and multimodal data, respectively, in which intelligent tasks are performed at the receiver directly. Moreover, a semantic communication system with a memory module, named Mem-DeepSC, is designed to support both memoryless and memory intelligent tasks. Finally, a lite distributed semantic communication system based on DL, named L-DeepSC, is proposed with low complexity, where the data transmission from the Internet-of-Things (IoT) devices to the cloud/edge works at the semantic level to improve transmission efficiency. The proposed various DeepSC systems can achieve less data transmission to reduce the transmission latency, lower complexity to fit capacity-constrained devices, higher robustness to multi-user interference and channel noise, and better performance to perform various intelligent tasks compared to the conventional communication systems.

Declaration

I, Huiqiang Xie, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature:

Date: April 21, 2023

Acknowledgments

Foremost, I would like to express my deepest gratitude to my primary supervisor Dr. Zhijin Qin for her prudent guidance, continuous support and patience during my PhD journey, and to my co-supervisor Dr. Akram Alomainy for his kind help and continuous support in the final year of my PhD With their professional supervision, I could figure out research problems, deal with challenges, and then contribute a list of publications.

Secondly, I want to thank Prof. Geoffrey Ye Li from Imperial College London for his patient revision of my research paper. I would also like to thank my collaborators, Prof. Khaled B. Letaief from Hong Kong University of Science and Technology and Prof. Biing-Hwang Juang from Georgia Institute of Technology, for their technical support and research feedback. Through our collaboration, I gained special research experience and a thorough understanding of my research works.

I would like to express my thanks to my second supervisor, Dr. James Kelly and my independent accessor Dr. Julian Hough, for their support of my PhD progression viva, and the committee for my viva defense: Prof. Osvaldo Simeone, and Dr. Mona Jaber, for their professional comments and feedback.

I would also like to thank all the colleagues including Dr. Xiaolan Liu, Dr. Jiadong Yu, Mellisa Yeo, Damon Owen, Xiaochuan Fang, Zhisu Qu, Hanchi Ruan, Mojan Omidvar, and all the EECS staff for their insightful comments and kind help.

To all my friends, thank you all for making this journey so much fun and enjoyable. Recall the PhD journey, there are so many treasures in my memory, the adventurous summoner's rift, the Scotland highland, the beautiful sea beach, the charming architectures, and the most important treasure, my friends, Zelin Ji, Zhenzi Weng, Yiyu Guo, Wenhan Shen, Chao Zhang, Anna Li, and Xiaotong Wang. I am so lucky to meet you

all. Thanks for you all to make each party, the lunch, and even the lockdown joyful, so I could stay happy and keep a positive attitude towards my PhD life.

Finally, I would like to thank my family for their support and giving me freedom to be myself, I could rebuild confidence quickly every time when I met challenges and felt worried about my research.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AWGN	Additive White Gaussian Noise
BLEU	Bi-Lingual Evaluation Understudy
BLSTM	Bi-directional Long Short-Term Memory
CSI	Channel State Information
CNN	Convolutional Neural Network
CE	Cross Entropy
DL	Deep Learning
DNN	Deep Neural Networks
E2E	End-to-End
FFN	Feed-Forward Network
GRU	Gated Recurrent Units
GMSC	Generic Model of Semantic Communication
IoT	Internet-of-Things
JPEG	Joint Photographic Experts Group
JSC	Joint Source-Channel
KL	Kullback-Leibler
LS	Least Squares
L-MMSE	Linear Minimum Mean-Squared Error
LSTM	Long Short Term Memory
LDPC	Low-Density Parity-Check

MAP	Maximum A Posteriori
MSE	Mean-Square Error
MIMO	Multiple-Input Multiple-Output
MHSA	Multi-Headed Self Attention
NLP	Natural Language Processing
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
QAT	Quantization-Aware Training
RF	Radio Frequency
RS	Reed-Solomon
SIT	Semantic Information Theory
SNR	Signal-to-Noise Ratio
SISO	Single-Input Single-Output
SGD	Stochastic Gradient Descent
STE	Straight-Through Estimator
VQA	Visual Question Answering
UTF-8	8-bit Unicode Transformation Format

List of Notations

$h, \boldsymbol{h}, \mathbf{H}$	the channel coefficient in scalar, vector, and matrix
$n, \boldsymbol{n}, \mathbf{N}$	the AWGN scalar, vector, and matrix
$s, \boldsymbol{s}, \mathbf{S}$	the source data in scalar, vector, and matrix
$z, \boldsymbol{z}, \mathbf{Z}$	the semantic information in scalar, vector, and matrix
$x, \boldsymbol{x}, \mathbf{X}$	the transmitted signal in scalar, vector, and matrix
$y, \boldsymbol{y}, \mathbf{Y}$	the received signal in scalar, vector, and matrix
$p, \boldsymbol{p}$	the probability in scalar, vector
$\hat{z}$, $\hat{\boldsymbol{z}}$, $\hat{\mathbf{Z}}$	the estimated semantic information in scalar, vector, and matrix
$\hat{x}$, $\hat{\boldsymbol{x}}$, $\hat{\mathbf{X}}$	the estimated transmitted signal in scalar, vector, and matrix
$\hat{h}$, $\hat{\boldsymbol{h}}$, $\hat{\mathbf{H}}$	the estimated channel coefficient in scalar, vector, and matrix
w_l	the l-th word in the sentence
$\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, γ, $\boldsymbol{\chi}$, $\boldsymbol{\theta}$	the learnable parameters
$\mathcal{L}$	the loss function
$\mathcal{D}$	the dataset
$\mathcal{B}$	the mini-batch
P	the transmission power
$\mathbf{W}$, $\boldsymbol{b}$	the weights and bias of neural network
a, $\hat{a}$	the real answer and predicted answer

Chapter 1

Introduction

The wireless communication systems were born to connect and transmit information between two ends, in which the data are collected at the transmitters and reconstructed at the receivers. With the development of wireless communication systems spanning from the first generation to the fifth generation, the achieved transmission rate has been improved tens of thousands of times than before and the system capacity is able to support connecting massive machines. Such evolution caters to various data-hungry applications, e.g., high-resolution video streams, multimodal data transmission, and real-time online games. However, as we step into the era of connected intelligence [2], the widely deployed devices have been generating unprecedented amounts of multimodal data. According to a report published by Ericsson in November of 2021, the monthly global data traffic is predicted to grow exponentially over the next five years [3]. Besides, the appearing various applications, such as artificial intelligence, autonomous cars [4], Internet-of-Things (IoT) [5], virtual/augmented reality [6, 7], and mobile robots [8], need the additional constraints, e.g., the larger bandwidth, higher power, and the lower latency, which makes conventional communications facing a new bottleneck and performance limit.

It's time to shift to the new communication paradigm. Based on Shannon and

Weaver [9], communications could be categorized into three levels:

- The technical problem: How accurately can the symbols of communication be transmitted?
- The semantic problem: How precisely do the transmitted symbols convey the desired meaning?
- The effective problem: How effectively does the received meaning affect conduct in the desired way?

Conventional communications focus on the first level which mainly concerns about the successful transmission of symbols from the transmitter to the receiver, where the transmission accuracy is mainly measured at the level of bits or symbols. Conventional communications are built upon the separate source-channel information theory [10], which faces several limitations to address the upcoming huge data traffic and serve the applications. First, conventional communications mainly consist of several modules over noisy channels, such as source coding, channel coding, modulation, channel estimation, and so on, in which each module is modeled mathematically and optimized individually. While the design of each module has been mature after decades of development, designing these modules separately may lead to error propagation and prevent reaching joint optimality. Second, conventional communications are content-irrelevant, which requires error-free full data reconstruction recovery. The generated unprecedented volume of data inevitably increases the transmission time, which makes conventional communications hard to satisfy latency-sensitive tasks in the future.

The second level of communication deals with the semantic information sent from the transmitter and the meaning interpreted at the receiver, named semantic communication. In semantic communications, the transmitted source can be different from the recovered source but with the same meanings, which can improve robustness and save communication overheads. For example, given the transmitted sentence, “where is the car?”, the semantic communication systems aim to recover a similar meaning sentence, i.e., “where is the automobile?” However, the sentence, “where is the automobile?” is not satisfied with the metrics of conventional communications, and then automatic repeat request is employed to recover the exact same sentence, “where is the car?”, which will cost the additional communication resource and introduce the higher latency.

The third level deals with the effects of communication that turn into the ability of the receiver to perform specific tasks in the way desired by the transmitter, named goal/task-oriented semantic communication. In goal/task-oriented semantic communication, only important, relevant, and useful information to the users/applications is extracted from a large amount of data and delivered to the destinations. Adopting such systems can reduce the transmission time to meet the requirements of latency-sensitive tasks as well as keep the same performance. For example, given the fire alert monitor scenario, the goal/task-oriented semantic communication systems can extract the fire-related information of images captured from a camera at the transmitter and infer whether sounds fire alert at the receiver directly. However, conventional communication systems have to recover the image first and then infer the categories of images, which causes a higher latency.

In summary, different from the conventional communications rooted in the first level, the semantic communications derived from the second level and the goal/task-oriented semantic communications derived from the third level do not require the recovery of accurate data but reconstructing the data having the same meanings or performing the intelligent tasks directly [11], which are promising solutions to the upcoming challenges. However, how to achieve semantic communications is still an open problem. In this thesis, we aim to explore semantic communications and goal/task-oriented semantic communications and give various designs for different scenarios.

1.1 Motivations and Contributions

In this thesis, we will investigate deep learning (DL) based semantic communications. The proposed semantic communication systems have the capable of gathering multimodal data from different users/devices, transmitting over the air, and processing/fusing multimodal data at the receiver, with low complexity to the transceiver. The specific motivations and contributions are summarized in the following.

1.1.1 Point-to-Point Semantic Communication

Historically, the concept of semantic communication was developed several decades ago. The pioneer works provide some insights into the semantic theory and remarks on the initial design of semantic communications. However, many issues remain unexplored, i.e., the measure of meanings, the design of semantic communication, and the effects of wireless channels. These remaining problems make semantic communication still far from satisfactory for practical applications. Therefore, a point-to-point semantic communication system over the physical channels is needed to solve the problems.

In this thesis, we propose a DL based semantic communication system, named DeepSC, for text transmission. Based on the Transformer, the DeepSC aims at maximizing the system capacity and minimizing the semantic errors by recovering the meaning of sentences, rather than bit- or symbol-errors in traditional communications. Moreover, transfer learning is used to ensure that the DeepSC is applicable to different communication environments and to accelerate the model training process. To justify the performance of semantic communications accurately, we also initialize a new metric, named sentence similarity. Compared with the traditional communication system without considering semantic information exchange, the proposed DeepSC is more robust to channel variation and is able to achieve better performance, especially in the low signal-to-noise ratio (SNR) regime, as demonstrated by the extensive simulation results.

1.1.2 Multi-User Semantic Communication

Except for point-to-point semantic communications, the task-oriented multiple users transmission is another challenge, in which different users extract important data at the transmitter to directly serve for different intelligent tasks at the receiver. In conventional communications, each user transmits and restores its data independently. However, in practice, we must gather multimodal data from different users/devices, transmit it over the air, and process/fuse multimodal data at the receiver. It is natural to evolve from single-user semantic communications to multiple users semantic communications.

In this thesis, we investigate DL based multi-user semantic communication systems for transmitting single-modal data and multimodal data, respectively. We will adopt three intelligent tasks, including, image retrieval, machine translation, and visual question answering (VQA) as the transmission goal of semantic communication systems. We will then propose a Transformer based unique framework to unify the structure of transmitters for different tasks. For the single-modal multi-user system, we will propose two Transformer based models, named, DeepSC-IR and DeepSC-MT, to perform image retrieval and machine translation, respectively. In this case, DeepSC-IR is trained to optimize the distance in embedding space between images, and DeepSC-MT is trained to minimize the semantic errors by recovering the semantic meaning of sentences. For the multimodal multi-user system, we develop a Transformer enabled model, named, DeepSC-VQA, for the VQA task by extracting text-image information at the transmitters and fusing it at the receiver. In particular, a novel layer-wise Transformer is designed to help fuse multimodal data by adding connections between each of the encoder and decoder layers. Numerical results will show that the proposed models are superior to traditional communications in terms of the robustness to channels, computational complexity, transmission delay, and task execution performance at various task-specific metrics.

1.1.3 Semantic Communication with Memory

While semantic communication succeeds in recovering data and performing intelligent tasks due to its ability to extract important information, it can only deal with memoryless transmission tasks. Similar to the memory and memoryless channel, memoryless tasks are only related to the current input, e.g., the image classification. The memory tasks are related to both the current input and the previous inputs, e.g., scenario question answering and scenario conversations. Therefore, in order to achieve general semantic communications to support both the memoryless and memory tasks, a memory semantic communication system is needed to be designed.

In this thesis, we investigate the DL based memory semantic communication systems, named Mem-DeepSC, by considering the scenario question answer task. We proposed the universal Transformer based transceiver to extract the semantic information and introduce the memory module to process the context information. To make the Mem-DeepSC applicable to various SNRs, we derive the semantic-aware channel capacity to validate the possibility of dynamic transmission. Specially, we propose two dynamic transmission methods to enhance the reliability of transmission as well as reduce the communication overhead, by masking some unimportant elements, in which the model is trained with mutual information to recognize the unimportant elements. Numerical results show that the proposed model with memory is superior to the benchmarks in terms of answer accuracy and transmission efficiency.

1.1.4 Low-Complexity Semantic Communication

The task-oriented semantic communication can transmit less number of symbols under the narrow band and achieve better performance for the tasks, which is suitable for communication in the IoT scenario. The DL-enabled IoT devices are capable of exploiting and processing different types of data more effectively as well as handling more intelligent tasks than before. Although some IoT devices have certain capabilities to process simple DL models, the limited memory, computing, and battery capability still prevent wide applications of DL [12]. Therefore, a low-complexity DL enabled semantic communication system for lower latency and power consumption at the IoT devices is needed to solve the challenges.

In this thesis, we consider an IoT network where the cloud/edge platform performs the DL based semantic communication model training and updating while IoT devices perform data collection and transmission based on the trained model. To make it affordable for IoT devices, we propose a lite distributed semantic communication system based on DL, named L-DeepSC, for text transmission with low complexity, where the data transmission from the IoT devices to the cloud/edge works at the semantic level to improve transmission efficiency. Particularly, by pruning the model redundancy and lowering the weight resolution, the L-DeepSC becomes affordable for IoT devices and the bandwidth required for model weight transmission between IoT devices and the cloud/edge is reduced significantly. Through analyzing the effects of fading channels in forward-propagation and back-propagation during the training of L-DeepSC, we develop a channel state information (CSI) aided training processing to decrease the effects of fading channels on transmission. Meanwhile, we tailor the semantic constellation to make it implementable on capacity-limited IoT devices. Simulation demonstrates that the proposed L-DeepSC achieves competitive performance compared with traditional methods, especially in the low SNR region. In particular, it can reach as large as a 20x compression ratio without performance degradation.

1.2 Associated Publications

The publications during my Ph.D. study are listed below. Part of [J1] and [C1] are included in Chapter 3. The work in [J3] is discussed in Chapter 4. The work in Chapter 5 has been published as [J2]. Moreover, the work in Chapter 6 has been submitted as [J5].

Journal Paper

1. **H. Xie**, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," in *IEEE Trans. Signal Processing*, vol. 69, pp. 2663-2675, Apr. 2021.

2. **H. Xie** and Z. Qin, "A Lite Distributed Semantic Communication System for Internet of Things," in *IEEE J. Select. Areas in Commun.*, vol. 39, no. 1, pp.

142–153, Jan. 2021.

3. **H. Xie**, Z. Qin, and G. Y. Li, "Task-Oriented Multi-User Semantic Communications for VQA Task," in *IEEE Wireless Commun. Letter*, Dec. 2021.

4. **H. Xie**, Z. Qin, X. Tao, and K. B. Letaief, "Task-Oriented Multi-User Semantic Communications," *in IEEE Journal on Selected Areas in Communications,* Jul. 2022.

5. **H. Xie**, Z. Qin, and G. Y. Li "Semantic Communication with Memory," *Submitted to IEEE Journal on Selected Areas in Communications.*

6. Q. Fu, **H. Xie**, Z. Qin, G. Slabaugh, and X. Tao "Vector Quantised Semantic Communication System", *Submitted to IEEE Wireless Commun. Letter.*

Conference Paper

1. **H. Xie**, Z. Qin, G. Y. Li, and B.-H. Juang "Deep Learning based Semantic Communications: An Initial Investigation," *IEEE Global Telecommunications Conference*, Taiwan China, Dec. 2020.

1.3 Thesis structure

Chapter 2 covers the background of conventional communication and semantic communications. Additionally, various intelligent tasks are introduced.

Chapter 3 investigates the point-to-point semantic communication system. Specifically, a deep learning enabled semantic communication system is proposed. The extensive simulation results demonstrate the effectiveness and robustness of the proposed system.

Chapter 4 proposes the unified multimodal multi-user semantic communications framework to support the text and image tasks. The numerical results verified the effectiveness and robustness of the proposed multi-user semantic communication systems.

Chapter 5 designs the semantic communication system with a memory module to support the memory tasks. Besides, the proposed dynamic transmission methods are detailed. Numerical results are presented to show the performance of the proposed memory semantic communication.

Chapter 6 proposes a distributed semantic communication system for IoT networks, where a lite DeepSC is proposed, called L-DeepSC. The numerical results show the low complexity of the proposed L-DeepSC.

Chapter 7 draws the conclusions of this thesis and potential future research work.

Chapter 2

Background

This chapter provides an overview of the background knowledge used in this thesis, including the concepts of conventional communications and semantic communications, the related literature review, the employed DL models in the thesis, the intelligent tasks, and their responding metrics.

2.1 Conventional Communications

In 1948, Shannon firstly introduced the concept of information entropy, which exploits uncertainty to measure the information in the unit of bits, as well as the concept of separation theorem stated in Theorem 1, which enables to design the source coding and channel coding separately.

Theorem 1. *If $s_1, s_2, \cdots, s_n$ satisfies asymptotic equipartition property (AEP) and $H(S) \leq C$, there exists a source-channel code with $p(\tilde{s}^n \neq s^n) \rightarrow 0$. Conversely, for stationary process, if $H(S) > C$, probability of error is bounded away from 0.*

Based on the Shannon separation theory, the conventional communication system shown in Fig. 2.1 consists of multiple modules. Given the source data, the system firstly converts the source data into the bit streams by source coding, in which Shannon source

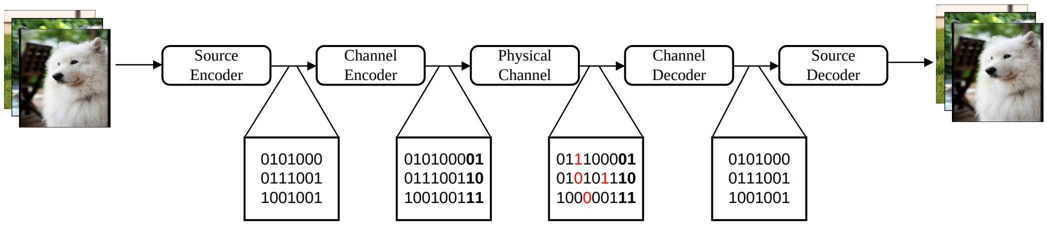


Figure 2.1: The conventional communication systems.

coding theorem gives the limit to compress data, and adds the parity bits by the channel coding, in which Shannon capacity indicates the bound of transmission rate to ensure the bit streams transmitted correctly, which can be expressed mathematically by

$$y = hf_c(f_s(s)) + n, \tag{2.1}$$

where s is the source data, h is the transmission channel coefficient, n is the additive white Gaussian noise (AWGN), $f_s(\cdot)$ is the source encoder, $f_c(\cdot)$ is the channel encoder. Notice that the output of $f_s(\cdot)$ and $f_c(\cdot)$ will be the *bit streams*.

At the receiver, the channel coding can correct the error bits disrupted by the physical channels and the source coding reconvert the bit streams to the original source data, which can be expressed mathematically by

$$\hat{s} = f_s^{-1}(f_c^{-1}(y)), \tag{2.2}$$

where $\hat{s}$ is the recovered data, $f_s^{-1}(\cdot)$ is the source decoder, $f_c^{-1}(\cdot)$ is the channel decoder. The source coding and channel coding follow $s = f_s^{-1}\left(f_s(s)\right)$ and $f_s(s) = f_c^{-1}\left(f_c\left(f_s(s)\right)\right)$, respectively.

In the past 70 years, this system has been the template for most modern communication systems. Massive efforts have been made to optimize each module of source coding and channel coding so that the recovered data can be as much as accurate. For the source coding, Shannon's source coding theorem gives the limit to compress data, it is possible to compress the source data by removing the redundancy in the entropy domain. The source coding can be divided into lossless compression and lossy compression. Lossless compression allows the original data to be perfectly reconstructed from the compressed data. Such codes are also called entropy coding, which analyzes the source distribution to design the code length. The representative codes are Huffman coding [13] and lempel-ziv coding [14] for general compression purposes, 8-bit Unicode transformation format (UTF-8) [15] for text, H.264 lossless [16] for video, tagged image file format (TIFF) lossless [17] for graphics data, and free lossless audio codec (FLAC) [18] for audio, and so on. By contrast, lossy compression permits reconstruction only of an approximation of the original data with greatly improved compression rates, where information loss is inevitable. In general, these information that cannot be observed by humans will be discarded, e.g, the high-frequency information in image and audio. The widely employed lossy codes are joint photographic experts group (JPEG) 2000 [19] for image, H.261 [20] for video, MPEG layer III (MP3) [21] for audio, linear predictive coding [22] for speech, and so on.

For channel coding, Shannon's channel coding theorem establishes that for any given degree of noise communication channels, it is possible to communicate discrete data (digital information) nearly error-free up to a computable maximum rate through the channel. Based on Shannon's noisy-channel coding theorem, it is possible to control errors in data transmission over noisy communication channels by encoding the bits in a redundant way, named error-correction coding (ECC), where the redundancy can be used to detect and correct the error bits. ECC has two main categories, block coding and convolutional codes. Block codes work on fixed-size blocks (packets) of bits or symbols of predetermined size, i.e., Reed-Solomon coding [23], Hamming coding [24], Bose–Chaudhuri–Hocquenghem (BCH) codes [25], Low-density parity-check (LDPC) codes [26], polar codes [27], and so on. Convolutional codes work on bit or symbol streams of arbitrary length, i.e., recursive systematic convolutional (RSC) codes [28], turbo codes [29], and so on.

Recently, there appear many works to optimize the source coding and channel coding separately by using machine learning (ML) techniques. These ML-based approaches can be expressed mathematically by

$$s = f_s^{-1}\left(f_s(s; \boldsymbol{w}_1); \boldsymbol{w}_2\right), \tag{2.3}$$

$$f_s(s) = f_c^{-1}\left(f_c\left(f_s(s); \boldsymbol{w}_3\right); \boldsymbol{w}_4\right), \tag{2.4}$$

in which $f_s(\cdot; \boldsymbol{w}_1)$ and $f_s^{-1}(\cdot; \boldsymbol{w}_2)$ are the source encoder and decoder with the learnable parameters $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$; $f_c(\cdot; \boldsymbol{w}_3)$ and $f_c^{-1}(\cdot; \boldsymbol{w}_4)$ are the channel encoder and decoder with the learnable parameters $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$. With introducing the learnable parameters, different from the traditional coding, these ML-based coding methods can automatically discover the latent structure behind the source data, rather than manually engineered, by updating the learnable parameters.

For ML-based source coding, Bottou *et al.* [30] introduced the DjVu format for document image compression, which employs techniques such as segmentation and K-means clustering to separate foreground from background, and analyzes the document's contents. More recently, with the introduction of the DL, the main idea to achieve image compression is to employ the autoencoder framework to learn the latent features and map into bit streams[31–33]. In [34], Toderici *et al.* firstly proposed the recurrent neural network (RNN) for binary feature extraction, which demonstrates the power of DL to compress the data. Rippel *et al.* [35] proposed real-time image compression by using the convolutional neural network (CNN) based autoencoder structure, in which the pyramidal decomposition analyzes individual scales and adversarial training pursues realistic image reconstructions.

For the ML based channel coding, most works focus on improving the channel encoding and decoding with the DL techniques [36–41], in which the channel coding modules are replaced by the neural network and trained by the designed loss function. In [36], Gruber *et al.* proposed the DL-based polar decoding method, in which the neural networks replace the traditional polar decoding and learn how to decode the structured codes. Similarly, Wu *et al.* [37] employed the neural networks to design the LDPC decodings. Jiang *et al.* [38] employed the autoencoder to re-design Turbo codes and Kim *et al.* [39] introduced the feedback codes by using the RNN.

All of the developed methods have fostered the connected intelligent society and have been applied to the on-demand mobile applications [2, 42]. For more and more latency-sensitive applications, it requires both high transmission efficiency to perform the applications and a shorter code length to satisfy the latency requirement. However, these separate optimization only hold on when the coding length is infinite, which incurs a larger latency. Besides, the main metrics for the separate-based methods are the bit-error ratio (BER) or symbol-error ratio (SER), which cannot directly measure users' quality of service. Meanwhile, these developed separate methods ignore semantics underlying the source data and the intentions behind the users, which cannot provide the desired services effectively for the users. Therefore, it needs to transfer to the new communication paradigm.

2.2 Semantic Communications

The concept of semantic communication was developed by engineers and philosophers several decades ago. As early as in 1925, Dewey [43] stated that "communication must be considered as a means to an end" and Wittgenstein [44] said that "brought to the forefront of philosophy." In 1949, Weaver [9] brought the semantics into the engineering problem and identified the three levels of communication. With the idea of semantic communications, Carnap and Bar-Hillel [45] attempted to outline the "Theory of Semantic Communication" in 1952. After that, the research of semantic communication proceeded in fits and starts. Until recently, semantic communications have been attracting more and more attention as the potential to solve high-level problems, such that the exchange of information is achieved most efficiently.

As shown in Fig. 2.2, different from conventional communications, semantic commu-

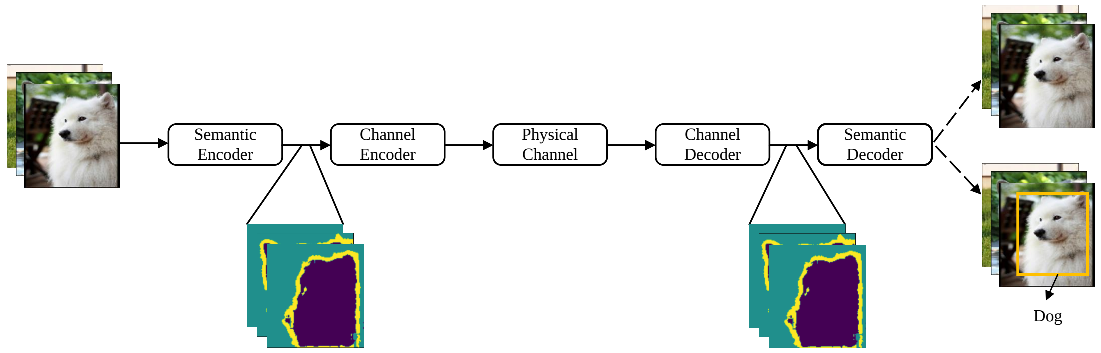


Figure 2.2: The semantic communication systems.

nications introduce semantic coding to find the semantics underlying the source data. In detail, semantic communications extract the semantic features related to the data or tasks by the semantic encoder and prevent the effects from physical channels by the channel encoder. At the receiver, the distorted semantic features are recovered by the channel decoder and employed to reconstruct the original data or perform different intelligent tasks directly by the semantic decoder. In semantic communications, only those semantic features will be transmitted, which reduces the required communication resources significantly to meet the requirements of latency-sensitive tasks as well as keep the same performance. Besides, the source data are not processed at the bit level, but at the semantic level. Therefore, the metrics for the semantic communications will no longer be the BER or SER but be the ones that can measure the performance of applications directly. The main difference between joint source-channel coding (JSCC) and semantic communication is semantic processing. The JSCC is a type of semantic-agnostic approach of conventional wireless communication systems. Most existing JSCC solutions combine conventional source and channel code designs, and jointly optimize their parameters for improved end-to-end performance. On the other hand, semantic communication is the type of semantic-relevant approach, which seeks to ensure that only the relevant information for the underlying task is communicated to the receiver.

Generally, as shown in Fig. 2.3, we can apply the semantic coding before the JSCC to compress the source at the semantic domain, then apply the JSCC to further com-

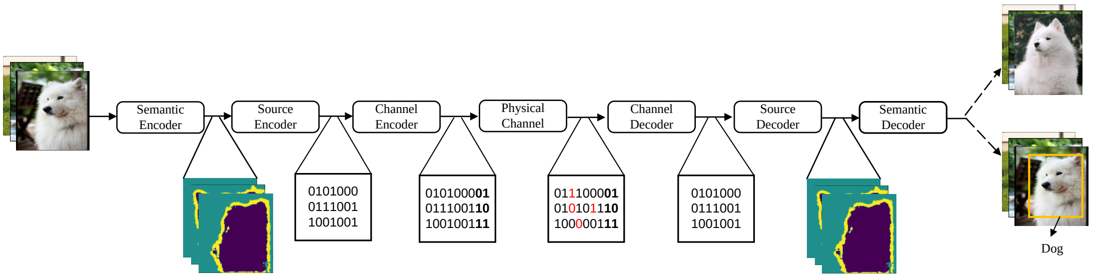


Figure 2.3: The semantic-JSCC communication systems.

press the semantic information at the entropy domain, which can further improve the transmission rate.

There are mainly four types of semantic communications: 1) logical probabilities-based semantic communications, in which the semantics are explained by the logical probabilities and truth values; 2) knowledge-based semantic communications, using structured knowledge to represent semantics; 3) DL-based semantic communications, which employs the DL model parameters to learn semantics; 4) goal-oriented semantic communications, in which the importance of information is semantics.

2.2.1 The Logical-Probability-based Semantic Communications

Inspired by Shannon and Weaver [9], Carnap *et al.* [45] were the first to introduce the semantic information theory (SIT) based on logical probabilities ranging over the contents. To elaborate, any sentence can either be logically true, logically false, or logically indeterminate. For example, for sentences i and j, we have i logically implies j defined to mean that "if i then j" is logically true. With the logical relations, the definition of semantic entropy is derived in Definition 1.

Definition 1. Let $m(x)$ denote the logical probability of a message x, the semantic entropy of x is defined by

$$H_s(x) = -\log_2\left(m(x)\right). \tag{2.5}$$

However, the logical contradiction was ignored in the [45], e.g., the contradiction relation, "*i* and not *i*," which will give the maximum semantic information instead of the minimum semantic information. In order to solve the problem, Floridi *et al.* [46] outlined theory of strongly semantic information (TSSI), which characterized the semantic information with the truth lies in the paradox, i.e., the degrees of vacuity and inaccuracy. Simon *et al.* [47] built the foundations for both the SIT and TSSI of quantifying the semantic information, in which two approaches, named Tichie-Oddie approach and Niiniluoto approach, have been proposed. Afterward, Bao *et al.* [48] have proposed a generic model of semantic communication (GMSC) as an extension of the SIT, where the concepts of the semantic channel were first defined (shown in Definition 2). The defined semantic channel capacity provides the bound of transmitting the information with arbitrary semantic errors.

Definition 2. For every discrete memoryless channel, the channel capacity can be given by

$$C_s = \sup_{P(X|W)} \left\{ I\left(X;Y\right) - H\left(W\left|X\right.\right) + \mathbb{E}\left[H_s\left(Y\right)\right] \right\}, \tag{2.6}$$

in which X, Y, and W are transmitted signals, received signals, and the set of interpretations, respectively, $I\left(X;Y\right)$ is the mutual information between X and Y, $H(W\left|X\right.)$ is the conditional entropy of $P(W\left|X\right.)$, and $H_s\left(Y\right)$ is the semantic entropy of Y.

In [49], a lossless semantic data compression theory by applying the GMSC was developed, which means that data can be compressed at the semantic level so that the size of the data to be transmitted can be reduced significantly. These prior works have more concerned with the semantic information theory and semantic communication theory, however, there are no applications of the proposed theory to confirm this potential.

2.2.2 Knowledge-based Semantic Communications

In psychology, the semantics reflect on how humans solve problems and represent knowledge in order to design formalisms that will make complex systems easier to design and build [50]. Followed by the idea, the research on knowledge representation have been begun as early as 1959, Simon *et al.* [51] developed the general problem solver system.

With the development in knowledge representation, the knowledge graph has been one of the prevalent methods of representing knowledge. A simple example of a knowledge graph is shown in Fig. 2.4. The graph is modeled as "*entity-relations-entity*", in which entities and relations are represented as the nodes and edges, respectively.

Jeong *et al.* [52] were the first to employ the knowledge graph in semantic communications, in which two knowledge graph-based methods, domain dictionary and ontology dictionary, were proposed to correct the semantic and lexical errors for speeches. The main idea behind [52] is to replace these syntactic or semantic errors using a semantic confusion table firstly, then the lexical errors are corrected by the domain or ontology dictionaries. Guler *et al.* [53] have proposed an end-to-end (E2E) semantic communication framework that integrates the semantic inference and physical layer communication problems, where the transceiver is optimized by minimizing the average semantic errors. This error is derived by the similarity between two word knowledge graphs proposed in [54]. Designing such transceiver is an NP hard problem until introducing the third agent to provide intentions of communication agents, it reached a sequential equilibrium. It is shown that, when sufficient information is available regarding the intentions of users involved in the communication, efficient semantic communication can be achieved. Wang *et al.* [55] have proposed the semantic communication framework, in which the semantic of text is defined as the knowledge graph. In more detail, the transmitter models the text data into the knowledge graph and the receiver recovers the text with the received knowledge graph, such that the recovered text is not the same as the original text but has similar meanings. Zhou *et al.* [56] have proposed a cognitive semantic communication framework, which is similar to the framework proposed in [55]. Different from [55] that only consider the resource allocations, the cognitive framework takes the physical channels into consideration and the semantic error can be corrected via inference driven by knowledge graph-based pre-trained model. Wang *et al.* [57] have proposed the semantic image reconstruction from scene graphs, in which the image is described by the text at the transmitter to reduce the communication overheads and the receiver can restore the

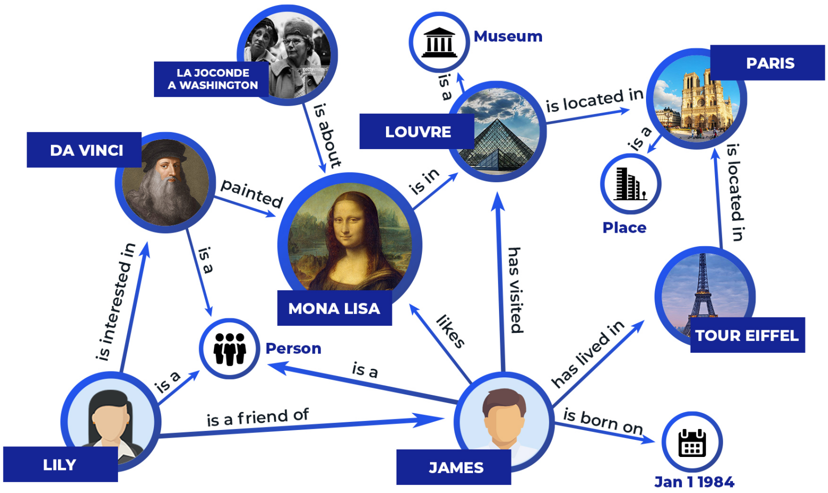


Figure 2.4: An example of knowledge graph [1].

scene graph directly according to the received text.

Knowledge graph is a natural way to represent knowledge in a system, which can then be used to facilitate semantic communication. However, this approach comes with several challenges. First, as knowledge in the system grows, working with knowledge graph becomes difficult since the latent knowledge graph becomes massive. Scalability for other modal data is also a challenge. The mainstream is built from text-to-graph-to-text. The applications of other modal data are still unexplored.

2.2.3 DL-based Semantic Communications

With the development of DL in the recent decade, computer vision and nature language processing (NLP) driven by deep neural networks have reached nearly or the same performance as people judgment [58, 59]. The early successful work was proposed by Krizhevsky *et al.* [60] in 2012, which employs the CNN to largely improve the performance of image classification in terms of top-1 and top-5 test error. To construct the deeper neural networks, He *et al.* [61] in 2016 proposed the residual connections to alleviate gradient explosion such that training the deep neural networks with more than 100 layers. Later one year, Vaswani *et al.* [62] proposed the Transformer network for NLP. It has shown that its ability to translate certain languages is the same as that of humans. These indicate that the machine has the initial ability to understand the meanings behind the data. Regarding the DL based semantic communications, a DL based general framework is shown in Fig. 2.5. The transceiver is replaced by DNNs to learn the meanings behind source data as well as merge the blocks in conventional communication systems to achieve the global optimum.

The DL-based semantic communications often depend on the modality of communication (text, images, speech, etc.). Farsad *et al.* [63] have designed the initial deep joint source-channel coding for text transmission, in which the text sentences are encoded into fixed-length bit streams over simple channel environments. With the depth exploration in semantic communications, Xie *et al.* [64] have developed more powerful joint semantic-channel coding, named DeepSC, to encode text information into various lengths over complex channels. Moreover, Xie *et al.* [65] also have proposed an environment-friendly semantic communication system, named L-DeepSC, for capacity-limited devices. Peng *et al.* [66] have designed robust semantic communication systems to prevent the semantic delivery from the source noise, e.g., typos and syntax errors. Except for the end-to-end semantic communication systems, Jiang *et al.* [67] have exploited hybrid automatic repeat request to reduce the semantic transmission error further for sentence transmission, where the system is the hybrid of semantic coding and traditional channel coding. Zhang *et al.* [68] have proposed the semantic-based Huffman coding to reduce the number of bits and the semantic decoder to correct the semantic errors.

Bourtsoulatze *et al.* [69] have investigated the initial deep image transmission semantic communication systems, in which the semantic and channel coding are optimized jointly. Kurka *et al.* [70] extended Bourtsoulatze's work with the channel feedback to improve the quality of image reconstruction. The authors [71] also have explored the initial relationship between the length of transmitted signals and the different bandwidths in the deep image transmission systems, which can achieve the initial adaptive transmission. Yang *et al.* [72] have combined the deep image transmission systems with the orthogonal frequency division multiplexing (OFDM) transmission so that transmitting

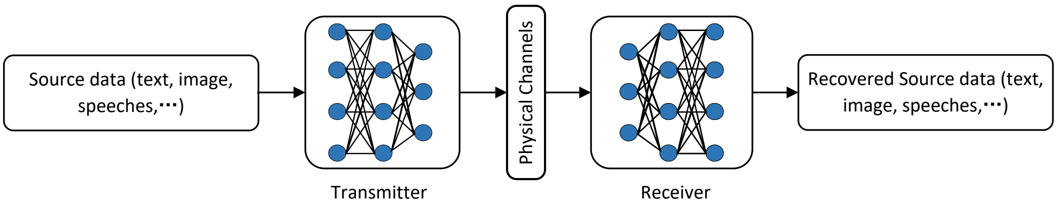

Source data (text, image, speeches,⋯)
Transmitter
Physical Channels
Receiver
Recovered Source data (text, image, speeches,⋯)

Figure 2.5: A general DL based semantic communication framework.

the image information effectively over the multipath fading channels. Besides, there exist several studies focusing on semantic coding. Huang *et al.* [73] have designed the image semantic coding method by introducing the framework of rate-distortion, which can save the number of bits as well as keep the good quality of the reconstructed image.

Weng *et al.* [74] have developed the initial deep speech semantic communication systems, named DeepSC-S, by employing an attention mechanism to extract the semantic features at the transmitter and reconstructing speech signals at the receiver. Tong *et al.* [75] have proposed a federal learning-based approach to further improve the accuracy of recovered speech signals at the receiver. Han *et al.* [76] have designed efficient speech semantic communication systems by introducing the connectionist temporal classification alignment module to identify the auxiliary to help reconstruct speech signals.

Tung *et al.* [77] have designed the initial deep video semantic communications by accounting for occlusion/disocclusion and camera movements. Especially, the authors considered the DL-based frame design for the video reconstruction. Wang *et al.* [78] have proposed the adaptive deep video semantic communication systems by learning to allocate the limited channel bandwidth within and among video frames to maximize the overall transmission performance. Jiang *et al.* [79] have investigated the application of semantic communications in the video conference, in which the proposed system can maintain high resolution by transmitting some keypoints to represent motions and keep the low communication overheads. Similarly, Tandon *et al.* [80] also considered the video conference transmission. Different from [79], the authors have designed the video semantic communication by converting the video to text at the transmitter and recovering the video from the text at the receiver.

DL is an efficient way to learn semantics. Many of the discussed works provide quantitative results demonstrating efficient semantic communication. However, the DL-based approach comes with an interpretation problem due to the black-box nature of DL models. In addition, the data-driven model needs large data to be fed such that improves intelligence.

2.2.4 Goal-Oriented Semantic Communications

The goal-oriented semantic communications take the importance of information as semantics. Such communication systems generally have the transmission goal, e.g., specific intelligent tasks (image classification, machine translation, ...), robot controlling, and so on. Given such transmission goals, the system can filter the information related to goals and discard the unimportant information. Regarding goal-oriented semantic communications, it can be divided into DL-based systems and signal sampling-based systems.

The DL-based goal-oriented semantic communication systems are similar to the DL-based semantic communication systems mentioned in section 2.2.3, which mainly consist of deep neural networks. The difference is that goal-oriented semantic communication systems directly perform intelligent tasks instead of data reconstruction. Xie *et al.* [81] have designed the task-oriented semantic communication systems by considering machine translation as the transmission task, in which the transmitter sends the text in one language and the receiver receives the text in another language but keeps the same meanings. Sana *et al.* [82] have designed the new loss function to achieve the balance between mutual information and semantic errors. Weng *et al.* [83] designed speech recognition-oriented semantic communications, named, DeepSC-SR, to directly recognize the speech signals into texts. Han *et al.* [84] have designed the more energy-efficient speech-to-text systems by introducing the redundancy removal module to reduce the transmission data. Lee *et al.* [85] developed an image classification-oriented semantic communications for improving the recognition accuracy rather than performing image reconstruction and classification separately. Hu *et al.* [86] have proposed robust semantic communication systems for the image classification by training the model against the source image noise. Jankowski *et al.* [87] considered image based re-identification for persons or cars as the communication task, in which two schemes (digital and analog) are proposed to improve the retrieval accuracy. Similarly, Xie *et al.* [81] have proposed the vision Transformer based image semantic communication systems for image retrieval, in which only the global semantic feature is transmitted to further adopt the narrow bandwidth. Kang *et al.* [88] have taken the scene classification into consideration, in which the drones capture the scene images, extract the image semantic features, and transmit them to the edge server for scene classification.

A multi-user semantic communication system was proposed to support multimodal data transmission. The MU-DeepSC is for serving the VQA task to improve the answer accuracy [89], which adopts Long Short Term Memory (LSTM) for the text transmitter and CNN for the image transmitter. Zhang *et al.* [90] have proposed the unified model, named U-DeepSC, to serve various transmission tasks with the sharing weights, where the text reconstruction, image reconstruction, and the VQA tasks are considered.

The signal sampling-based semantic communications aim to sample the more valuable information for the applications. The idea can be tracked as early as 1956, Yen [91] proposed the nonuniform sampling for bandwidth-limited signals, in which the more significant values of signals are sampled instead of sampling the uniformly-spaced values. Thereafter, a lot of works were proposed [92–96], which reveals the importance of significant values in signal sampling. Inspired by this idea, Kountouris *et al.* [97] have proposed the concept of the value of information, in which the more significant and useful values to the task are sampled. It is shown that, considering the transmission goal over the wireless communication channels, the performance of the goal can be improved largely in terms of real-time reconstruction error and cost of actuation error. Uysal *et al.* [98] have proposed a similar idea “a redesign of the entire process of information generation, transmission, and usage in unison.” The entire system is optimized by the freshness, relevance, and value of information.

The two types of goal-oriented semantic communications are facing several challenges, i.e., the interpretation and training data requirements for the DL-based systems and the realization problems for sampling-based systems.

We aim to build such semantic communication systems for the various types of sources and serve intelligent tasks. Compared to the four types of semantic communications, DL-based semantic communications can support various modal data and different intelligent tasks, which follow the aim. In general, the DL-based perform the semantic coding locally. However, if the DL-based semantic communication systems do not have enough computational capacity, it needs to offload the semantic processing to the edge but with the cost of more communication rounds and the risk of privacy leakage.

2.3 Transformer Models and Loss Functions

In this section, we will introduce the basic Transformer model and its variants for the Transformer model. Besides, we also elaborate on the classic loss functions.

2.3.1 Transformer Models

The Basic Transformer Model

The basic Transformer model [62] was proposed for the natural language processing firstly, which consists of the encoder blocks and decoder blocks. Each encoder block includes two main modules: 1) a Multi-Headed Self Attention (MHSA) network; and 2) a Feed-Forward network (FFN). The MHSA applies a self-attention operation to different projections of input tokens, which can learn what are the important tokens in the input tokens. The MHSA employs the softmax activation to perform the self-attention operation, which is expressed as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.7}$$

where the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the projections of the input data $\boldsymbol{s}$. Then, the FFN applies two dense layers to the weighted projections. This consists of two linear transformations with a ReLU activation in between, which is

$$\mathrm{FFN}\left(\mathbf{O}\right) = f_2\left(\mathbf{W}_2 f_1\left(\mathbf{W}_1\mathbf{O} + \boldsymbol{b}_1\right) + \boldsymbol{b}_2\right) \tag{2.8}$$

where $\mathbf{O}$ is the output of MHSA.

Similarly, the decoder block includes three main modules: 1) a Multi-Headed Self Attention network; 2) a Multi-Headed Guided Attention network, which applies an attention operation to the projections of input tokens and the output tokens of the encoder; and 3) a Feed-Forward network. All modules are preceded by layer normalization and followed by a skip connection.

Universal Transformer Model

Although Transformer shows its success on some sequence modeling tasks such as machine translation, it fails to generalize some tasks that RNN can handle with. Dehghani *et al.* [99] proposed the universal Transformer by combining the characteristic of RNN and the Transformer, the model of which is shown in Fig. 2.7.

The universal Transformer consists of the encoder and decoder. The encoder consists of only one Transformer encoder block, in which a recurrent mechanism connects the outputs of encoder block to the inputs of encoder block. After T steps, the outputs of encoder will be inputted into the decoder. Similarly, the decoder includes only one Transformer decoder block and performs forward-propagation recurrently. After T steps, the decoder will give the outputs.

Vision Transformer Model

Transformer was proposed for text-related tasks initially. However, the researcher found its potential in image-related tasks. Dosovitskiy *et al.* [100] have applied the Transformer model in image classification, in which the vision Transformer was proposed.

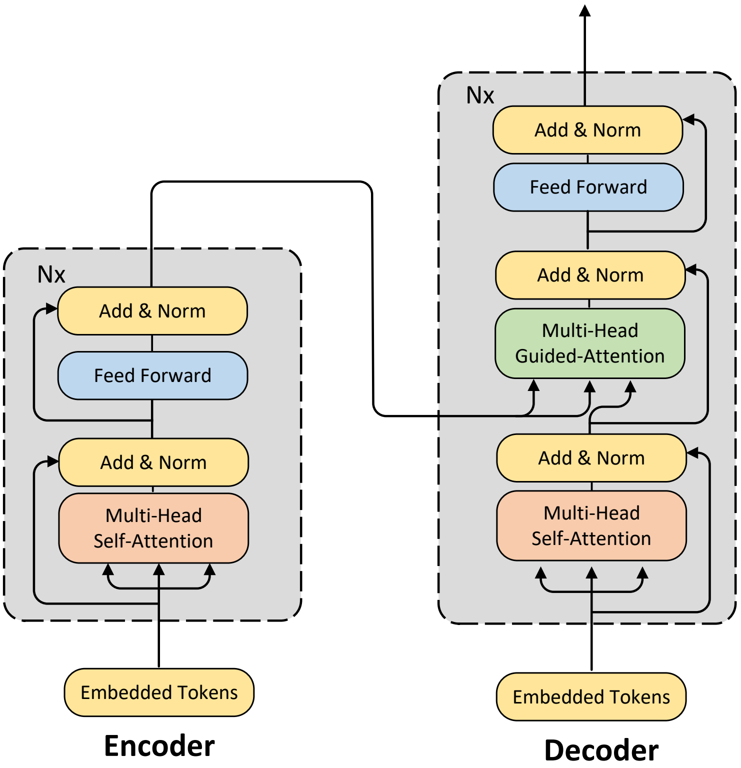


Figure 2.6: The structure of Transformer.

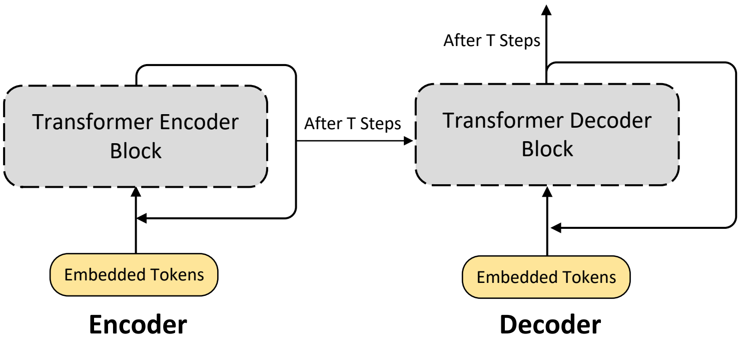


Figure 2.7: The universal Transformer.

The vision Transformer is shown in Fig. 2.8. The vision Transformer employs the same structure as the Transformer encoder. The main challenge is how to map an image to several tokens. In the vision Transformer, the input image is first decomposed into fixed-sized patches, e.g., 16×16. Each patch is linearly projected into vector-shaped tokens and used as an input to the Transformer. An extra learnable <CLS> token is added to the input sequence such that its corresponding output token serves as a global representation of the input sequence.

2.3.2 Loss Functions

The loss function is the learning objective to guide the DNN. In general, intelligent tasks can be categorized into three parts, regression tasks, classification tasks, and knowledge representation. The typical loss functions are MSE loss function, CE loss function, and contrastive loss function

Mean-Square Error Loss

The regression task is to predict the contiguous values, e.g., images, speeches, and prices. The MSE loss function can measure the differences between two contiguous values as

$$\mathcal{L}_{\text{MSE}} = \|\boldsymbol{o} - \boldsymbol{o}_r\|^2, \tag{2.9}$$

where $\boldsymbol{o}$ and $\boldsymbol{o}_r$ are the predicted values and real valus, respectively. After minimizing the $\mathcal{L}_{\text{MSE}}$, the differences between $\boldsymbol{o}$ and $\boldsymbol{o}_r$ decrease, which means that the DNN learns to create the desired outputs.

Cross-Entropy Loss

The classification task is to predict the discrete values, e.g., the image category, and the answer category. The CE loss function can be computed by

$$\mathcal{L}_{\text{CE}} = -p_r(\boldsymbol{o}) \log p(\boldsymbol{o}), \tag{2.10}$$

where $p(\boldsymbol{o})$ and $p_r(\boldsymbol{o})$ are the predicted and the real probability, respectively. By minimizing $\mathcal{L}_{\text{CE}}$, the DNN can learn the real probability distribution, thereby increasing the probability of predicting the correct category.

Contrastive Loss

The knowledge representation employs the DNNs to learn the latent semantic vectors.

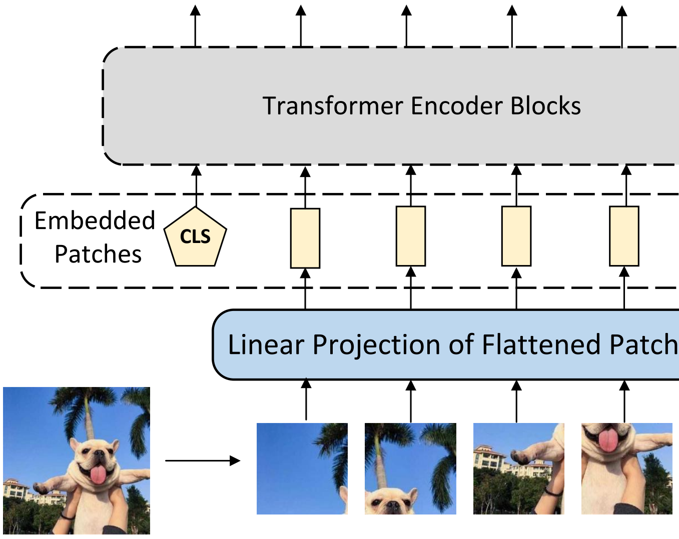

Transformer Encoder Blocks
Embedded Patches
CLS
Linear Projection of Flattened Patches

Figure 2.8: The vision Transformer.

The contrastive loss function [101] is expressed by

$$\mathcal{L}_{\text{Contrastive}} = \mathbb{E}\left[\sum_{l_i=l_j} \left(1 - \boldsymbol{o}_i^{\mathrm{T}} \boldsymbol{o}_j\right)\right] + \mathbb{E}\left[\sum_{l_i \neq l_j} \left(\boldsymbol{o}_i^{\mathrm{T}} \boldsymbol{o}_j - \xi\right)_+\right], \tag{2.11}$$

where $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ are the features of i-th and j-th datum, respectively, and l_i and l_j are the corresponding labels. The operator $(x)_+$ returns $\max(x, 0)$. The goal of contrastive loss is to discriminate the source that belongs to different categories.

2.4 Intelligent Tasks and its Metrics

In this section, four different intelligent tasks chosen in this thesis are detailed. Its corresponding metrics are also introduced.

2.4.1 Image Retrieval

Task Descriptions

The image retrieval task aims to identify the top-k similar images by matching the query images with those stored in a remote server and returning the similar ones to users sending the query.. For example, the user uploads a dress image to the Amazon app and wishes to find similar dress products. Such image retrieval tasks cannot be performed locally due to the centralized database at the server. For the image retrieval task, the transmitter sends the query image to the receiver, and then the receiver identifies and returns the top-k images to users by reliable channels.

Modern methods for image retrieval typically rely on DL-based models by extracting compact image-level features [102] for image matching or classification. Recent techniques mainly focus on two parts: deep network architectures and training algorithms. The deep network architectures include single forward pass models [103], multiple forward pass models [104], attention-based models [105], and deep hashing embedding based models [106]. The training algorithms focus on classification based learning [107], metric based learning [108], and unsupervised-based learning [109].

Metrics

The general metric to measure image retrieval is the recall, which is the fraction of the relevant items that are successfully retrieved. Assume the system gets a top-K recommended list of items, the recall@k can be computed by

$$recall@k = \frac{\text{Relevant Items Recommended in top-k}}{\text{Total Relevant Items}}. \tag{2.12}$$

The output of recall@k is a number between 0 and 1, which indicates how the ability of the system to recommend correct items. Here is an example to understand Recall@K. Assume we are providing 5 recommendations in this order — 1 0 1 0 1, where 1 represents relevant and 0 irrelevant. The total relevant item is 3. The recall@k would be, recall@3 is 2/3, recall@4 is 2/3, and recall@5 is 3/3.

2.4.2 Machine Translation

Task Descriptions

Machine translation task aims to translate text or speech from one language to another, e.g., from Chinese to English. One core of communication is to transmit the meanings behind the text, and one of the major obstructs to communication is the different grammar and presentations for different languages. Therefore, for the machine translation task, the intention is that the transmitter sends one language, e.g., Chinese, and the receiver directly receives the desired language, e.g., English, which aims to break the obstruct of communications and improve communication efficiency.

The recent successful approaches for machine translation problems are mostly based on the classic encoder-decoder structure [110], in which the encoder extracts the sentence-level intermediate features at the source language and the decoder provides the entire sentence at the target language based on the intermediate features. The representative models include CNN-based models [111], Transformer based models [62], and RNN-based models, e.g., LSTM networks [112] and Gated Recurrent Units (GRU) networks [113].

Metrics

BLEU Score is a general metric to measure the performance of machine translation. Through counting the difference of n-grams between transmitted and received texts, where n-grams means the size of a word group. For example, for the sentence "weather is good today", 1-gram: "weather", "is", "good" and "today", 2-grams: "weather is", "is good" and "good today". The same rule applies to the rest.

For the transmitted sentence $\boldsymbol{s}$ with length $l_{\boldsymbol{s}}$ and the decoded sentence $\hat{s}$ with length $l_{\hat{s}}$, the BLEU can be expressed as

$$\log \text{BLEU} = \min\left(1 - \frac{l_{\hat{s}}}{l_{\boldsymbol{s}}}, 0\right) + \sum_{n=1}^{N} w_n \log p_n, \tag{2.13}$$

where w_n is the weights of n-grams and p_n is the n-grams score by computing the word frequency.

The output of BLEU is a number between 0 and 1, which indicates how similar the decoded text is to the transmitted text, with 1 representing the highest similarity. However, few human translations will attain the score of 1 since word errors may not make the meaning of a sentence different. For instance, the two sentences, “my car was parked there” and “my automobile was parked there”, have the same meaning but with different BLEU scores since they use different words. To characterize such a feature, we propose a new metric, sentence similarity, at the sentence level in addition to the BLEU score.

2.4.3 Visual Question Answering

Task Descriptions

Given an image and a natural language question about the image, the VQA task is to provide an accurate natural language answer. Vision information and text information are the representative type of data used in daily life, which can be adopted for many IoT scenarios, e.g., environment monitor, autonomous retail, and intelligent assistant. Take the fire alarm scenario as an example, the camera sends the scenario vision information and the temperature sensor transmits the scenario temperature information. Then, the transmitted vision and temperature information from different users is employed to carry out a scenario fire alarm monitor at the remote receiver. For the VQA task, one transmitter sends an image and another one sends a question related to the image. The receiver directly gets the answer based on the received image and question.

The core of VQA tasks is multimodal data fusion techniques [114], in which the image and questions in text are first represented as global features and then fused by a multimodal fusion model to predict the answer. Recent approaches adopt the visual attention mechanism by attending image features with given question features, which include multimodal bilinear pooling methods [115], stacked attention network [116], bottom-up and top-down attention mechanism [117], and co-attention network [118].

Metrics

The metric for the VQA is the answer accuracy, which is the fraction of the number of correctly predicted answers and the total number of answers. The answer accuracy is computed by

$$Answer\ Accuracy = \frac{\text{The Number of Correct Answers}}{\text{The Number of Total Answers}}. \tag{2.14}$$

The answer accuracy is between 0 and 1, indicating how accurately the system answers the questions.

2.4.4 Scenario Question Answering

Task Descriptions

Given the scenario information, the scenario QA task aims to provide an accurate natural language answer for the text question about the scenario. This task can be applied to the scenario monitor, intelligent conversations, and so on. In this context, the transmitter sends the scenario text information over multiple time slots. The receiver directly gets the answer based on the received scenario information.

Unlike these memoryless tasks related to only the current input, the scenario QA task belongs to the memory tasks related to both the current input and the previous inputs. Therefore, the core of the scenario QA task is the processing of scenario information. Recent approaches focus on the processing of memory module, which includes different variants, e.g., dynamic memory networks [119] and end-to-end memory neural networks [120].

Metrics

The metric for the scenario QA task is the same as the VQA task. The answer accuracy is employed here to measure the performance of the scenario QA task.

2.5 Summary

This chapter presents the fundamental concepts of conventional communications, as well as semantic communication systems. Additionally, the concept of Transformer is demonstrated in this chapter. Finally, various intelligent tasks are introduced in this chapter.

Chapter 3

Point-to-Point Semantic Communications

In this chapter, the main contributions are reviewed in Section 3.1. The framework of a semantic communication system is presented and a corresponding problem is formulated in Section 3.2. Section 3.3 details the proposed DeepSC and extends it to dynamic environments. Numerical results are presented in Section 3.4 to show the performance of the DeepSC. Finally, Section 3.5 concludes this chapter.

3.1 Introduction

Recent advancements on DL based NLP and communication systems inspire us to investigate semantic communication to realize the second level communications as aforementioned [58, 59, 121–124]. The considered semantic communication system mainly focuses on the joint semantic-channel coding and decoding, which aims to extract and encode the semantic information of sentences rather than simply a sequence of bits or a word. For the semantic communication system, we face the following questions:

Question 1: How to define the meaning behind the bits?

Question 2: How to measure the semantic error of sentences?

Question 3: How to jointly design the semantic and channel coding?

The main contributions of this chapter are summarized as follows:

- Based on the Transformer [62], a novel framework for the DeepSC is proposed, which can effectively extract the semantic information from texts with robustness to noise. In the proposed DeepSC, a joint semantic-channel coding is designed to cope with channel noise and semantic distortion, which addresses aforementioned *Question 3*.
- The transceiver of the DeepSC is composed of semantic encoder, channel encoder, channel decoder, and semantic decoder. To understand the semantic meaning as well as maximize the system capacity at the same time, the receiver is optimized with two loss functions: cross-entropy and mutual information. Moreover, a new metric is proposed to accurately reflect the performance of the DeepSC at the semantic level. These address the aforementioned *Questions 1* and *2*.
- Based on extensive simulation results, the proposed DeepSC outperforms the traditional communication system and improves the system robustness at the low SNR regime.

3.2 System Model and Problem Formulation

The considered system model consists of two levels: semantic level and transmission level, as shown in Fig. 3.1. The semantic level addresses semantic information processing for encoding and decoding to extract the semantic information. The transmission level guarantees that semantic information can be exchanged correctly over the transmission medium. Overall, we consider an intelligent E2E communication system with the stochastic physical channel, where the transmitter and the receiver have certain background knowledge, i.e., different training data. The background knowledge could be various for different application scenarios.

Definition 3. Semantic noise is a type of disturbance in the exchange of a message that interferes with the interpretation of the message due to ambiguity in words, a sentence or symbols used in the message transmission.

Definition 4. Physical channel noise is caused by the physical channel impairment, such as, additive white Gaussian noise (AWGN), fading channel, and multiple path, which incurs the signal attenuation and distortion.

3.2.1 Problem Description

As in Fig. 3.1, the transmitter maps a sentence, $\boldsymbol{s}$, into a complex symbol stream, $\boldsymbol{x}$, and then passes it through the physical channel with transmission impairments, such as distortion and noise. The received, $\boldsymbol{y}$, is decoded at the receiver to estimate the original sentence, $\boldsymbol{s}$. We jointly design the transmitter and receiver with DNNs since DL enables us to train a model with inputting variable-length sentences and different languages.

Particularly, we assume that the input of the DeepSC is a sentence, $\boldsymbol{s} = [w_1, w_2, \cdots, w_L]$, where w_l represents the l-th word in the sentence. As shown in Fig. 3.1, the transmitter consists of two parts, named semantic encoder and channel encoder, to extract the semantic information from $\boldsymbol{s}$ and guarantee successful transmission of semantic information over the physical channel. The semantic information can be extracted by

$$\boldsymbol{z} = S\left(\boldsymbol{s}; \boldsymbol{\alpha}\right), \tag{3.1}$$

where $\boldsymbol{z}$ is the semantic information and $S\left(\cdot; \boldsymbol{\alpha}\right)$ is the semantic encoder network with the parameter set $\boldsymbol{\alpha}$.

Then, the transmitted symbol stream can be represented by

$$\boldsymbol{x} = C\left(\boldsymbol{z}; \boldsymbol{\beta}\right), \tag{3.2}$$

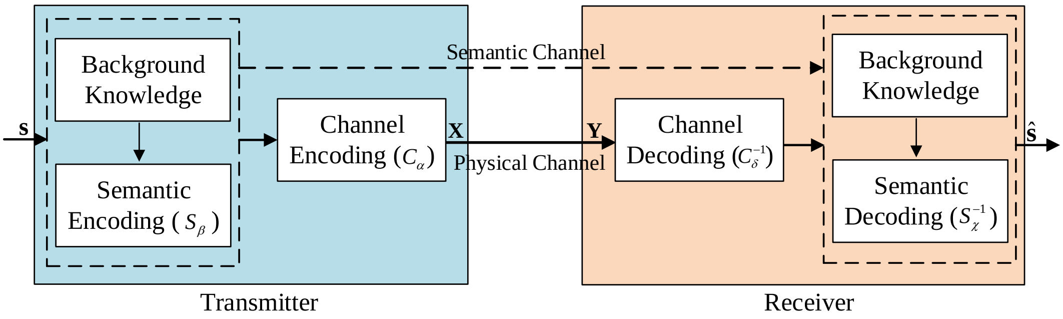

Semantic Channel
Background Knowledge
Semantic Encoding (S_β)
Channel Encoding (C_α)
s
X
Physical Channel
Y
Channel Decoding (C_δ^{-1})
Background Knowledge
Semantic Decoding (S_χ^{-1})
ŝ
Transmitter
Receiver

Figure 3.1: The framework of proposed DL enabled semantic communication system, DeepSC.

where $\boldsymbol{x}$ is the encoded signal, and $C(\cdot;\boldsymbol{\beta})$ is the joint source-channel (JSC) encoder with the parameter set $\boldsymbol{\beta}$. The JSC encoder can compress semantic information and deal with the effects from physical channels.

If $\boldsymbol{x}$ is sent, the signal received at the receiver will be

$$\boldsymbol{y} = h\boldsymbol{x} + \boldsymbol{n}, \tag{3.3}$$

where $\boldsymbol{y}$ is the received signal, h represents the Rayleigh fading channel with $\mathcal{CN}(0,1)$ and $\boldsymbol{n} \sim \mathcal{CN}\left(0, \sigma_n^2\right)$. For E2E training of the encoder and the decoder, the channel must allow back-propagation. Physical channels can be formulated by neural networks. For example, simple neural networks could be used to model the AWGN channel, multiplicative Gaussian noise channel, and the erasure channel [125]. While for the fading channels, more complicated neural networks are required [126]. In this chapter, we mainly consider the AWGN channel and Rayleigh fading channel for simplicity while focus on semantic coding and decoding.

As shown in Fig. 3.1, the receiver includes channel decoder and semantic decoder to recover the transmitted symbols and then transmitted sentences, respectively. Before recover the semantic information, the least squares (LS) signal detection is applied to alleviate the effects from physical channels, which is given by

$$\hat{\boldsymbol{x}} = \frac{\hat{h}^{\mathtt{H}}\boldsymbol{y}}{\|\hat{h}\|^2}, \tag{3.4}$$

where $\hat{\boldsymbol{x}}$ is the estimated transmitted symbols and $\hat{h}$ is the estimated CSI.

Then, the recovered semantic information can be expressed as

$$\hat{\boldsymbol{z}} = C^{-1}\left(\hat{\boldsymbol{x}}; \boldsymbol{\gamma}\right) \tag{3.5}$$

where $C^{-1}\left(\cdot; \boldsymbol{\gamma}\right)$ is the JSC decoder with the parameter set $\boldsymbol{\gamma}$ to decompress semantic information and eliminate the distortion from channels.

Finally, the recovered source signal can be represented as

$$\hat{\boldsymbol{s}} = S^{-1}\left(\hat{\boldsymbol{z}}; \boldsymbol{\varphi}\right), \tag{3.6}$$

where the $\hat{\boldsymbol{s}}$ is the recovered sentence and $S^{-1}\left(\cdot; \boldsymbol{\varphi}\right)$ is the semantic decoder network with the parameter set $\boldsymbol{\varphi}$.

The goal of the system is to minimize the semantic errors while reducing the number of symbols to be transmitted. However, we face two challenges in the considered system. The first challenge is how to design joint semantic-channel coding. The other one is semantic transmission, which has not been considered in the traditional communication system. Even if the existing communication system can achieve a low BER, several bits, distorted by the noise and beyond error correction capability, could lead to understanding difficulty as the partial semantic information of the whole sentence might be missed. In order to achieve successful recovery at semantic level, we design semantic and channel coding jointly in order to keep the meaning between $\hat{\boldsymbol{s}}$ and $\boldsymbol{s}$ unchanged, which is enabled by a new DNN framework. The cross-entropy (CE) is used as the loss function to measure the difference between $\boldsymbol{s}$ and $\hat{s}$, which can be formulated as

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{s}, \hat{\boldsymbol{s}}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\chi}, \boldsymbol{\delta}) = -\sum_{l=1} q\left(w_l\right) \log\left(p\left(w_l\right)\right) + \left(1 - q\left(w_l\right)\right) \log\left(1 - p\left(w_l\right)\right), \qquad (3.7)$$

where $q(w_l)$ is the real probability that the l-th word, w_l, appears in estimated sentence $\boldsymbol{s}$, and $p(w_l)$ is the predicted probability that the i-th word, w_i, appears in sentence $\hat{\boldsymbol{s}}$. The CE can measure the difference between two probability distributions. Through reducing the loss value of CE, the network can learn the word distribution, $q(w_l)$, in the source sentence, $\boldsymbol{s}$, which indicates that the syntax, phrase, the meaning of words in context can be learnt by the network. Besides, jointly designing and training semantic-channel coding can make the whole network learning the knowledge for the specific goal. In other words, the channel coding can pay more attention in protecting the semantic information related to transmission goal while neglecting other irrelevant information. Separately designing will make channel coding addressing all information equally.

3.2.2 Channel Encoder and Decoder Design

One important goal on designing a communication system is to maximize the capacity or the data transmission rate. Compared with BER, the mutual information can provide extra information to train a receiver. The mutual information of the transmitted symbols, $\boldsymbol{x}$, and the received symbols, $\boldsymbol{y}$, can be computed by

$$I\left(\boldsymbol{x}; \boldsymbol{y}\right) = \int \mathcal{X} \times \mathcal{Y} p\left(x, y\right) \log \frac{p\left(x, y\right)}{p\left(x\right) p\left(y\right)} dx dy = \mathbb{E}_{p(x,y)}\left[\log \frac{p\left(x, y\right)}{p\left(y\right) p\left(x\right)}\right], \qquad (3.8)$$

where $(\boldsymbol{x}, \boldsymbol{y})$ is a pair of random variables with values over the space $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the spaces for $\boldsymbol{x}$ and $\boldsymbol{y}$. $p(x)$ and $p(y)$ are the marginal probability of sending $\boldsymbol{x}$ and received $\boldsymbol{y}$, respectively, and $p(x, y)$ is the joint probability of $\boldsymbol{x}$ and $\boldsymbol{y}$. The mutual information is equivalent to the Kullback-Leibler (KL) divergence between the marginal probabilities and the joint probability, which is given by

$$I\left(\boldsymbol{x};\boldsymbol{y}\right)=D_{\mathrm{KL}}\left(p\left(x,y\right)\|p\left(x\right)p\left(y\right)\right). \tag{3.9}$$

From [127], we have the following theorem,

Theorem 2. The KL divergence admits the following dual representation

$$D_{\mathrm{KL}}\left(P\|Q\right)=\sup_{T:\Omega\to R}E_P\left[T\right]-\log\left(E_Q\left[e^T\right]\right), \tag{3.10}$$

where the supremum is taken over all functions T such that the two expectations are finite.

According to Theorem 2, the KL divergence can also be represented as

$$D_{\mathrm{KL}}\left(p\left(x,y\right)\|p\left(x\right)p\left(y\right)\right)\geqslant\mathbb{E}_{p(x,y)}\left[T\right]-\log\left(\mathbb{E}_{p(x)p(y)}\left[e^T\right]\right). \tag{3.11}$$

Thus, the lower bound of $I\left(\boldsymbol{x};y\right)$ can be obtained from (3.9) and (3.11). In order to find a tight bound on the $I\left(\boldsymbol{x};y\right)$, an unsupervised method is used to train the network T, where T can be approximated by neural network. Meanwhile, the expectation in (3.11) can be computed by sampling, which converges to the true value as the number of samples increases. Then, we can optimize the encoder by maximizing the mutual information defined in (3.11) and the related loss function can be given by

$$\mathcal{L}_{\mathrm{MI}}(\boldsymbol{x},y;T)=\mathbb{E}_{p(x,y)}\left[f_T\right]-\log\left(\mathbb{E}_{p(x)p(y)}\left[e^{f_T}\right]\right), \tag{3.12}$$

where f_T is composed by a neural network, in which the inputs are samples from $p(x,y)$, $p(x)$, and $p(y)$. In our proposed design, $\boldsymbol{x}$ is generated by the function $C_{\boldsymbol{\alpha}}$ and $S_{\boldsymbol{\beta}}$, thus the loss function can be represented by $\mathcal{L}_{\mathrm{MI}}(\boldsymbol{x},y;T,\boldsymbol{\alpha},\boldsymbol{\beta})$ with

$$\mathcal{L}_{\mathrm{MI}}(\boldsymbol{x},y;T,\boldsymbol{\alpha},\boldsymbol{\beta})\leqslant I(\boldsymbol{x};y). \tag{3.13}$$

From (3.13), the loss function can be used to train neural networks to get $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and

T. For example, the mutual information can be estimated by training network T when the encoders $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are fixed. Similarly, the encoder can be optimized by training $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ when the mutual information is obtained.

3.2.3 Performance Metrics

Performance criteria are important to the system design. In the E2E communication system, the BER is usually taken as the training target by the transmitter and receiver, which sometimes neglects the other aspect goals of communication. For text transmission, BER cannot reflect performance well. Except from human judgement to establish the similarity between sentences, bilingual evaluation understudy (BLEU) score is usually used to measure the results in machine translation [128], which will be used as one of the performance metrics in this section. However, the BLEU score can only compare the difference between words in two sentences rather than their semantic information. Therefore, we initialize a new metric, named sentence similarity, to describe the similarity level of two sentences in terms of their semantic information, which is introduced in the following. This provides a solution to *Question 2*.

3.2.3.1 Sentence Similarity

A word can take different meanings in different contexts. For instance, the meanings of mouse in biology and machine are different. The traditional method, such as *word2vec* [129], cannot recognise the polysemy, of which the problem is how to use an numerical vector to express the word while the numerical vector varies in different contexts. According to the semantic similarity, we propose to calculate the sentence similarity between the original sentence, $\boldsymbol{s}$, and the recovered sentence, $\hat{\boldsymbol{s}}$, as

$$\operatorname{match}\left(\hat{\boldsymbol{s}}, \boldsymbol{s}\right) = \frac{\boldsymbol{B}_{\boldsymbol{\Phi}}\left(\boldsymbol{s}\right) \cdot \boldsymbol{B}_{\boldsymbol{\Phi}}(\hat{\boldsymbol{s}})^{T}}{\left\|\boldsymbol{B}_{\boldsymbol{\Phi}}\left(\boldsymbol{s}\right)\right\| \left\|\boldsymbol{B}_{\boldsymbol{\Phi}}\left(\hat{\boldsymbol{s}}\right)\right\|}, \tag{3.14}$$

where $\boldsymbol{B}_{\boldsymbol{\Phi}}$, representing BERT [130], is a huge pre-trained model including billions of parameters used for extracting the semantic information. The sentence similarity defined

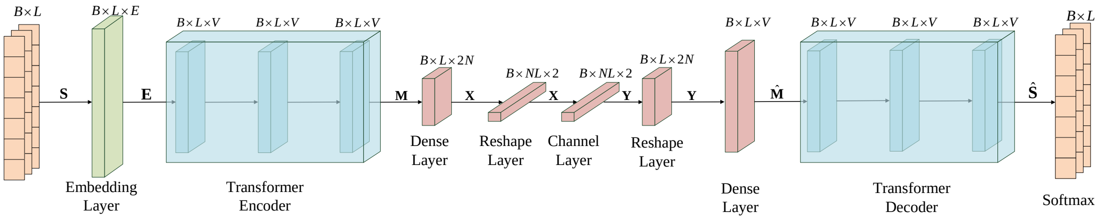


Figure 3.2: The proposed neural network structure for the semantic communication system.

in (3.14) is a number between 0 and 1, which indicates how similar the decoded sentence is to the transmitted sentence, with 1 representing highest similarity and 0 representing no similarity between $\boldsymbol{s}$ and $\hat{s}$.

Compared with BLEU score, BERT has been fed by billions of sentences. Therefore, it has already learnt the semantic information from these sentences and can generate different semantic vectors in different contexts effectively. With the BERT, the semantic information behind a transmitted sentence, $\boldsymbol{s}$, can be expressed as $\boldsymbol{c}$. Meanwhile, the semantic information conveyed by the estimated sentence is expressed as $\hat{\boldsymbol{c}}$. For $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$, we can compute the sentence similarity by $\text{match}(\boldsymbol{c}, \hat{\boldsymbol{c}})$.

3.3 Proposed Deep Semantic Communication Systems

In this section, we propose a DNN for the considered semantic communication system, named as DeepSC, of which the Transformer is adopted for text understanding. Then, transfer learning is adopted to make the DeepSC applicable to different background knowledge and dynamic communication environments. This provides the solutions to *Question 1,3*.

3.3.1 Basic Model

The proposed DeepSC is as shown in Fig 3.2. Particularly, the transmitter consists of a semantic encoder to extract the semantic features from the texts to be transmitted and a channel encoder to generate symbols to facilitate the transmission subsequently. The semantic encoder includes multiple Transformer encoder layers and the channel encoder uses dense layers with different units. The AWGN channel is interpreted as one layer in the model. Accordingly, the DeepSC receiver is composited with a channel decoder for symbol detection and a semantic decoder for text estimation, the channel decoder includes dense layers with different units and the semantic decoder includes multiple Transformer decoder layers. The loss function can be expressed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(\boldsymbol{s}, \hat{\boldsymbol{s}}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\chi}, \boldsymbol{\delta}) - \lambda \mathcal{L}_{\text{MI}}(\boldsymbol{x}, y; T, \boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{3.15}$$

where the first term is the loss function considering the sentence similarity, which aims to minimize the semantic difference between $\boldsymbol{s}$ and $\hat{s}$ by training the whole system. The second one is the loss function for mutual information, which maximize the achieved data rate during the transmitter training. The parameter λ $(0 \leq \lambda \leq 1)$ is the weight for the second term.

The core of Transformer is the multi-head self-attention mechanism, which enables the Transformer to view the previous predicted word in the sequence, thereby better predicting the next word. Fig. 3.3 gives an example of the self-attention mechanism for the word ‘it’. From Fig. 3.3, attention attend to a distant dependency of the pronoun, ‘it’, completing pronoun reference “the animal”, which demonstrates that the self-attention mechanism can learn the semantic and therefore solve aforementioned *Question 1*.

Algorithm 3.1: *DeepSC network training algorithm.*

```
Initialization: Initial the weights $\mathbf{W}$ and bias $\boldsymbol{b}$.
1: Input: The background knowledge set $\mathcal{K}$.
2: Create the index to words and words to index, and then embedding words.
3: while Stop criterion is not met do
4:     Train the mutual information estimated model.
5:     Train the whole network.
6: end while
7: Output: The whole network $S_{\boldsymbol{\beta}}(\cdot), C_{\boldsymbol{\alpha}}(\cdot), C_{\boldsymbol{\delta}}^{-1}(\cdot), S_{\boldsymbol{\chi}}^{-1}(\cdot)$.
```

As shown in Algorithm 3.1, the training process of the DeepSC consists of two phases due to different loss functions. After initializing the weights, $\mathbf{W}$, bias, $\boldsymbol{b}$, and using embedding vector to represent the input words, the first phase is to train the mutual

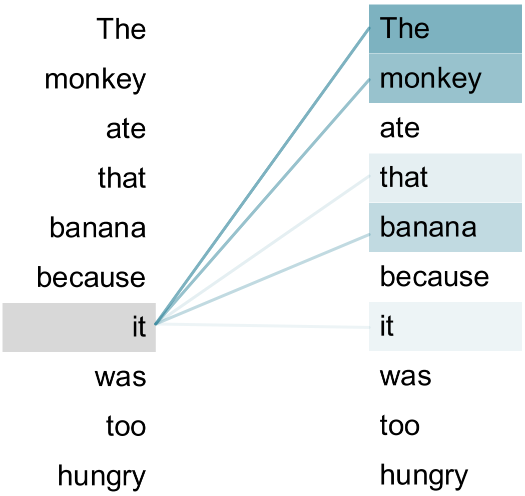


Figure 3.3: An example of the self-attention mechanism following long-distance dependency in the Transformer encoder.

information model by unsupervised learning to estimate the achieved data rate for the second phase. The second phase is to train the whole system with (3.15) as the loss function. Each phase aims to minimize the loss by gradient descent with mini-batch until the stop criterion is met, the max number of iteration is reached, or none of terms in the loss function is decreased any more. Different from performing semantic coding and channel coding separately, where the channel encoder/decoder will deal with the digital bits rather than the semantic information, the joint semantic-channel coding can preserve semantic information when compressing data, which provides the detailed solution for aforementioned *Question 3*. The two training phases are described in the following:

3.3.1.1 Training of mutual information estimation model

The mutual information estimation model training process is illustrated in Fig. 3.4 and the pseudocode is given in Algorithm 3.2. First, the knowledge set $\mathcal{K}$ generates a minibatch of sentences $\mathbf{S} \in \Re^{B\times L\times 1}$, where B is the batch size, L is the length of sentences. Through the embedding layer, the sentences can be represented as a dense word vector $\mathbf{E} \in \Re^{B\times L\times E}$, where E is the dimension of the word vector. Then, pass the semantic encoder layer to obtain $\mathbf{M} \in \Re^{B\times L\times V}$, the semantic information conveyed by

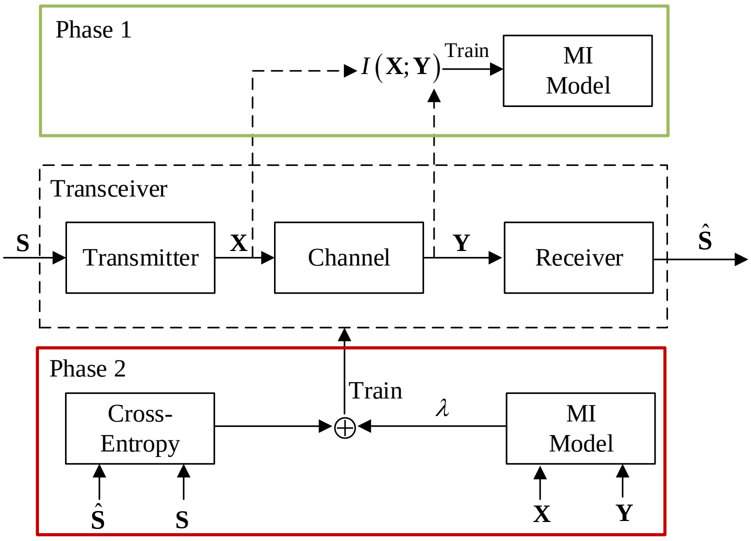


Figure 3.4: The training framework of the DeepSC: phase 1 trains the mutual information estimation model; phase 2 trains the whole network based on the cross-entropy and mutual information.

$\mathbf{S}$, where V is the dimension of Transformer encoder's output. Then, $\mathbf{M}$ is encoded into symbols $\mathbf{X}$ to cope with the effects from the physical channel, where $\mathbf{X} \in \Re^{B\times NL\times 2}$. After passing through the channel, the receiver obtains signal $\mathbf{Y}$ distorted by the channel noise. Based on (3.10), the loss, $\mathcal{L}_{\text{MI}}(\mathbf{X}, \mathbf{Y}; T, \boldsymbol{\alpha}, \boldsymbol{\beta})$, can be computed based on the transmitted symbols, $\mathbf{X}$, and the received symbols, $\mathbf{Y}$, under the AWGN channels. Finally, according to computed $\mathcal{L}_{\text{MI}}$, the stochastic gradient descent (SGD) is exploited to optimize the weights and bias of $f_T(\cdot)$.

Algorithm 3.2: *Train mutual information estimation model.*

```
Input: The knowledge set $\mathcal{K}$.
Transmitter:
    BatchSource($\mathcal{K}$) $\rightarrow \mathbf{S}$.
    $S_{\boldsymbol{\beta}}(\mathbf{S}) \rightarrow \mathbf{M}$.
    $C_{\boldsymbol{\alpha}}(\mathbf{M}) \rightarrow \mathbf{X}$.
    Transmit $\mathbf{X}$ over the channel.
Receiver:
    Receive $\mathbf{Y}$.
    Compute loss $\mathcal{L}_{\text{MI}}$ by (3.10).
    Train $T \rightarrow$ Gradient descent $(T, \mathcal{L}_{\text{MI}})$.
Output: The mutual information estimated model $f_T(\cdot)$.
```

3.3.1.2 Whole network training

The whole network training process is illustrated in Algorithm 3.3. First, minibatch $\mathbf{S}$ from knowledge $\mathcal{K}$ is encoded into $\mathbf{Z}$ at the semantic level, then $\mathbf{Z}$ is encoded into symbol $\mathbf{X}$ for transmission over the physical channels. At the receiver, distorted symbols $\mathbf{Y}$ are received and then decoded by the channel decoder layer, where $\hat{\mathbf{Z}} \in \Re^{B\times L\times V}$ is the recovered semantic information of the sources. Afterwards, the transmitted sentences are estimated by the semantic decoder layer. Finally, the whole network is optimized by the SGD, where the loss is computed by (3.15).

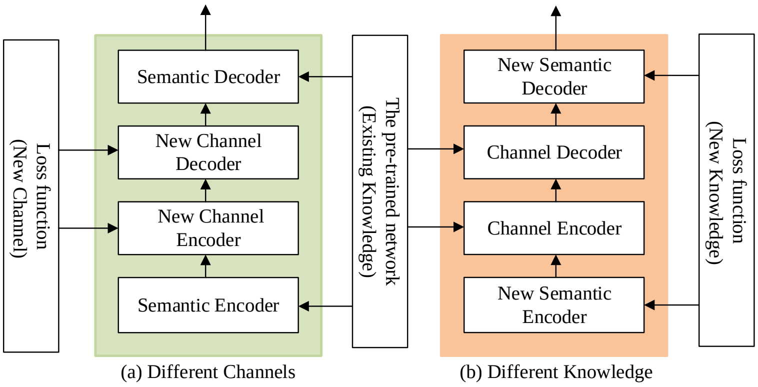
Algorithm 3.3: *Train the whole network.*

```
Input: The knowledge set $\mathcal{K}$.
Transmitter:
    BatchSource($\mathcal{K}$) $\rightarrow \mathbf{S}$.
    $S_{\boldsymbol{\beta}}(\mathbf{S}) \rightarrow \mathbf{Z}$.
    $C_{\boldsymbol{\alpha}}(\mathbf{M}) \rightarrow \mathbf{X}$.
    Transmit $\mathbf{X}$ over the channel.
Receiver:
    Receive $\mathbf{Y}$.
    $C_{\boldsymbol{\delta}}^{-1}(\mathbf{Y}) \rightarrow \hat{\mathbf{Z}}$.
    $S_{\boldsymbol{\chi}}^{-1}(\hat{\mathbf{M}}) \rightarrow \hat{\mathbf{S}}$.
    Compute loss function $\mathcal{L}_{\text{total}}$ by (3.15).
    Train $\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\delta}, \boldsymbol{\chi} \rightarrow$ Gradient descent $(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\delta}, \boldsymbol{\chi}, \mathcal{L}_{\text{total}})$.
Output: The whole network $S_{\boldsymbol{\beta}}(\cdot), C_{\boldsymbol{\alpha}}(\cdot), C_{\boldsymbol{\delta}}^{-1}(\cdot), S_{\boldsymbol{\chi}}^{-1}(\cdot)$.
```

3.3.2 Transfer Learning for Dynamic Environment

In practice, different communication scenarios result in the different channels and the training data. However, the re-training of transmitter and receiver to meet the requirements of dynamic scenarios introduces extra costs. To address this, a deep transfer learning approach is adopted, which focuses on storing knowledge gained while solving a problem and applying it to a different but related problem.

The training process of adopting transfer learning is illustrated in Fig. 3.5 and the pseudocode is given in Algorithm 3.4, where the training modules, mutual information estimation model training, and whole network training, are the same as Algorithm 3.2 and Algorithm 3.3. First, load the pre-trained transmitter and receiver based on knowl-



(a) Different Channels (b) Different Knowledge

Figure 3.5: Transfer learning based training framework: (a) re-train channel encoder and decoder for different channels; (b) re-train semantic encoder and decoder for different background knowledge.

Algorithm 3.4: *Transfer learning based training for dynamic environment.*

```
Initialization: Load the pre-trained model $S_{\boldsymbol{\beta}}(\cdot), C_{\boldsymbol{\alpha}}(\cdot)$,
$C_{\boldsymbol{\delta}}^{-1}(\cdot), S_{\boldsymbol{\chi}}^{-1}(\cdot)$.
Optimal algorithm Training for different background knowledge
Input: The different background knowledge set $\mathcal{K}_1$ .
Freeze $C_{\boldsymbol{\alpha}}(\cdot)$ and $C_{\boldsymbol{\delta}}^{-1}(\cdot)$.
Redesign and train part of $S_{\boldsymbol{\beta}}(\cdot)$ and $S_{\boldsymbol{\chi}}^{-1}(\cdot)$.
while Stop criterion is not met do
    Train the mutual information estimated model.
    Train the whole network.
end while
Output: The adopted whole network.
Optimal algorithm Training for different channel conditions
Input: The background knowledge set $\mathcal{K}$ with the different channel parameters.
Freeze $S_{\boldsymbol{\beta}}(\cdot)$ and $S_{\boldsymbol{\chi}}^{-1}(\cdot)$.
Redesign and re-train part of $C_{\boldsymbol{\alpha}}(\cdot)$ and $C_{\boldsymbol{\delta}}^{-1}(\cdot)$.
while Stop criterion is not met do
    Train the mutual information estimated model.
    Train the whole network.
end while
Output: The re-trained network.
```

edge $\mathcal{K}_0$ and channel $\mathcal{N}_0$. For applications with different background knowledge, we only need to redesign and train part of the semantic encoder and decoder layers and freeze the channel encoder and decoder layers. For different communication environments, we redesign and train part of the channel encoder and decoder layers and freeze the semantic encoder and decoder layers. If the knowledge and channel are totally different, the pre-trained transceiver can also reduce the time consumption because the weights of some layers in the pre-trained model can be reused in the new model even if the most layers need to redesign. After the other modules are trained, we will unfreeze them and train the whole network with few epochs to converge to the global optimum.

3.4 Numerical Results

In this section, we compare the proposed DeepSC with other DNN algorithms and the traditional source coding and channel coding approaches under the AWGN channels and Rayleigh fading channels, where we assume perfect CSI for all schemes. The transfer learning aided DeepSC is also verified under the erase channel and fading channel as well as different background knowledge.

3.4.1 Simulation Settings

The adopted dataset is the proceedings of the European Parliament [131], which consists of around 2.0 million sentences and 53 million words. The dataset is pre-processed into lengths of sentences with 4 to 30 words and is split into training data and testing data.

In the experiment, we set three Transformer encoder and decoder layer with 8 heads and the channel encoder and decoder are set as dense with 16 units and 128 units, respectively. For the mutual information estimation model, we set two dense layers with 256 units and one dense layer with 1 unit to mimic the function T in (3.10), where 256 units can extract full information and 1 unit can integrate information. These settings can be found in Table 3-A. For the baselines, we adopt joint source-channel coding based on neural network and the typical methods for separate source and channel codings.

- DNN based joint source-channel coding [125]: The network consists of Bi-directional Long Short-Term Memory (BLSTM) layers. We label it as JSCC [125] in the simulation figures.

- Traditional methods: To perform the source and channel coding separately, we use

Table 3-A: The setting of the developed semantic network.

	Layer Name	Units	Activation
Transmitter (Encoder)	3×Transformer Encoder	128 (8 heads)	Linear
	Dense	256	Relu
	Dense	16	Relu
Channel	AWGN	None	None
Receiver (Decoder)	Dense	256	Relu
	Dense	128	Relu
	3×Transformer Decoder	128 (8 heads)	Linear
	Prediction Layer	Dictionary Size	Softmax
MI Model	Dense	256	Relu
	Dense	256	Relu
	Dense	1	Relu

the following technologies respectively:

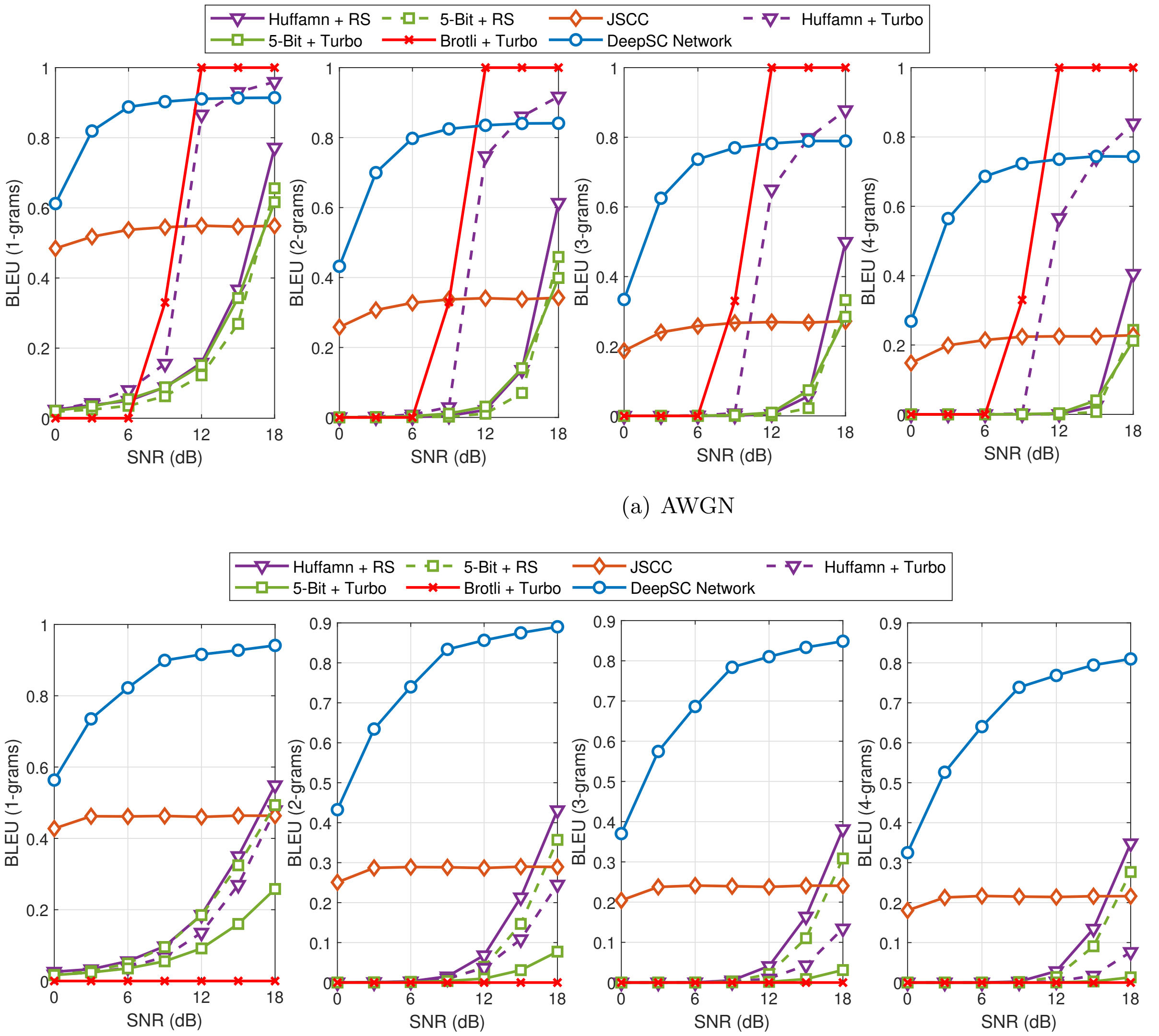
- Source coding: Huffman coding, fixed-length coding (5-bit), and Brotli coding, where Brotli coding uses 2nd context model to compress the context information and every 128 sentences are compressed together in the simulation.

- Channel coding: Turbo coding [132] and Reed-Solomon (RS) coding [133]. We adopt turbo decoding method is log-MAP algorithm with 5 iterations.

The BLEU and sentence similarity are used to measure the performance. The simulation is performed by the computer with Intel Core i7-9700 CPU@3.00GHz and NVIDIA GeForce GTX 2060.

3.4.2 Basic Model

Fig. 3.6 shows the relationship between the BLEU score and the SNR under the same number of transmitted symbols over AWGN and Rayleigh fading channels, where the traditional approaches use 8-QAM, 64-QAM, and 128-QAM for the modulation. Among the traditional baselines in Fig. 3.6(a), Brotli coding outperforms the Huffman and fixed-length encoding over AWGN channels when the turbo coding is adopted for channel coding. The traditional approaches perform better than the DNN based method when the SNR is above 12 dB since the distortion from channel is decreased, where the Brotli



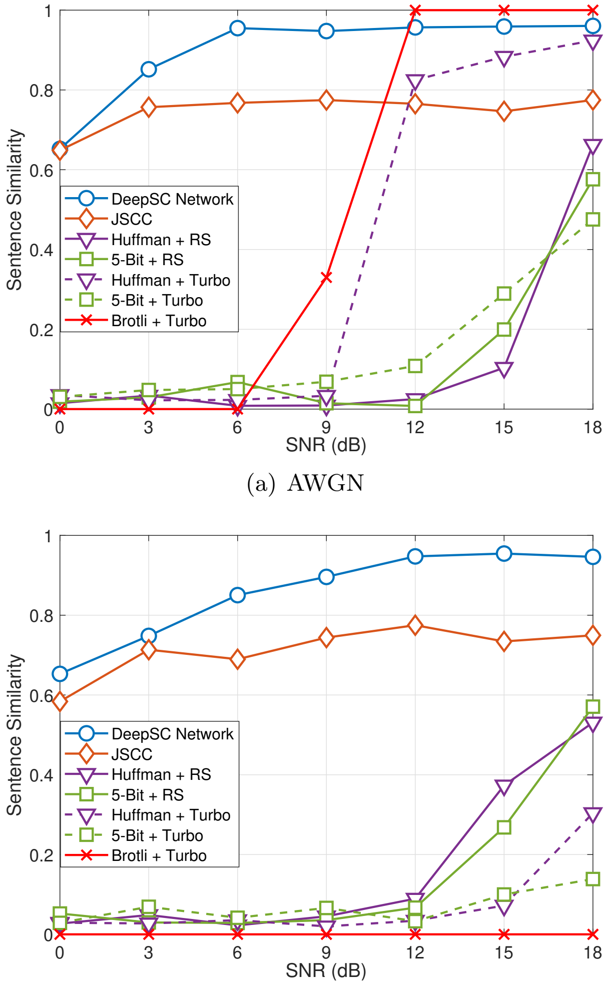
(a) AWGN

(b) Rayleigh Fading

Figure 3.6: BLEU score versus SNR for the similar total number of transmitted symbols over the AWGN channels and Rayleigh fading channels.

with turbo coding performs better than the DeepSC. We observe that all DL enabled approaches are more competitive in the low SNR regime.

In Fig. 3.6(b), the DL enabled approaches outperform all traditional approaches over the Rayleigh fading channels, where RS coding is better than turbo coding in terms of 2-grams to 4-grams. This is because RS coding is linear block coding with long block-length, and can correct long series of bits, however, turbo coding is a type of convolu-



(a) AWGN

(b) Rayleigh Fading

Figure 3.7: Sentence similarity versus SNR for the similar total number of transmitted symbols over the AWGN channels and Rayleigh fading channels.

tional coding with short block-length, so that the adjacent words have higher error rate. DeepSC is not only suitable for short block-length but also performs better in decoding adjacent words, i.e., 4-grams. Note that the BLEU score of the method with Brotil coding and turbo coding is always 0 over Rayleigh fading channels. This is because that 128 sentences are compressed together, while Brotil decoding requires error-free codes after channel decoding for the codes corresponding to the 128 sentences. However, it is almost to guarantee the error-free transmission over Rayleigh fading channels. Therefore, we fail to restore any of the 128 sentences compressed together in Brotil coding as shown in Fig. 3.6(b). Besides, the lower BLEU score of the DL enabled approaches may not be caused by word errors. For example, it may be due to substitutions of words using synonyms or rephrasing, which does not change the meaning of the word. Fig. 3.6 also demonstrates that the joint semantic-channel coding design outperforms the traditional methods, which provides the solution to *Question 1* and *3*.

Table 3-B: The sample sentences received using different methods over Rayleigh fading channels when SNR is 18 *dB*.

Transmitted sentence	it is an important step towards equal rights for all passengers.
DeepSC	it is an important step towards equal rights for all passengers.
JSCC-[22]	it is an essential way towards our principles for democracy.
Huffman + Turbo coding	rt is a imeomant step tomdrt equal rights for atp passurerrs.
Huffman + RS coding	it is an important step towards ewiral rlrsuo for all passengess.
Bit5 + Turbo coding	it is an yoportbnt ssep sowart euual qighd fkr ill passeneers.
Bit5 + RS coding	it iw an ymp!rdbnd stgo to!atds eq.al ryghts dkr alk passengers.

Fig. 3.7 shows that the proposed performance metric, the sentence similarity, with respect to the SNR under the same total number of symbols, where the traditional approaches use 8-QAM, 64-QAM and 128-QAM. In Fig. 3.7(a), the proposed metric has shown the same tendency compared with the BLEU scores. Note that for part of the traditional methods, i.e., Huffman with Turbo coding, even if it can achieve about 20% word accuracy in BLEU score (1-gram) from Fig. 3.6(a) when SNR = 9 dB, people are usually unable to understand the meaning of texts full of errors. Thus, the sentence similarity in Fig. 3.7(a) almost converges to 0. For the DeepSC, it achieves more than 90% word accuracy in BLEU score (1-gram) when SNR is higher than 6 dB in Fig. 3.6(a), which means people can understand the texts well. Therefore the sentence similarity tends to 1. Fig. 3.6(b) and Fig. 3.7(b) show the same tendency. The benchmark, including the DNN based JSCC method in [125] under Rayleigh fading channels, also gets much higher score than the traditional approaches in terms of the sentence similarity since it can capture the features of the syntax and the relationship of the words, as well as present texts that is easier for people to understand. Few representative results are shown in Table 3-B.

In brief, we can conclude that the tendency in sentence similarity is more closer to human judgment and the DeepSC achieves the best performance in terms of both BLEU score and sentence similarity. Compared to the simulation results with BLEU score as the metric, the sentence similarity score can better measure the semantic error, which solves the *Question 2*.

Fig. 3.8 illustrates that the impact of the number of symbols per word on the 1-gram BLEU score when SNR is 12 dB. As the number of symbols per word grows, the BLEU scores increase significantly due to the increasing distance between constellations gradually. Generally, people can understand the basic meaning of transmitted sentences with over 85% word accuracy in BLEU score (1-gram). For short sentences consisted of 5 to 13 words, our proposed DeepSC can achieve 85% accuracy with 4 symbols per word, which means that we can use fewer symbols to represent one word in the environment that mainly transmits short sentences. Therefore, it can achieve high speed transmission rate. For longer sentences consisted from of 21 to 30 words, the proposed DeepSC faces more difficulties to understand the complex structure of the sentences in the transmitted texts. Hence the performance is degraded with longer sentences. One way to improve the BLEU score is to increase the average number of symbols used for each word.

3.4.3 Mutual Information

Fig. 3.9 demonstrates the relationship between SNR and mutual information after training. As we can imagine, the mutual information increases with SNR. From the figure, the performance of the transceiver trained with the mutual information estimation model outperforms that without such a model. From Fig. 3.9, with the proposed mutual information estimation model, the obtained mutual information at SNR = 4 dB is approximately same as that without the training model at SNR = 9dB. From another point of view, the mutual information estimation model leads to better learning results, i.e., data distribution, at the encoder to achieve higher data rate. In addition, this shows that introducing (3.10) in loss function can improve the mutual information of the system.

Fig. 3.10 draws the relationship between the loss value in (3.15) and the mutual

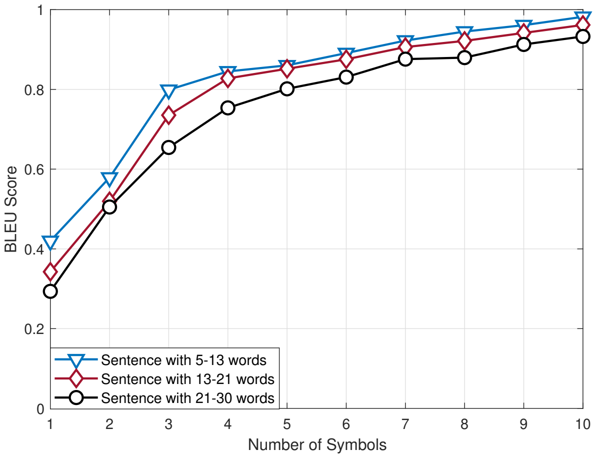


Figure 3.8: BLEU score (1-gram) versus the average number of symbols used for one word in the DeepSC.

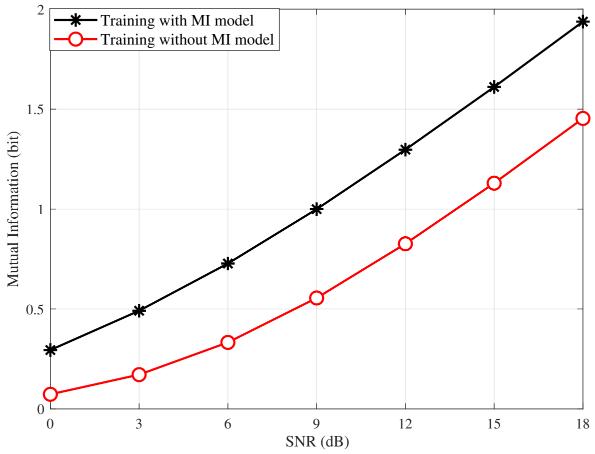


Figure 3.9: SNR versus mutual information for different trained encoders, with 8 symbols per word.

information with increasing epoch. Fig. 3.11 indicates the relationship between BLEU score and SNR. The two figures are based on models with the same structure but different training parameters, i.e., learning rate. In Fig. 3.10, the obtained mutual information is different, i.e., the mutual information of model with learning rate 0.001 increases along with decreasing loss value while the other one with learning rate 0.002 stays zero although the loss values of two models gradually converge to a stable state. From Fig. 3.11, the BLEU score with learning rate 0.001 outperforms that with learning rate 0.002, which means that even if the neural network converges to a stable state, it is possible that gradient decreases to a local minimum instead of the global minimum. During the

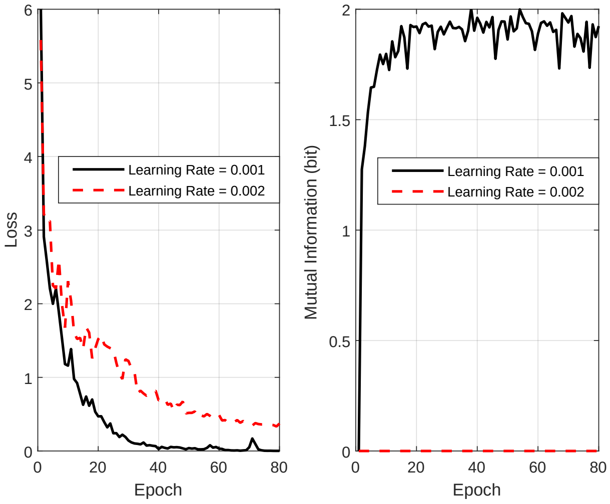


Figure 3.10: The impact of different learning rates.

training process, the mutual information can be used as a tool to decide whether the model converges effectively.

3.4.4 Transfer Learning for Dynamic Environment

In this experiment, we present the performance of transfer learning aided DeepSC for two tasks: transmitter and receiver re-training over different channels and diffident background knowledge.

Fig. 3.12 shows the training efficiency and the performance for different background knowledge, where the model will be trained and re-trained in new background knowledge with the same channel (AWGN) for different background knowledge. The models have the same structure and re-train with the same parameters in each scenario. From Fig. 3.12(a), the epochs are reduced from 30 to 5 to reach convergence. In Fig. 3.12(b), the pre-trained model can provide additional knowledge so that the corresponding model training outperforms that of re-training the whole system. This demonstrates that the transfer learning aided DeepSC can help the transceiver to accommodate the new requirements of communication environment.

Fig. 3.13 shows the training efficiency and the performance for different channels,

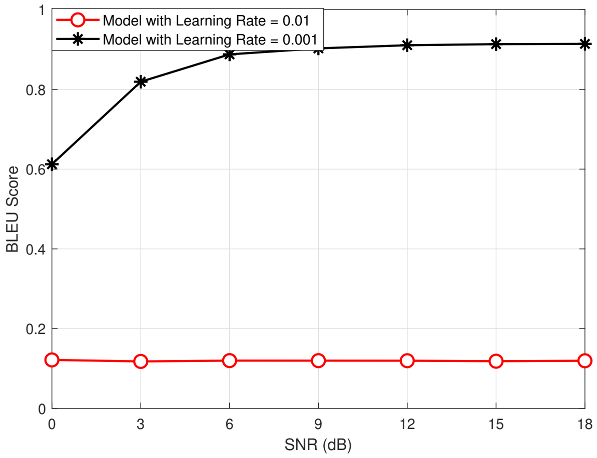


Figure 3.11: BLEU score (1-gram) versus SNR for different learning rates.

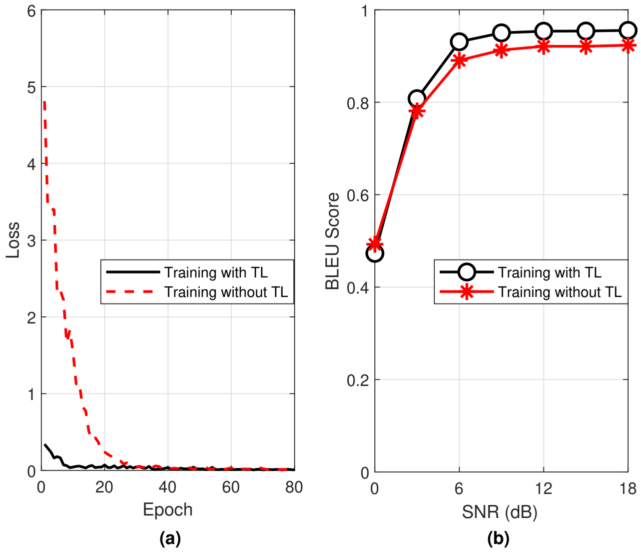


Figure 3.12: Transfer learning (TL) aided DeepSC with different background knowledge: (a) loss values versus the number of training epochs, (b) BLEU score (1-gram) versus the SNR.

where the DeepSC transceiver is pre-trained under the AWGN channel, and then it is re-trained under the erasure channel and the Rician fading channel, respectively, with the same background knowledge. The models have the same structure and re-train with the same parameters in each scenario. From Fig. 3.13(a) and Fig. 3.13(b), the adoption of the pre-trained model can speed up the training process for both the erasure channel and Rician fading channel. In Fig. 3.13(c) and Fig. 3.13(d), the performance of the DeepSC with pre-trained model is similar to that without pre-trained model channel

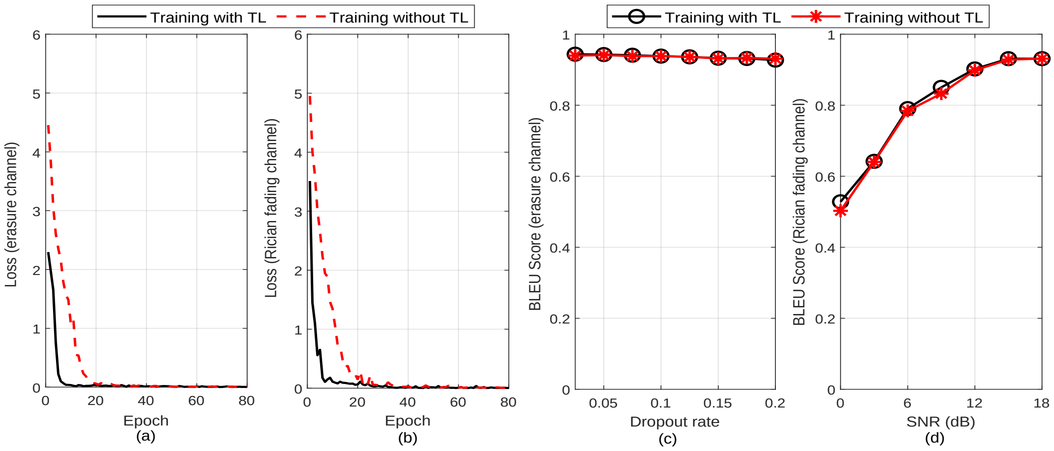


Figure 3.13: Transfer learning aided DeepSC with different channels: (a) loss values versus epochs under the erasure channel; (b) Loss values versus epochs under the Rician fading channel; (c) BLEU score (1-gram) versus the dropout rate; (d) BLEU score (1-gram) versus the SNR.

while the required complexity is reduced significantly as less number of epochs is required during the re-training process. It is further noted that the BLEU score achieved by the DeepSC is slightly degraded under the fading channel, especially in the lower SNR region, compared to that under the erasure channel.

3.4.5 Complexity Analysis

The computational complexities of the proposed DeepSC, the JSCC in [125], the RS coding, Turbo coding, are compared in Table 3-C in terms of the average processing runtime per sentence[1]. All the DL enabled approaches have lower runtime than the traditional approaches, where turbo coding costs much longer runtime in log-map iterations and the JSCC [125] requires the lowest average time due to its simple network architecture, however, it comes with poorer semantic processing capability. As a comparison, the runtime of our proposed DeepSC significantly outperforms the traditional schemes and is slight higher than JSCC [125] but with significant performance improvement.

[1] The runtime of source coding and decoding are omitted in the comparison.

Table 3-C: The average sentence processing runtime versus various schemes.

	DeepSC	JSCC [22]	RS coding	Turbo coding
Runtime	3.27ms	2.71ms	4.14ms	8.59ms

3.5 Summary

In this chapter, we have proposed a semantic communication system, named DeepSC, which jointly performs the semantic-channel coding for texts transmission. With the DeepSC, the length of input texts and output symbols are variable, and the mutual information is considered as a part of the loss function to achieve higher data rate. Besides, the deep transfer learning has been adopted to meet different transmission conditions and speed up the training of new networks by exploiting the knowledge from the pre-trained model. Moreover, we initialized sentence similarity as a new performance metric for the semantic error, which is a measure closer to human judgement. The simulation results has demonstrated that the DeepSC outperforms various benchmarks, especially in the low SNR regime. The proposed transfer learning aided DeepSC has shown its ability to adapt to different channels and knowledge with fast convergence speed. Therefore, our proposed DeepSC is a good candidate for text transmission, especially in the low SNR regime, which could be very useful for cases with massive number of devices to be connected with the limited spectrum resource. In the next chapter, we will investigate multimodal multi-user semantic communications for serving specific tasks.

Chapter 4

Task-Oriented Multi-User Semantic Communications

In this chapter, the contributions are introduced in Section 4.1. The system model is introduced in Section 4.2. The proposed single-modal multi-user semantic communications are proposed in Section 4.3. Section 4.4 details the proposed multimodal multi-user semantic communications. Numerical results are presented in Section 4.5 to show the performance of the proposed frameworks. Finally, Section 4.6 concludes this chapter.

4.1 Introduction

While semantic communications have shown the potential in the case of single-modality single-user, its applications to multi-user remain limited. In this chapter, we investigate the DL based single-modality and multimodal multi-user semantic communication systems in consideration of three intelligent tasks: image retrieval, machine translation, and VQA. For the design of multi-user semantic communications, we face the following challenges:

- *Question 1: How to extract semantic information at the transmitter for both single-*

modal and multimodal multi-user semantic communications?

- *Question 2: How to reduce the interference from other users for both single-modality and multimodal multi-user semantic communications?*
- *Question 3: How to process/fuse the received semantic information at the receiver for multi-user semantic communications to transmit multimodal data?*

For the above questions, we proposed three different DL enabled multiuser semantic communication systems, named DeepSC-IR for image retrieval, DeepSC-MT for machine translation, and DeepSC-VQA for VQA task, to address the aforementioned challenges. The main contributions of this chapter are summarized as follows:

- We propose a Transformer [62] based transmitter structure, which is applicable for both text and image transmission by effectively extracting semantic information for different tasks. This addresses the aforementioned *Q1*.
- We demonstrate the efficient methods for training the proposed structure. In particular, the transmitters and receiver in the proposed frameworks are trained jointly to eliminate distortion from the channels and interference from other users. This addresses the aforementioned *Q2*.
- Based on the proposed structure, we propose three different DL enabled multiuser semantic communication frameworks, named DeepSC-IR for image retrieval, DeepSC-MT for machine translation, and DeepSC-VQA for VQA. Specially, we propose a novel layer-wise Transformer, which can exploit more text information to guide image information, to fuse the text and image information. This addresses the aforementioned *Q3*.
- Based on extensive simulation results, the proposed frameworks outperform the traditional communication systems with lower requirements on the communication resources and improved system robustness at the low SNR regimes.

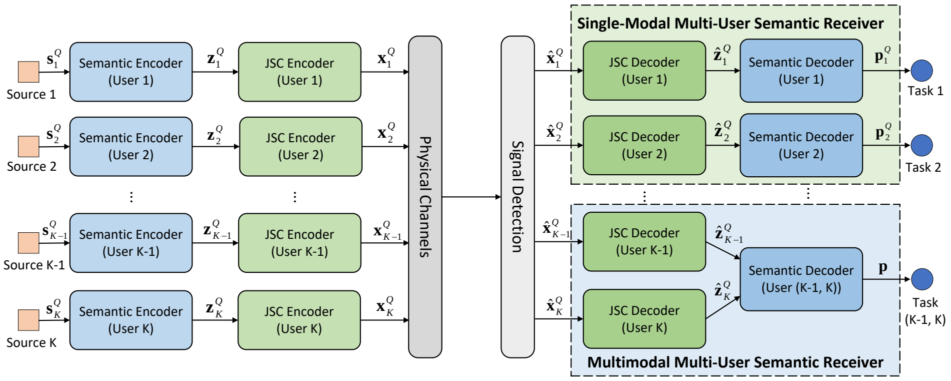


Figure 4.1: The proposed framework for multi-user semantic communication systems.

4.2 System Model

As shown in Fig. 4.1, we consider the multi-user semantic communication system, which consists of one receiver equipped with M antennas and K single-antenna transmitters. We will focus on the multi-user semantic communication system with single-modal data and multimodal data to transmit, respectively. The single-modal multi-user scenario means that each user transmits independent semantic information to perform its own task. The multimodal multi-user scenario indicates that the different types of data from different users are semantically complementary. The semantic complementary means the different multimodal data, e.g., image and text, can provide the complementary information for each other. For example, images are usually associated with tags and text explanations.

4.2.1 Semantic Transmitter

As shown in Fig. 4.1, we denote the source data of the k-th user as $\boldsymbol{s}_k^{\mathcal{Q}}$ with modality $\mathcal{Q} \subseteq \{\mathcal{I} : image, \mathcal{T} : text, \mathcal{V} : video, \mathcal{S} : speech\}$, where each source contains the semantic information. The semantic information is extracted first by

$$\boldsymbol{z}_k^{\mathcal{Q}} = S\left(\boldsymbol{s}_k^{\mathcal{Q}}; \boldsymbol{\alpha}_k^{\mathcal{Q}}\right), \tag{4.1}$$

where $\boldsymbol{z}_k^{\mathcal{Q}} \in \mathbb{R}^{L_S \times 1}$ is the semantic information with length L_S [1] and $S\left(\cdot;\boldsymbol{\alpha}_k^{\mathcal{Q}}\right)$ is the modality $\mathcal{Q}$ semantic encoder for the k-th user with learnable parameters $\boldsymbol{\alpha}_k^{\mathcal{Q}}$. Due to the limited resource and complex wireless communication environments, the semantic information of the k-th user is compressed by

$$\boldsymbol{x}_k^{\mathcal{Q}} = C\left(\boldsymbol{z}_k^{\mathcal{Q}};\boldsymbol{\beta}_k^{\mathcal{Q}}\right), \tag{4.2}$$

where $\boldsymbol{x}_k^{\mathcal{Q}} \in \mathbb{C}^{L_C \times 1}$ is the transmitted complex signal with length $L_C < L_S$ and $C\left(\cdot;\boldsymbol{\beta}_k\right)$ is the k-th user JSC encoder for modality $\mathcal{Q}$ with learnable parameters, $\boldsymbol{\beta}_k$. The neural JSC encoder in semantic communications compresses semantic information to reduce the number of transmitted symbols, as well as improve the robustness to channel variations.

After the joint source-channel encoder, the signal power is normalized as

$$\frac{1}{L_c}\mathbb{E}\left[\|\mathbf{x}_k^{\mathcal{Q}}\|_2^2\right] \leq P, \tag{4.3}$$

where P is the power constraint for each transmitter.

4.2.2 Semantic Receiver

When the transmitted signal passes a multiple-input multiple-output (MIMO) physical channel, the received signal, $\mathbf{Y} \in \mathbb{C}^{M \times L_C}$, at the receiver can be expressed as

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}, \tag{4.4}$$

where $\mathbf{X}^T = \left[\boldsymbol{x}_1^{\mathcal{Q}}, \boldsymbol{x}_2^{\mathcal{Q}}, \cdots, \boldsymbol{x}_K^{\mathcal{Q}}\right] \in \mathbb{C}^{L_C \times K}$ denotes transmit symbols from all K users, $\mathbf{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_K] \in \mathbb{C}^{M \times K}$ is the channel matrix between the BS and users. For the Rayleigh fading channel, the channel coefficients follows $\boldsymbol{h}_k \sim \mathcal{CN}(0, \mathbf{I}_M)$; for the Rician fading channel, it follows$\boldsymbol{h}_k \sim \mathcal{CN}(\mu\mathbf{1}_{M\times1}, \sigma^2\mathbf{I}_M)$ with $\mu = \sqrt{r/(r+1)}$ and $\sigma = \sqrt{1/(r+1)}$, where $\mathbf{I}_M$ is the $M \times M$ is the identity matrix, $\mathbf{1}_{M\times1}$ is the all-one vector with

[1] The transmitted lengths of $\boldsymbol{z}_k^{\mathcal{Q}}$ for different users could be different. To simplify the analysis, we choose the same length here for all users.

length M, and r is the Rician coefficient. $\mathbf{N} \in \mathbb{C}^{M \times L_C}$ denotes the circular symmetric Gaussian noise. The elements of $\mathbf{N}$ are i.i.d with zero mean and variance σ_n^2, and SNR is $\sum\limits_k \left\|\boldsymbol{h}_k \boldsymbol{x}_k^{\mathcal{Q}}\right\|^2 / \sigma_n^2$.

Subsequently, the transmitted signals are recovered by the linear minimum mean-squared error (L-MMSE) detector with the estimated CSI, which could be given by

$$\widehat{\mathbf{X}} = \widehat{\mathbf{H}}^{\mathrm{H}} \left(\widehat{\mathbf{H}} \widehat{\mathbf{H}}^{\mathrm{H}} + \sigma_n^2 \mathbf{I} \right)^{-1} \mathbf{Y}, \tag{4.5}$$

where $\widehat{\mathbf{X}}^T = \left[\hat{\boldsymbol{x}}_1^{\mathcal{Q}}; \hat{\boldsymbol{x}}_2^{\mathcal{Q}}; \cdots; \hat{\boldsymbol{x}}_K^{\mathcal{Q}}\right] \in \mathbb{C}^{L_C \times K}$ is the recovered transmitted signals, $\widehat{\mathbf{H}} = \mathbf{H} + \Delta\mathbf{H}$ is the estimated CSI, in which $\Delta\mathbf{H}$ is the estimation error with $\Delta\mathbf{H} \in \mathcal{CN}(0, \sigma_e^2)$. Here, σ_e^2 is the measure of how accurate the channel estimation is.

The semantic information from the k-th user, $\hat{\boldsymbol{z}}_k^{\mathcal{Q}} \in \mathbb{R}^{L_S \times 1}$, is recovered by the JSC decoder as

$$\hat{\boldsymbol{z}}_k^{\mathcal{Q}} = C^{-1}\left(\hat{\boldsymbol{x}}_k^{\mathcal{Q}}; \boldsymbol{\gamma}_k^{\mathcal{Q}}\right), \tag{4.6}$$

where $C^{-1}\left(\hat{\boldsymbol{x}}_k^{\mathcal{Q}}; \boldsymbol{\gamma}_k^{\mathcal{Q}}\right)$[2] is JSC decoder for the k-th user with the modality $\mathcal{Q}$ and the learned parameters $\boldsymbol{\gamma}_k^{\mathcal{Q}}$. The JSC decoder aims to decompress the semantic information while mitigating the effects of channel distortion and inter-user interference. For serving the different transmission tasks, we will have the single-modal semantic receiver and the multimodal semantic receiver.

4.2.2.1 Single-Modal Semantic Receiver

For single-modal semantic transmission, the semantic information from each user is exploited to perform different tasks independently. The recovered semantic information is employed for the task of the k-th user by

$$\boldsymbol{p}_k^{\mathcal{Q}} = S^{-1}\left(\hat{\boldsymbol{z}}_k^{\mathcal{Q}}; \boldsymbol{\varphi}_k^{\mathcal{Q}}\right), \tag{4.7}$$

[2] In order to reduce the number of representation symbols, we use $\cdot^{-1}$ here to represent the decoder.

where $\boldsymbol{p}_k^{\mathcal{Q}}$ is the predicted probability of the task, e.g., the predict probability of each word in the translated sentence for the machine learning task, and retrieval results for the image retrieval task. $S^{-1}(\cdot; \boldsymbol{\varphi}_k^{\mathcal{Q}})$ is the modality $\mathcal{Q}$ semantic decoder for the k-th user with learning parameters $\boldsymbol{\varphi}_k^{\mathcal{Q}}$.

4.2.2.2 Multimodal Semantic Receiver

With the multimodal semantic information, the final task is performed directly by merging the semantic information from different users. This is expressed by

$$\boldsymbol{p} = S^{-1}\left(\hat{\boldsymbol{z}}_1^{\mathcal{Q}}, \hat{\boldsymbol{z}}_2^{\mathcal{Q}}, \cdots, \hat{\boldsymbol{z}}_K^{\mathcal{Q}}; \boldsymbol{\varphi}_{(1,2,\cdots,K)}\right), \tag{4.8}$$

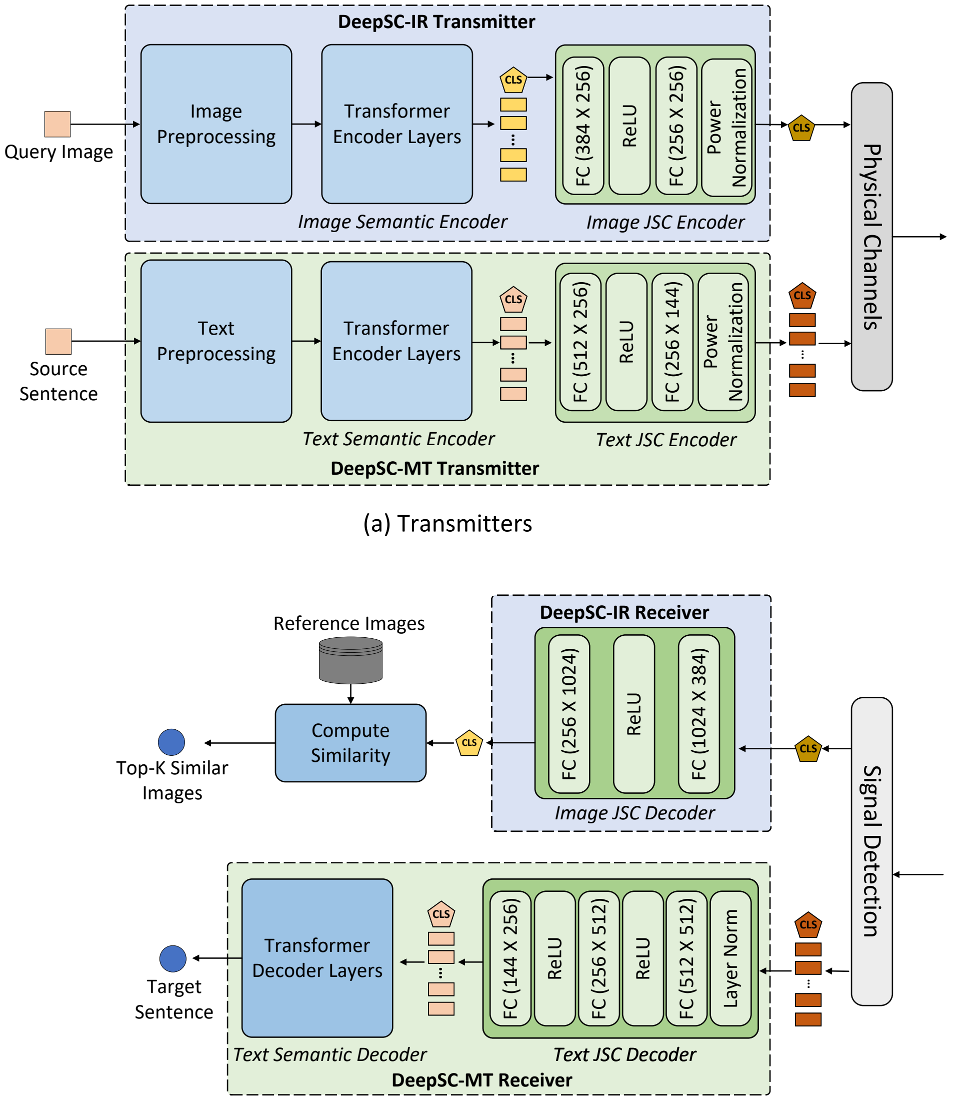
where $\boldsymbol{p}$ is the results of the multimodal task and $S^{-1}\left(\cdot; \boldsymbol{\varphi}_{(1,2,\cdots,K)}\right)$ is the multimodal semantic decoder with learnable parameters $\boldsymbol{\varphi}_{(1,2,\cdots,K)}$.

4.3 Single-Modal Multi-user Semantic Communications

In this section, we focus on the multi-user semantic communication system to transmit single-modal data from multiple users. We propose semantic communication systems for the image retrieval task (e.g., DeepSC-IR), and the machine translation task (e.g., DeepSC-MT). Particularly, we adopt the vision Transformer for image understanding and text Transformer for text understanding, in which the vision Transformer and text Transformer are assumed to have the same network structure.

4.3.1 Image Retrieval Task

Assume that $\mathcal{D}_k^{\mathcal{I}} = \left\{\left(\boldsymbol{s}_{k,j}^{\mathcal{I}}, l_{k,j}^{\mathcal{I}}\right)\right\}_{j=1}^{D}$ with size D is the training image dataset for the k-th user, where $\boldsymbol{s}_{k,j}^{\mathcal{I}}$ and $l_{k,j}^{\mathcal{I}}$ are the j-th image and its corresponding label in $\mathcal{D}_k^{\mathcal{I}}$, respectively. $S_{\texttt{IR}}\left(\cdot; \boldsymbol{\alpha}_k^{\mathcal{I}}\right)$, $C_{\texttt{IR}}\left(\cdot; \boldsymbol{\beta}_k^{\mathcal{I}}\right)$, and $C_{\texttt{IR}}^{-1}\left(\cdot; \boldsymbol{\gamma}_k^{\mathcal{I}}\right)$ represent the semantic encoder, JSC encoder, and JSC decoder of the i-th user for the image retrieval task, respectively.



(a) Transmitters

(b) Receivers

Figure 4.2: The proposed network structure of single-modal multi-user semantic communications, including the DeepSC-IR transceiver and DeepSC-MT transceiver.

4.3.1.1 Model Description

The proposed image retrieval network is shown in Fig. 4.2. Specifically, the DeepSC-IR transmitter consists of an image semantic encoder to extract image semantic information to be transmitted and a JSC encoder to compress the semantic information, where the semantic encoder includes multiple vision Transformer layers and the JSC encoder uses fully-connected layers with different units. Compared with CNNs, the vision Transformer can be better in capturing the global features and more robust to input image distortions. Specifically, we choose only the <CLS> vector-token to be transmitted as it represents the global image information. After transmitting and performing signal detection, the DeepSC-IR receiver employs the JSC decoder with different units to recover the transmitted image semantic information. Compared with CNNs, the FC layer is good at dealing with the two-dimensions inputs and preserving the entire attributes at once.

The recovered semantic information after the JSC decoder at the receiver can be used to match the other image semantic information in the database by computing the euclidean distance to find similar images as

$$d(\boldsymbol{z}_{k,j}^{\mathcal{I}}, \boldsymbol{z}_{k,i}^{\mathcal{I}}) = \left\|\boldsymbol{z}_{k,j}^{\mathcal{I}} - \boldsymbol{z}_{k,i}^{\mathcal{I}}\right\|_2. \tag{4.9}$$

The euclidean distance becomes the cosine similarity when $\boldsymbol{z}_{k,j}^{\mathcal{I}}$ and $\boldsymbol{z}_{k,i}^{\mathcal{I}}$ are l^2 normalized.

4.3.1.2 Training Algorithm

As shown in Algorithm 4.1, the training process of the DeepSC-IR consists of two phases due to different loss functions. The first phase is to train the semantic encoder, and the second phase is to train the JSC codec.

In the first phase, the semantic encoder will be trained by the function, `Train Semantic Encoder`. Different from other tasks, image retrieval is performed by computing the distance between images to return similar images. Therefore, we choose metric learning as the learning paradigm. Such paradigm aims at minimizing the distance between images belonging to the same category and maximizing the distance between images belonging to different categories. The loss function [101] is expressed by

$$\mathcal{L}_{\texttt{IR}} = \mathbb{E}\left[\sum_{l_{k,j}^{\mathcal{I}}=l_{k,i}^{\mathcal{I}}} \left(1-(\boldsymbol{z}_{k,j}^{\mathcal{I}})^{\mathrm{T}}\boldsymbol{z}_{k,i}^{\mathcal{I}}\right)\right] + \mathbb{E}\left[\sum_{l_{k,j}^{\mathcal{I}}\neq l_{k,i}^{\mathcal{I}}} \left((\boldsymbol{z}_{k,j}^{\mathcal{I}})^{\mathrm{T}}\boldsymbol{z}_{k,i}^{\mathcal{I}}-\xi\right)_+\right], \tag{4.10}$$

where the operator $(x)_+$ returns $\max(x,0)$, $\boldsymbol{z}_{k,j}^{\mathcal{I}}$ is the image semantic information, i.e., the <CLS> token from the outputs of image semantic encoder, ξ is a constant margin to prevent the training signal from being overwhelmed by easy negatives. After training the semantic encoder with (4.10), the semantic encoder becomes capable of extracting semantic image information, which returns a smaller euclidean distance if they are from images within the same category.

In order to compress semantic redundancy while overcoming the distortion from the channels, the JSC codec is trained in the second phase. The MSE [134] is employed as the loss function to minimize the difference between the transmitted and recovered semantic image information, which is represented as

$$\mathcal{L}_{\texttt{MSE}} = \mathbb{E}\left[\left\|\hat{\boldsymbol{z}}_{k,j}^{\mathcal{I}} - \boldsymbol{z}_{k,j}^{\mathcal{I}}\right\|_2^2\right], \tag{4.11}$$

where $\hat{\boldsymbol{z}}_{k,j}^{\mathcal{I}}$ is the semantic image information recovered at receiver and $\boldsymbol{z}_{k,j}^{\mathcal{I}}$ is the transmitted semantic image information. By minimizing the $\mathcal{L}_{\texttt{MSE}}$, the JSC codec will learn to compress and decompress semantic image information for fewer transmitted symbols while guaranteeing on accurate semantic recovery by dealing with the distortion and interference from the channels and inter-users.

4.3.2 Machine Translation Task

Assume $\mathcal{D}_k^{\mathcal{T}} = \left\{\left(\boldsymbol{s}_{k,j}^{\mathcal{T}}, \boldsymbol{p}_{k,j}^{\mathcal{T}}\right)\right\}_{j=1}^{D}$ with size D as the training text dataset for the k-th user, where $\boldsymbol{s}_{k,j}^{\mathcal{T}}$ and $\boldsymbol{p}_{k,j}^{\mathcal{T}}$ are the j-th sentence in the source language and the translated sentence in the target language, respectively. $\boldsymbol{s}_{k,j}^{\mathcal{T}}[n]$ and $\boldsymbol{p}_{k,j}^{\mathcal{T}}[n]$ represent the n-th word in sentence $\boldsymbol{s}_{k,j}^{\mathcal{T}}$ and $\boldsymbol{p}_{k,j}^{\mathcal{T}}$, respectively. $S_{\texttt{MT}}\left(\cdot;\boldsymbol{\alpha}_k^{\mathcal{T}}\right)$, $C_{\texttt{MT}}\left(\cdot;\boldsymbol{\beta}_k^{\mathcal{T}}\right)$, $C_{\texttt{MT}}^{-1}\left(\cdot;\boldsymbol{\gamma}_k^{\mathcal{T}}\right)$, and $S_{\texttt{MT}}^{-1}\left(\cdot;\boldsymbol{\varphi}_k^{\mathcal{T}}\right)$ represent the semantic encoder, JSC encoder, JSC decoder, and semantic decoder of the k-th user for the machine translation task, respectively.

4.3.2.1 Model Description

The proposed machine translation network is shown in Fig. 4.2. The transmitter includes a text semantic encoder and a text JSC encoder to extract and compress the semantic text

Algorithm 4.1: *DeepSC-IR Training Algorithm.*

```
**Input**: The training dataset $\mathcal{D}_i$.
**Function**: Train Semantic Coder{}
    Choose mini-batch data $\{(\boldsymbol{s}_{i,j}, l_{i,j})\}_{j=n}^{n+B}$ from $\mathcal{D}_i$.
    $\{S_{\mathtt{IR}}(\boldsymbol{s}_{i,j};\boldsymbol{\alpha}_i)\}_{j=n}^{n+B} \rightarrow \{\boldsymbol{z}_{i,j}\}_{j=n}^{n+B}$,
    Compute the $\mathcal{L}_{\mathtt{IR}}$ by (4.10) with $\{\boldsymbol{z}_{i,j}\}_{j=n}^{n+B}$,
    Train $\boldsymbol{\alpha}_i \rightarrow$ Gradient descent $(\boldsymbol{\alpha}_i, \mathcal{L}_{\mathtt{IR}})$,
**Return** $S_{\mathtt{IR}}(;\boldsymbol{\alpha}_i)$.

**Function**: Train JSC Codec{}
    Choose mini-batch data $\{(\boldsymbol{s}_{i,j}, l_{i,j})\}_{j=n}^{n+B}$ from $\mathcal{D}_i$.
    **Transmitter**:
        $S_{\mathtt{IR}}(\boldsymbol{s}_{i,j};\boldsymbol{\alpha}_i) \rightarrow \boldsymbol{z}_{i,j}$,
        $C_{\mathtt{IR}}(\boldsymbol{z}_{i,j};\boldsymbol{\beta}_i) \rightarrow \boldsymbol{x}_{i,j}$,
        Transmit $\boldsymbol{x}_{i,j}$ over the channel,
    **Receiver**:
        Receive $\mathbf{Y}$,
        MIMO detection by (4.5) to get $\hat{\boldsymbol{x}}_{i,j}$,
        $C_{\mathtt{IR}}^{-1}(\hat{\boldsymbol{x}}_{i,j};\boldsymbol{\gamma}_i) \rightarrow \hat{\boldsymbol{z}}_{i,j}$,
    Compute the $\mathcal{L}_{\mathtt{MSE}}$ by (4.11) with $\boldsymbol{z}_{i,j}$, $\hat{\boldsymbol{z}}_{i,j}$,
    Train $\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i \rightarrow$ Gradient descent $(\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i, \mathcal{L}_{\mathtt{MSE}})$,
**Return** $C_{\mathtt{IR}}(;\boldsymbol{\beta}_i), C_{\mathtt{IR}}^{-1}(;\boldsymbol{\gamma}_i)$.
```

information, respectively, where the text semantic encoder adopts multiple Transformer encoder layers and the designed text JSC encoder in Fig. 4.2 is with multiple dense layers. Compared with RNNs, the Transformer can be better in capturing the relations between sentences and trained significantly faster than architectures based on recurrent layers. At the receiver, the designed text JSC decoder recovers the semantic text information from distorted signals. Subsequently, the semantic decoder consists of multiple Transformer decoder layers to derive the translated sentence based on the recovered semantic text information.

4.3.2.2 Training Algorithm

As shown in Algorithm 4.2, the training process of DeepSC-MT consists of three phases: `Train Semantic Codec`, `Train JSC Codec`, and `Train Whole Network`.

The first phase is `Train Semantic Codec`. The semantic codec, $S_{\mathtt{MT}}\left(\cdot;\boldsymbol{\alpha}_k^{\mathcal{T}}\right)$ and $S_{\mathtt{MT}}^{-1}\left(\cdot;\boldsymbol{\varphi}_k^{\mathcal{T}}\right)$, will be trained firstly with the CE loss function, which enables the model

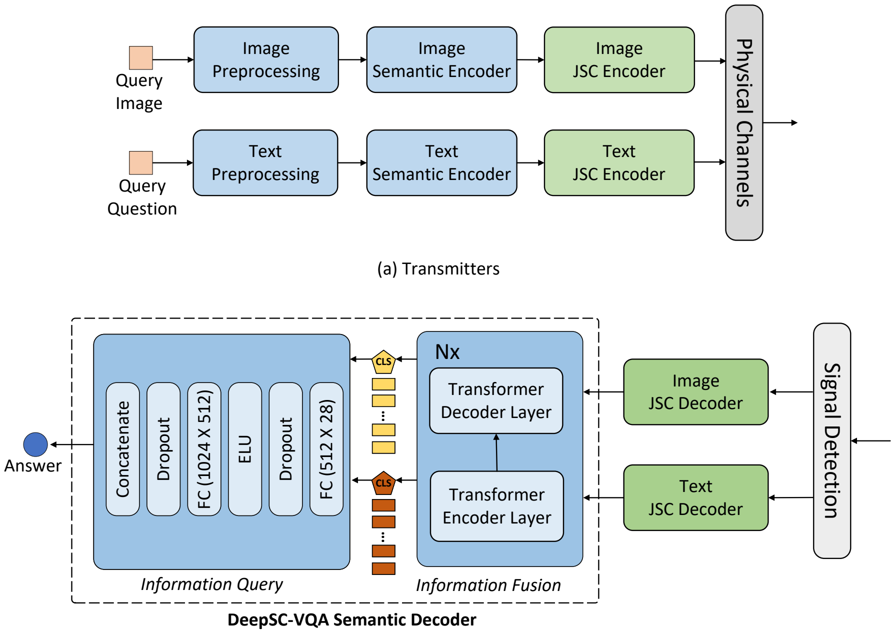
Algorithm 4.2: *DeepSC-MT Training Algorithm.*

```
 1: Input: The training dataset $\mathcal{D}_i$.
 2: Function: Train Semantic Codec{}
 3:   Choose mini-batch data $\{(\boldsymbol{s}_{k,j}, \boldsymbol{p}_{k,j})\}_{j=n}^{n+B}$ from $\mathcal{D}_k^{\mathtt{T}}$.
 4:   For $j = n \rightarrow n + B$
 5:     $S_{\mathtt{MT}}\left(\boldsymbol{s}_{k,j}; \boldsymbol{\alpha}_k\right) \rightarrow \boldsymbol{z}_{k,j}$,
 6:     $S_{\mathtt{MT}}^{-1}\left(\boldsymbol{z}_{k,j}; \boldsymbol{\varphi}_k\right) \rightarrow \hat{\boldsymbol{p}}_{k,j}$,
 7:     Compute the $\mathcal{L}_{\mathtt{MT}}$ by (4.12) with $\boldsymbol{p}_{k,j}$, $\hat{\boldsymbol{p}}_{k,j}$,
 8:   End
 9:   Train $\boldsymbol{\alpha}_k, \boldsymbol{\varphi}_k \rightarrow$ Gradient descent $(\boldsymbol{\alpha}_k, \boldsymbol{\varphi}_k, \mathcal{L}_{\mathtt{MT}})$,
10: Return $S_{\mathtt{MT}}\left(; \boldsymbol{\alpha}_k\right)$, $S_{\mathtt{MT}}^{-1}\left(; \boldsymbol{\varphi}_k\right)$.

 1: Function: Train JSC Codec{}
 2:   Choose mini-batch data $\{(\boldsymbol{s}_{k,j}, \boldsymbol{p}_{k,j})\}_{j=n}^{n+B}$ from $\mathcal{D}_k^{\mathtt{T}}$.
 3:   For $j = n \rightarrow n + B$
 4:   Transmitter:
 5:     $S_{\mathtt{MT}}\left(\boldsymbol{s}_{k,j}; \boldsymbol{\alpha}_k\right) \rightarrow \boldsymbol{z}_{k,j}$,
 6:     $C_{\mathtt{MT}}\left(\boldsymbol{z}_{k,j}; \boldsymbol{\beta}_k\right) \rightarrow \boldsymbol{x}_{k,j}$,
 7:     Transmit $\boldsymbol{x}_{k,j}$ over the channel,
 8:   Receiver:
 9:     Receive $\mathbf{Y}$,
10:     MIMO detection by (4.5) to get $\hat{\boldsymbol{x}}_{k,j}$,
11:     $C_{\mathtt{MT}}^{-1}\left(\hat{\boldsymbol{x}}_{k,j}; \boldsymbol{\gamma}_k\right) \rightarrow \hat{\boldsymbol{z}}_{k,j}$,
12:   Compute the $\mathcal{L}_{\mathtt{MSE}}$ with (4.13),
13:   Train $\boldsymbol{\beta}_k, \boldsymbol{\gamma}_k \rightarrow$ Gradient descent $(\boldsymbol{\beta}_k, \boldsymbol{\gamma}_k, \mathcal{L}_{\mathtt{MSE}})$,
14: Return $C_{\mathtt{MT}}\left(; \boldsymbol{\beta}_k\right), C_{\mathtt{MT}}^{-1}\left(; \boldsymbol{\gamma}_k\right)$.

 1: Function: Train Whole Network{}
 2:   Choose mini-batch data $\{(\boldsymbol{s}_{i,j}, l_{i,j})\}_{j=n}^{n+B}$ from $\mathcal{D}_i$.
 3:   For $j = n \rightarrow n + B$
 4:     Repeat line 5, 4-11, and 6 to get $\hat{\boldsymbol{p}}_{k,j}^{\mathcal{T}}$,
 5:   Compute the $\mathcal{L}_{\mathtt{MT}}$ by (4.12) with $\boldsymbol{p}_{k,j}, \hat{\boldsymbol{p}}_{k,j}$,
 6:   Train $\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k, \boldsymbol{\gamma}_k, \boldsymbol{\varphi}_k \rightarrow$ Gradient descent $(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k, \boldsymbol{\gamma}_k, \boldsymbol{\varphi}_k, \mathcal{L}_{\mathtt{MT}})$,
 7: Return $S_{\mathtt{MT}}\left(; \boldsymbol{\alpha}_k\right), C_{\mathtt{MT}}\left(; \boldsymbol{\beta}_k\right), C_{\mathtt{MT}}^{-1}\left(; \boldsymbol{\gamma}_k\right)$, $S_{\mathtt{MT}}^{-1}\left(; \boldsymbol{\varphi}_k\right)$.
```

to convert the meaning to the target sentence by learning the target language word distribution. The CE loss function [58] is represented by

$$\mathcal{L}_{\mathtt{MT}} = \mathbb{E}\left[-\sum_{n} P(\boldsymbol{p}_{k,j}^{\mathcal{T}}[n]) \log\left(P(\hat{\boldsymbol{p}}_{k,j}^{\mathcal{T}}[n])\right)\right], \tag{4.12}$$

where $P(\hat{\boldsymbol{p}}_{k,j}^{\mathcal{T}}[n])$ is the predicted probability that the n-th word appears in sentence $\hat{\boldsymbol{p}}_{k,j}^{\mathcal{T}}$, and $P(\boldsymbol{p}_{k,j}^{\mathcal{T}}[n])$ is the real probability that the n-th word appears in the sentence $\boldsymbol{p}_{k,j}^{\mathcal{T}}$.



(a) Transmitters

(b) Receivers

Figure 4.3: The proposed network structure of multimodal multi-user semantic communication system with DeepSC-VQA transceiver.

After convergence, the model learns the syntax, phrase, the meaning of words in the target language.

In the second training phase that is listed as `Train JSC Codec` of Algorithm 2, the JSC codec, $C_{\texttt{MT}}(\cdot;\boldsymbol{\beta}_k^{\mathcal{T}})$ and $C_{\texttt{MT}}^{-1}(\cdot;\boldsymbol{\gamma}_k^{\mathcal{T}})$, are also trained to learn the compress and decompress semantic text information, as well as deal with the channel distortion and multi-user interference with the MSE loss function given by

$$\mathcal{L}_{\texttt{MSE}} = \mathbb{E}\left[\left\|\hat{\boldsymbol{z}}_{k,j}^{\mathcal{T}} - \boldsymbol{z}_{k,j}^{\mathcal{T}}\right\|_2^2\right], \tag{4.13}$$

where $\hat{\boldsymbol{z}}_{k,j}^{\mathcal{T}}$ is the recovered semantic text information at the receiver and $\boldsymbol{z}_{k,j}^{\mathcal{T}}$ is the transmitted semantic text information, i.e., the all outputs of text semantic encoder.

Different from the DeepSC-IR training algorithm, there exists a semantic decoder at the DeepSC-MT receiver. This means that semantic errors between $\hat{\boldsymbol{z}}_{k,j}^{\mathcal{T}}$ and $\boldsymbol{z}_{k,j}^{\mathcal{T}}$ can be mitigated by jointly training the whole system shown as `Train Whole Network` in

Algorithm 2 with the loss function (4.12).

The separate training can make it easier to fit with adaptive communication environment. e.g., new channel environment. With such a separate design, we only need to replace and train the JSC encoder, and then train the entire system for several epochs, which can converge quickly and reduce the difficulty of design. In contrast, with the joint design, we will have to re-design and re-train the entire system from scratch. Even if the system is designed separately, it can still be trained and optimized jointly due to the advantages of deep learning.

4.4 Multimodal Multi-user Semantic Communications

In this section, the multimodal multi-user semantic communications are investigated for serving the VQA task, namely DeepSC-VQA, in which the transmitters adopt the same structures as that of DeepSC-IR for images and DeepSC-MT for texts. They also share the same JSC decoder design. Particularly, a novel semantic decoder is proposed to merge the image-text semantic information.

4.4.1 Model Description

Assume that the k-th user for image transmission and the i-th user for text transmission, $\mathcal{D}_{k,i}^{\mathcal{I},\mathcal{T}} = \left\{ \left(\boldsymbol{s}_{k,j}^{\mathcal{I}}, \boldsymbol{s}_{i,j}^{\mathcal{T}}, l_{(k,i),j} \right) \right\}_{j=1}^{D}$ with size D is the training dataset, where $\boldsymbol{s}_{k,j}^{\mathcal{I}}$ is the j-th image from the k-th user, $\boldsymbol{s}_{i,j}^{\mathcal{T}}$ is the j-th text from the i-th user, and $l_{(k,i),j}$ is the answer label for $\boldsymbol{s}_{k,j}^{\mathcal{I}}$ and $\boldsymbol{s}_{i,j}^{\mathcal{T}}$. $S_{\texttt{VQA}}\left(\cdot;\boldsymbol{\alpha}_k^{\mathcal{I}}\right)$, $C_{\texttt{VQA}}\left(\cdot;\boldsymbol{\beta}_k^{\mathcal{I}}\right)$, $C_{\texttt{VQA}}^{-1}\left(\cdot;\boldsymbol{\gamma}_k^{\mathcal{I}}\right)$ are the image semantic encoder, image JSC encoder, and image JSC decoder of the k-th user, respectively. $S_{\texttt{VQA}}\left(\cdot;\boldsymbol{\alpha}_i^{\mathcal{T}}\right)$, $C_{\texttt{VQA}}\left(\cdot;\boldsymbol{\beta}_i^{\mathcal{T}}\right)$, $C_{\texttt{VQA}}^{-1}\left(\cdot;\boldsymbol{\gamma}_i^{\mathcal{T}}\right)$ are the text semantic encoder, text JSC encoder, and text JSC decoder of the i-th user, respectively. $S_{\texttt{VQA}}^{-1}\left(\cdot;\boldsymbol{\varphi}_{(k,i)}\right)$ represents joint semantic decoder of the i-th and the k-th user for the VQA task.

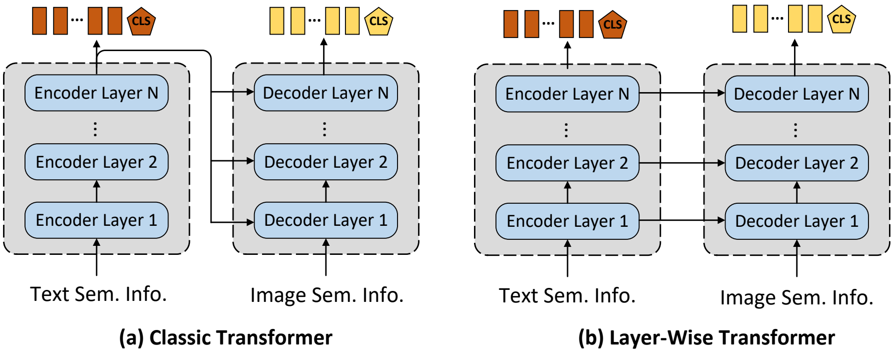
As shown in Fig. 4.3, the proposed DeepSC-VQA network consists of one image transmitter, one text transmitter, and one receiver for simplicity. For the DeepSC-VQA transmitters and receivers, we adopt the same structures as the image transmitter of DeepSC-IR and text transmitter of DeepSC-MT to unify the transmitter paradigm. At the receiver, the structures of the image JSC decoder and text JSC decoder are also the same as that of the image JSC decoder in DeepSC-IR and that of the text JSC decoder in DeepSC-MT. Besides, we develop a new semantic decoder network for image-text information fusion, which includes two modules: information query module and information fusion module.

4.4.1.1 Information Query

Image and text are with different modalities, in which each modality can provide the semantic complementary information for each other. To exploit the semantic complementary, the layer-wise Transformer is adopted. Fig. 4.4 shows the comparison between the classic Transformer and the layer-wise Transformer. Different from the classic Transformer, where the decoder layers exploit the output tokens of the last layer of encoder as the input, the layer-wise Transformer employs the output tokens of each encoder layer as the input of each decoder layer. Such a design can generate more text information than classic Transformer and guide the image information query in the decoder more efficiently, which does not introduce any costs.

4.4.1.2 Information Fusion

After the information query, the layer-wise Transformer has already captured keywords in the text information and the corresponding regions in image information, which has reflected in the output tokens. We will then need to fuse keywords and the corresponding image regions to get the answer. As mentioned in Section II, the <CLS> token represents the global descriptor. Therefore, the <CLS> tokens in the output tokens of the Transformer encoder and Transformer decoder represent the global text information and global image information, respectively. Using the text <CLS> and image <CLS>, we design the information fusion module as shown in Fig. 4.3, where dropout layers are used here to avoid over-fitting. Compared with employing descriptor fusion networks to get



(a) Classic Transformer

(b) Layer-Wise Transformer

Figure 4.4: Comparison between classic Transformer and layer-wise Transformer.

global descriptor, the proposed information fusion employ the <CLS> global descriptors directly and achieve the similar answer accuracy but without the additional descriptor fusion networks.

4.4.2 Training Algorithm

Similar to the DeepSC-MT training algorithm, the DeepSC-VQA is trained jointly by three phases but with different loss functions.

The first phase is `Train Semantic Codec`, the semantic codec of DeepSC-VQA, $S_{\texttt{VQA}}\left(\cdot;\boldsymbol{\alpha}_k^{\mathcal{I}}\right)$, $S_{\texttt{VQA}}\left(\cdot;\boldsymbol{\alpha}_i^{\mathcal{T}}\right)$, $S_{\texttt{VQA}}^{-1}\left(\cdot;\boldsymbol{\varphi}_{(k,i)}\right)$, is trained jointly by the CE loss function,

$$\mathcal{L}_{\texttt{VQA}} = \mathbb{E}\left[-P\left(l_{(k,i),j}\right)\log\left(P\left(\hat{l}_{(k,i),j}\right)\right)\right], \tag{4.14}$$

where $P(l_{(k,i),j})$ and $P(\hat{l}_{(k,i),j})$ are the real and predicted probability of answer, respectively. By reducing the loss value of CE, the network learns to predict the answer with the highest probability of accuracy.

After training the semantic codec, JSC codecs are trained to compress by JSC encoder to reduce the number of transmitted symbols, and then decompress by the JSC decoder to recover semantic information accurately over multiple user physical channels. The image and text JSC codec are trained jointly by the function `Train JSC Codec`, in which the loss function is designed as

$$\mathcal{L}_{\texttt{MSE}}^{(\texttt{VQA})} = \mathbb{E}\left[\left\|\hat{\boldsymbol{z}}_{k,j}^{\mathcal{I}} - \boldsymbol{z}_{k,j}^{\mathcal{I}}\right\|_2^2 + \left\|\hat{\boldsymbol{z}}_{i,j}^{\mathcal{T}} - \boldsymbol{z}_{i,j}^{\mathcal{T}}\right\|_2^2\right], \tag{4.15}$$

where $\boldsymbol{z}_{k,j}^{\mathcal{I}}$ and $\boldsymbol{z}_{i,j}^{\mathcal{T}}$ are the transmitted semantic image and text information, respectively. $\hat{\boldsymbol{z}}_{k,j}^{\mathcal{I}}$ and $\hat{\boldsymbol{z}}_{i,j}^{\mathcal{T}}$ are the recovered semantic image and text information at the receiver, respectively.

There exists error propagation from the JSC decoders to the semantic receiver because of the imperfect semantic information recovery in the low SNR regimes. Therefore, the whole DeepSC-VQA network is trained jointly with loss function (4.14) to reduce the error propagation, which is the function `Train Whole Network`.

4.5 Simulation Results

In this section, we compare the proposed multi-user semantic communication systems with traditional source coding and channel coding methods over various channels, in which both the perfect and imperfect CSI are considered.

4.5.1 Implementation Details

4.5.1.1 The Datasets

We choose four popular datasets commonly used for the image retrieval task. *Stanford Online Products* [135] consists of 120,053 online products images representing 22,634 categories, in which 11,318 categories are used for training and the remaining 11,316 categories are used for testing. *CUB-200-2011* [136] has 200 bird categories with 11,789 images. We split the first 100 classes for training and the rest of 100 classes for testing. *Cars196* [137] contains 16,185 images corresponding to 196 car categories with the first 98 categories to be used for training. The remaining 98 categories are used for testing. *In-Shop Clothes* [138] contains 72,172 cloth images belonging to 7,986 categories, in which 3997 categories are used for training and the other 3985 categories will be used for testing.

For the machine translation task, we adopt the *WMT 2018 Chinese-English news track*, which contains 202,221 pairs for training and 50,556 pairs for testing. The dataset is filtered into the length of English sentences with 5 to 75 words.

For the VQA task, we adopt the popular dataset: *CLEVR* [139], which consists of a training set of 70,000 images and 699,989 questions and a test set of 15,000 images and 149,991 questions.

4.5.1.2 Training Settings

The image semantic encoder of DeepSC-IR is based on the public implementation of DeiT-small model[3] with 12 Transformer encoder layers, in which the width of each layer is 384. The setting of the `Train Semantic Encoder` of DeepSC-IR is the Adam optimizer with learning rate 3×10^{-5}, weight decay 5×10^{-4}, batch size of 64, and epoch of 40. The setting of the `Train JSC Encoder` of DeepSC-IR is the Adam optimizer with learning rate 1×10^{-3}, batch size of 64, and epoch of 100. During the training phase, the data augmentation is used to resize the image to 256 × 256 and then take a random crop of size 224 × 224 combined with random horizontal flipping. In the test phase, the images are resized to 256 × 256 first and centrally cropped to 224 × 224.

The text semantic codec of DeepSC-MT is based on the public implementation of the Transformer model[4] with 6 Transformer encoder layers and decoder layers, in which the width of each layer is 512. The setting of the `Train Semantic Codec` of DeepSC-MT is the Adam optimizer with learning rate 1×10^{-5}, betas of 0.9 and 0.98, batch size of 64, and epoch of 10. The setting of the `Train JSC Codec` of DeepSC-MT is the Adam optimizer with learning rate 1×10^{-3}, batch size of 64, and epoch of 20. The setting of the `Train Whole Network` of DeepSC-MT is the same as that of `Train Semantic Codec` but with epoch of 20.

[3] https://github.com/facebookresearch/deit.

[4] https://huggingface.co/Helsinki-NLP.

The image semantic encoder of DeepSC-VQA is also based on the pre-trained DeiT-small model but the other parts are trained from scratch, where the text semantic encoder is with 6 Transformer encoder layers and the semantic decoder is with 4 Transformer encoder layers and decoder layers. We freeze the image semantic encoder to speed up training. The output dimension for the vision Transformer and text Transformer are set differently, which requires the dimension increasing operations after the image JSC decoder. The dimension-increasing operations successively include the dropout layer, dense layer from 384 to 512, ELU activation layer, dropout layer, and dense layer from 512 to 512, and ELU activation layer. The setting of the `Train Semantic Codec` of DeepSC-VQA is the Adam optimizer with learning rate 1×10^{-4}, betas of 0.9 and 0.98, batch size of 64, and epoch of 80. The setting of the `Train JSC Codec` of DeepSC-VQA is the Adam optimizer with learning rate 1×10^{-3}, batch size of 128, and epoch of 30. The setting of `Train Whole Network` of DeepSC-MT is the same as that of the `Train Semantic Codec` but with epoch of 10. The data augmentation is used to resize images to 224 × 224 with BICUBIC interpolation for both training and testing.

4.5.1.3 Benchmarks and Performance Metrics

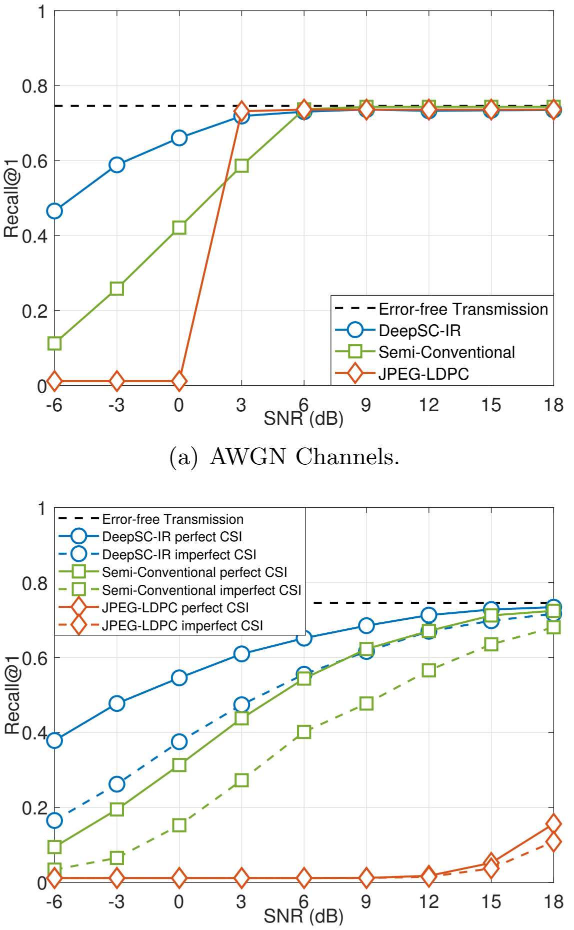
Our benchmark will adopt several typical source and channel coding methods.

- Error-free transmission: The full, noiseless images and texts are delivered to the receiver, which will serve as the upper bound.
- Semi-conventional method: Transmit the semantic information, which is extracted by semantic encoder, by conventional separate source-channel coding, we use the following technologies, respectively:
  - 8-bit μ-law quantization for mapping semantic information into bits;
  - Low-density parity-check code (LDPC) for channel coding.
  - 8-QAM for modulation.

- Conventional methods: To perform the source and channel coding separately, we use the following technologies, respectively:
    - 8-bit unicode transformation format (UTF-8) encoding for text source coding, a commonly used method in text compression;
    - Joint photographic experts group (JEPG) for image source coding, a widely used method in image compression;
    - Turbo coding for text channel coding, popular channel coding for a small size file;
    - LDPC for image channel coding, and classic channel coding for big size files.
- Hybrid methods: Transmit multimodal data by using conventional methods and proposed DeepSC together. We use the following technologies, respectively:
    - UTF-8 and Turbo coding for text transmission and DeepSC-VQA for image transmission;
    - JPEG and LDPC coding for image transmission and DeepSC-VQA for text transmission.

In the simulation, the coding rate is 1/3 and the block length is 256 for the Turbo codes. The LDPC codes employ DVB-S.2 standard. Specifically, the coding rate is 1/3, the size of parity-check matrix is $43,200 \times 64,800$, and the block length is 64,800. We employ the LS channel estimation, in which perfect and imperfect CSI are considered with $\sigma_e^2 = 0$ and $\sigma_e^2 = \sigma_n^2$, respectively. We set $r = 2$ for Rician channels and $\mathbf{H} = \mathbf{I}$ for AWGN channels. The coherent time is set as the transmission time for each batch in the simulation. We set $M = K = 2$ for metrics versus SNRs and $M = K > 2$ for metrics versus different number of users.

The Recall@1 evaluation metric [140] is adopted as performance metric for the image retrieval task, which is the ratio of the number of correct retrieval top-1 images and the number of all related images. BLEU score is adopted for the machine translation task [128] by comparing the n-grams words between the predicted sentence and the reference sentence. Answer accuracy is used for VQA task to compute the ratio between the number of correct answers and the number of all generated answers.



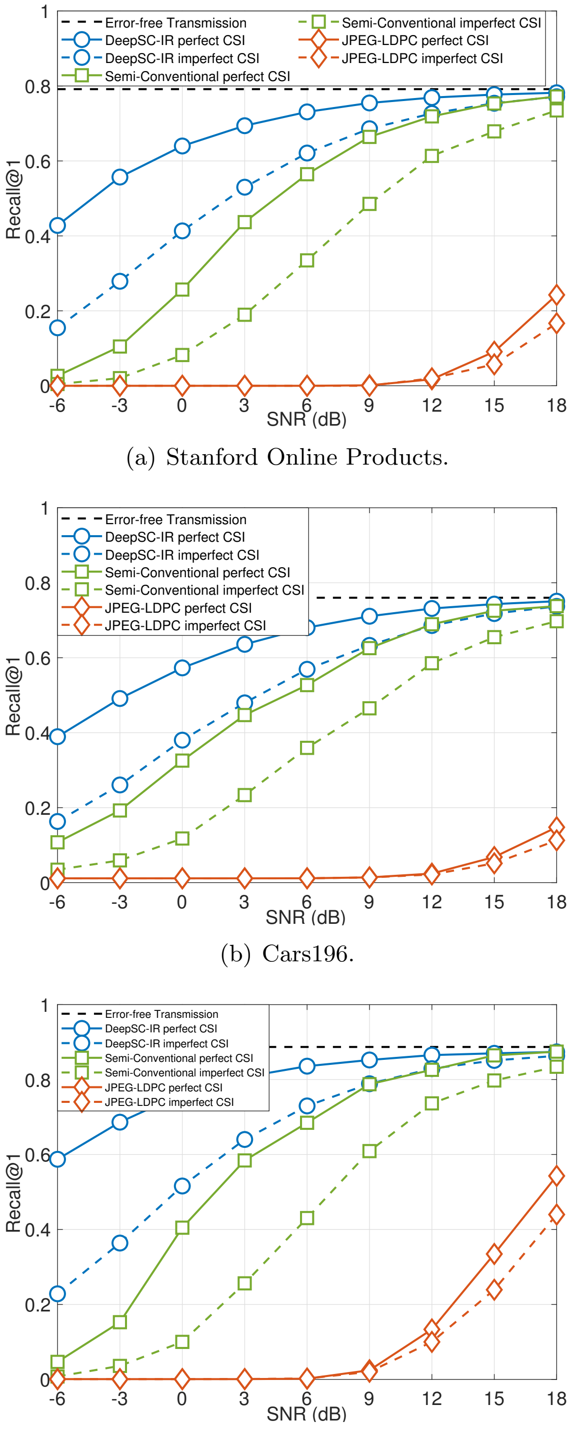
(a) AWGN Channels.

(b) Rician Channels.

Figure 4.5: Recall@1 comparison between DeepSC-IR and JPEG-LDPC with 8-QAM over different channels, in which the dataset is CUB-200-2011.

4.5.2 Single-Modal Multi-User Semantic Communication

The Recall@1 performance comparison for different channels on CUB-200-2011 and for different datasets over Rician channels are shown in Fig. 4.5 and Fig. 4.6, respectively. From Fig. 4.5, for different channels on CUB-200-2011, the proposed DeepSC-IR provides a significant gain at the low SNR regimes and approaches to the upper bound at the

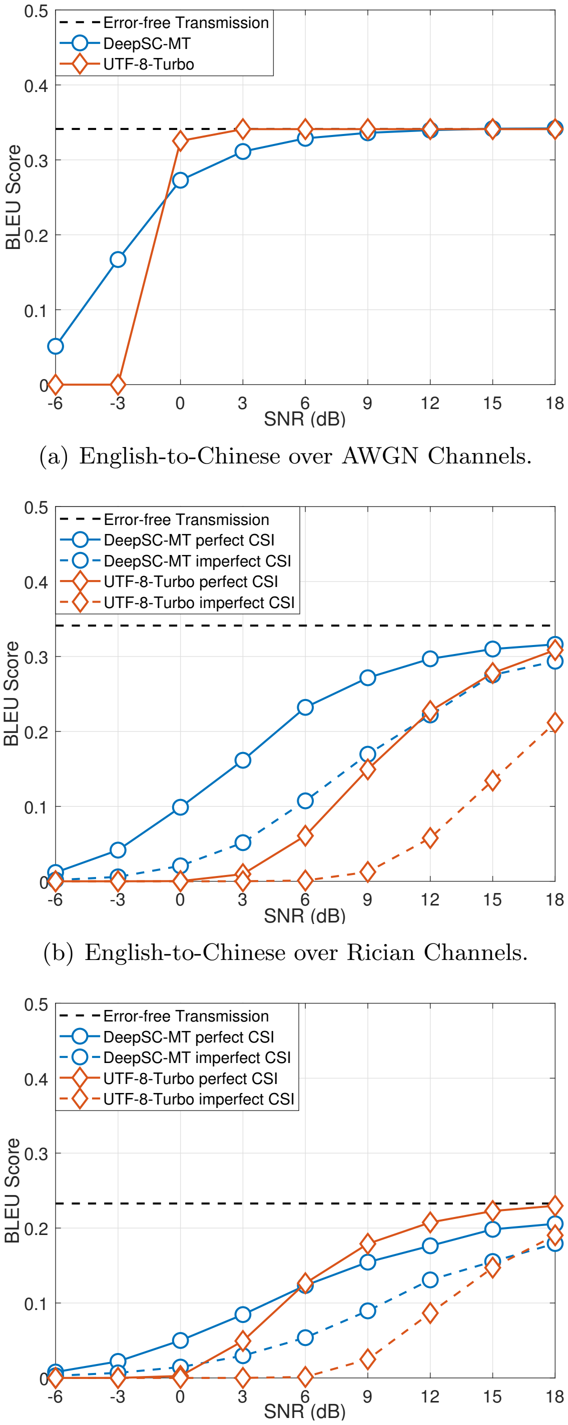


(a) Stanford Online Products.

(b) Cars196.

(c) In-shop Clothes.

Figure 4.6: Recall@1 comparison between DeepSC-IR and JPEG-LDPC with 8-QAM for different datasets under Rician channels.



(a) English-to-Chinese over AWGN Channels.

(b) English-to-Chinese over Rician Channels.

(c) Chinese-to-English over Rician Channels.

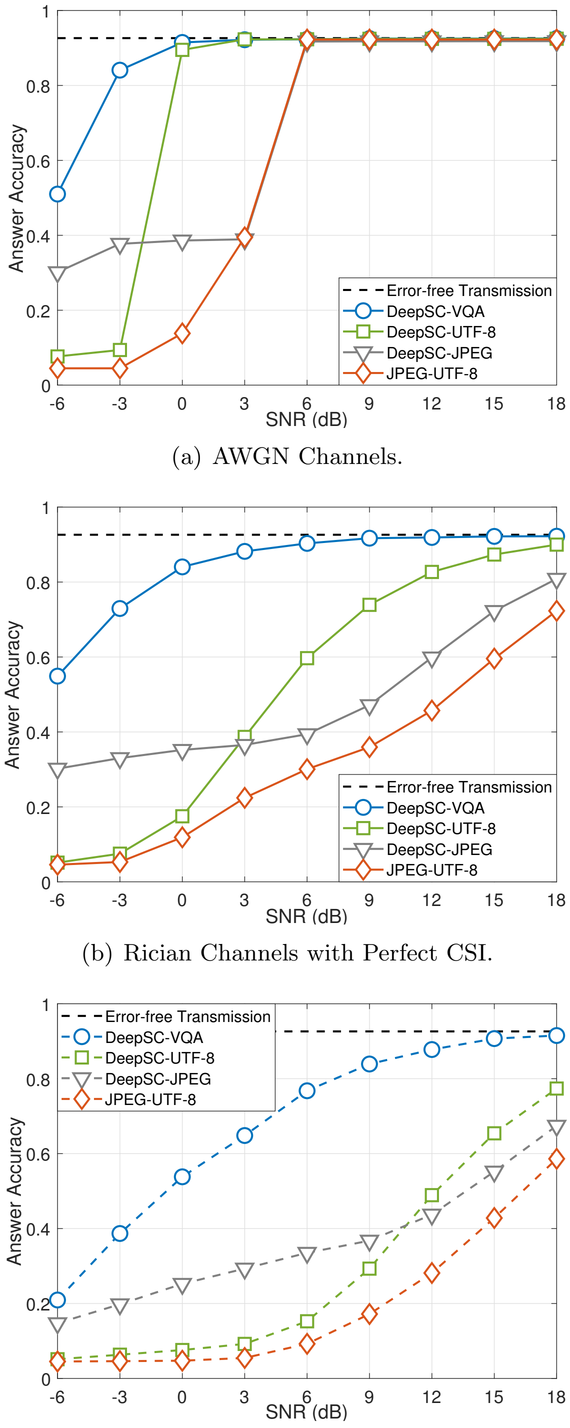
Figure 4.7: BLEU score comparison between DeepSC-MT and UTF-8-Turbo with QPSK under AWGN channels and Rician channels.

high SNR regimes among the reported methods, outperforming the JPEG-LDPC with 8-QAM by a margin of more than 24 dB gain for 0.4 Recall@1 over fading channels. From Fig. 4.6, for different datasets over Rician channels, the DeepSC-IR also outperforms the JPEG-LDPC with 8-QAM in the three popular datasets at Recall@1 with more than 24 dB gain, respectively. In both figures, transmitting semantic information by μ-law quantization and LDPC also provides the higher Recall@1 at all SNR regimes compared to JPEG-LDPC, which validates the robustness of semantic information to the noise. In addition, replacing the μ-law quantization and LDPC with DL-enabled JSC, a significant improvement to the Recall@1 at the low SNR regimes appears. This suggests that the DL-enabled JSC can further improve the robustness to noise. Besides, exploiting dynamically imperfect CSI considerably decreases the performance at Recall@1 but still outperforms the benchmarks.

The BLEU score performance comparison for different channels on English-to-Chinese and on Chinese-to-English is reported in Fig. 4.7. From Fig. 4.7, on English-to-Chinese over different channels, the DeepSC-MT outperforms the UTF-8-Turbo with QPSK at the low SNR regimes over AWGN, as well as at all SNR regimes over fading channels. More inaccurate CSI decreases BLEU score for both systems, in which the DeepSC-MT outperforms the benchmark and retains its high robustness to imperfect CSI. On Chinese-to-English over Rician channels, the DeepSC-MT performs well except at the high SNR regimes. Although the UTF-8-Turbo in BSPK has a higher BLEU score than DeepSC-MT as SNR increases, it performs worse than DeepSC-MT at all SNR regimes w.r.t. imperfect CSI.

4.5.3 Multimodal Multi-User Semantic Communication

The answer accuracy performance comparison for VQA task over different channels is presented in Fig. 4.8, in which the benchmark consists of the conventional method with UTF-8-Turbo with BPSK for text and JPEG-LDPC with 8-QAM for image and two hybrid methods. The DeepSC-VQA outperforms the benchmark at the low SNR regimes

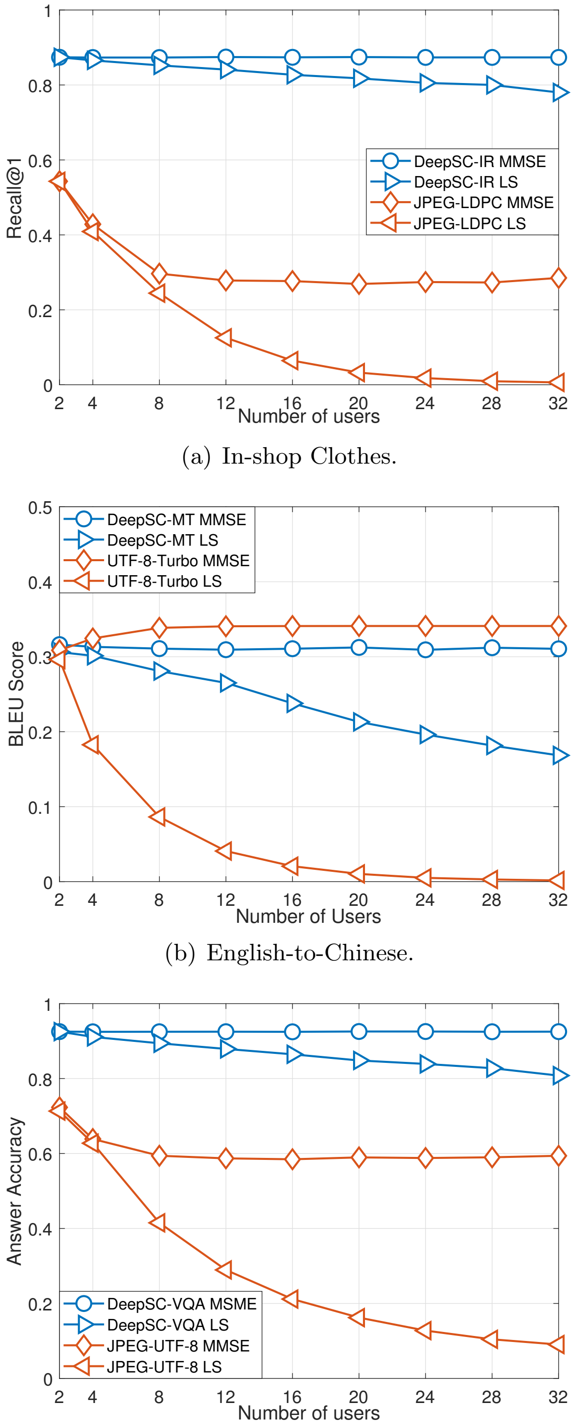


(a) AWGN Channels.

(b) Rician Channels with Perfect CSI.

(c) Rician Channels with Imperfect CSI.

Figure 4.8: Answer accuracy comparison between DeepSC-VQA, conventional methods, and hybrid methods, in which different channels are considered.



(a) In-shop Clothes.

(b) English-to-Chinese.

(c) VQA.

Figure 4.9: Recall@1, BLEU score, and answer accuracy comparisons versus the number of users over Rician channel with SNR = 18dB.

over the AWGN channels and at all SNR regimes over fading channels. In particular, the DeepSC-VQA achieves the upper bound at approximate SNR = 9dB over fading channels. The answer accuracy considerably decreases from the AWGN to fading channels for benchmarks but experiences only little performance degradation at the low SNR regimes and no performance loss at the high SNR regimes for DeepSC-VQA. Besides, replacing one of the conventional methods (i.e., UTF-8 or JPEG) with DeepSC-VQA achieves a higher answer accuracy in all SNR regimes than JPEG-UTF-8. This suggests that employing semantic information can improve the robustness of multimodal data transmission. For dynamic imperfect CSI in Fig. 4.8(c), the robustness of DeepSC-VQA also outperforms all benchmarks and is better than that of JPEG-UTF-8 with 16dB gain at 0.6 answer accuracy. Similarly, the transmitting semantic information can also improve the answer accuracy for dynamic imperfect CSI compared to JPEG-UTF-8.

Table 4-A shows the answer accuracy comparison between the classic Transformer and the layer-wise Transformer trained with 50 epochs. From Table 4-A, the layer-wise Transformer with proposed fusion method outperforms the classic Transformer with standard fusion method by 37.4% in terms of the answer accuracy. This also verifies the effectiveness of DeepSC-VQA.

Table 4-A: Answer accuracy comparison between the layer-wise Transformer based multimodal fusion method and the classic Transformer based standard fusion method.

Classic Transformer with classic fusion	Classic Transformer with proposed fusion
55.1%	57.3%
Layer-wise Transformer with classic fusion	Layer-wise Transformer with proposed fusion
92.5%	92.3%

4.5.4 Different Number of Users

In Fig. 4.9, different tasks versus the different number of users with MMSE detector and LS detector are compared, in which perfect CSI is employed. For the MMSE detector, all proposed methods perform steadily as the number of users increases but the

Table 4-B: The number of transmitted symbols comparison between multi-user semantic communication systems and traditional source-channel communication systems.

Task	Dataset	Methods	Average Number of Transmitted Symbols for One Image or One Word	Ratio
Image Retrieval	Cars196	DeepSC-IR / JPEG-LDPC with 8-QAM	128/499,920	0.02%
	CUB-200-2011		128/247,312	0.05%
	In-Shop Clothes		128/60,696	0.21%
	Stanford Online Products		128/174,808	0.07%
Machine Translation	English-to-Chinese	DeepSC-MT / UTF-8-Turbo with QPSK	77/76	101.31%
	Chinese-to-English	DeepSC-MT / UTF-8-Turbo with BPSK	77/68	113.23%
VQA	CLEVR: Text	DeepSC-VQA / UTF-8-Turbo with BPSK	77/152	50.66%
	CLEVR: Image	DeepSC-VQA / JPEG-LDPC with 8-QAM	25,216/55,624	45.33%

Table 4-C: Computational complexity comparison between multi-user semantic communication systems and traditional source-channel communication systems.

Task	Dataset	Methods	Computational Complexity	
			Additions	Multiplications
Image Retrieval	Cars196	DeepSC-IR / JPEG-LDPC with 8-QAM	$8.2 \times 10^5/9.0 \times 10^9$	$8.2 \times 10^5/1.7 \times 10^{10}$
	CUB-200-2011		$8.2 \times 10^5/4.4 \times 10^9$	$8.2 \times 10^5/8.4 \times 10^9$
	In-Shop Clothes		$8.2 \times 10^5/1.0 \times 10^9$	$8.2 \times 10^5/2.1 \times 10^9$
	Stanford Online Products		$8.2 \times 10^5/3.1 \times 10^9$	$8.2 \times 10^5/6.0 \times 10^9$
Machine Translation	English-to-Chinese	DeepSC-MT / UTF-8-Turbo with QPSK	$5.9 \times 10^5/1.0 \times 10^5$	$5.9 \times 10^5/1.6 \times 10^5$
	Chinese-to-English	DeepSC-MT / UTF-8-Turbo with BPSK	$5.9 \times 10^5/4.5 \times 10^4$	$5.9 \times 10^5/7.3 \times 10^4$
VQA	CLEVR: Text	DeepSC-VQA / UTF-8-Turbo with BPSK	$5.9 \times 10^5/1.0 \times 10^5$	$5.9 \times 10^5/1.6 \times 10^5$
	CLEVR: Image	DeepSC-VQA / JPEG-LDPC with 8-QAM	$1.6 \times 10^8/1.0 \times 10^9$	$1.6 \times 10^8/1.9 \times 10^9$

benchmarks experience performance improvement or degradation. The difference in performance trends between benchmarks are because of the gains from channel coding and low-order modulation methods. Both for image retrieval task and VQA task, the DeepSC-IR and DeepSC-VQA outperform their benchmarks at Recall@1 and at answer accuracy, respectively, in which the performance at Recall@1 and answer accuracy of benchmarks decrease first and achieve floor as the number of users increases. For the machine translation task, the BLEU score of the benchmark increases with the number of users, making the benchmark outperform DeepSC-MT with respect to perfect CSI. The performance floor appears in the MMSE detector is because the MMSE has capable of reducing the multi-access interference [141].

For the LS detector, the Recall@1, BLEU score, and answer accuracy of all methods decrease as the number of users increases due to the inter-user interference, in which all proposed semantic communication systems outperform the benchmarks with a little bit performance degradation. This indicates that the proposed semantic communication systems show high robustness to inter-user interference than that of the benchmarks.

4.5.5 Number of Transmitted Symbols

The numbers of transmission symbols for different methods are compared in Table 4-B. For image transmission, the proposed multi-user semantic communication systems significantly decrease the number of transmission symbols, especially for the image retrieval task with the DeepSC-IR only transmitting 0.02% symbols of the benchmarks for one image. For text transmission, although the proposed methods transmit a similar or slightly more number of symbols compared with the benchmark in machine translation task, they achieve approximately 50% saving in the numbers of symbols when the benchmark employs a lower order modulation in the VQA task. This suggests that the proposed multi-user semantic communications can decrease the transmission delay with a lower number of transmission symbols and hence are suitable for lower latency scenarios.

4.5.6 Computational Complexity

The computational complexity for different methods is compared in Table 4-C. We only analyze the complexity of channel coding for both methods because the other parts are shared in both methods and the complexity of source coding is low and can be omitted. The computational complexity of the proposed method mainly depends on the matrix multiplication. For the traditional method, we mainly calculate the decode complex per bit, which mainly follows the computation method shown in [142]. For image transmission, all of the proposed methods have a lower computational complexity than traditional methods, in which the complexity of DeepSC-IR can decrease by more than one order of magnitude. For text transmission, the proposed DeepSC-MT shows a similar computational complexity in English transmission but has a slightly higher computational complexity in the Chinese transmission compared to the benchmarks. This suggests that the proposed multi-user semantic communication systems achieve lower power consumption when transmitting a large size of data.

4.6 Summary

In this chapter, we have explored task-oriented multi-user semantic communications to transmit data with single-modality and multiple modalities, respectively. We considered two single-modal tasks, image retrieval and machine translation, as well as one multimodal task, VQA. In this context, we have proposed three Transformer based transceivers, DeepSC-IR, DeepSC-MT, and DeepSC-VQA, which share the same transmitter structures but with different receiver structures. Each transceiver is trained jointly by the proposed training algorithm. In addition, all of the proposed multi-user semantic communication systems were found to outperform the traditional ones in the low SNR regimes and provide graceful performance degradation with imperfect CSI. For both image retrieval and VQA tasks, the proposed DeepSC-IR and DeepSC-VQA can provide more than 18 dB gain and reduce by more than 50% the number of transmission symbols and computational complexity compared to traditional communications. In particular, compared with traditional methods, DeepSC-IR only needs 1‰ transmission symbols on average and decreases the complexity by more than one order of magnitude. As a result, we conclude that multi-user semantic communication systems are an attractive alternative to traditional communication systems for particular tasks. In the next chapter, we will investigate the semantic communication with memory to perform the memory tasks.

Chapter 5

Semantic Communications with Memory

In this chapter, the contributions are introduced in Section 5.1. The system model is introduced in Section 5.2. The semantic communication system with memory module is proposed in Section 5.3. Section 5.4 details the proposed dynamic transmission methods. Numerical results are presented in Section 5.5 to show the performance of the proposed frameworks. Finally, Section 5.6 concludes this chapter.

5.1 Introduction

Compared the human communication system [143], the existing semantic communication systems miss an important module, *memory*. In general, memory can be categorized into short-term memory to enable scenario conversations and long-term memory to help humans train their thinking. Introducing the memory module to semantic communications will enable the system to execute not only memoryless tasks but also tasks with memory. Memoryless tasks are only related to the current input, e.g., the aforementioned tasks, while tasks with memory are related to both the current input and the past inputs, e.g., scenario question answer and scenario conversations. Considering the memory module, the communication between machines and human-to-machine will be more intelligent and human-like. For the design of semantic communication systems with memory, we face the following challenges:

Q1: *How to design the semantic-aware transceiver with memory module?*

Q2: *How to ensure the effectiveness of transmitting memory over multiple time slots?*

For the above questions, we investigate the task-oriented semantic communication for memory tasks by using the scenario question answer task as an example. We develop a DL enabled semantic communication system with memory (Mem-DeepSC) to address the aforementioned challenges. The main contributions of this chapter are summarized as follows:

- Based on the universal Transformer [99], a transceiver with a memory module is proposed. In the proposed Mem-DeepSC, the transmitter can extract the semantic features at the sentence level effectively and the receiver can process received semantic features from the previous time-slots by employing the memory module, which addresses the aforementioned *Q1*.
- To make the Mem-DeepSC applicable to various SNRs, the relationship between the length of semantic signal and channel noise between semantic noise and channel noise is derived. Especially, two dynamic transmission methods are proposed to preserve semantic features from distortion and reduce the communication resources. Two lower bounds of mutual information are derived to train the dynamic transmission methods. This addresses the aforementioned *Q2*.

5.2 System Model

As shown in Fig. 5.1, we consider a single-input single-output (SISO) communication system, which is with one antenna at the transmitter and one at the receiver. The transceiver has three modules, a semantic codec to extract the semantic features of the source and perform the task, a JSC codec to compress and recover the semantic features over the channels, and the memory module to store the context from different time slots and help the semantic decoder to perform the task. We focus on the text scenario question answer, therefore, the transmission can be categorized into two phases: 1) memory shaping to transmit the context to the receiver via multiple time slots; 2) task execution to transmit the question to the receiver to get the answer.

5.2.1 Memory Shaping

Assume the k-th sentence is transmitted at the k-th time slot and denote $\boldsymbol{s}^c$ and $\boldsymbol{s}^q$ as the context sentence and question sentence, respectively. In the memory shaping phase, the transmitter sends the context, e.g., multiple sentences, images, or speeches, to the receiver over multiple time slots. Then, with the semantic encoder and channel encoder, the k-th context sentence over the k-th time slot can be encoded as

$$\boldsymbol{x}_k^c = C\left(S\left(\boldsymbol{s}_k^c; \boldsymbol{\alpha}\right); \boldsymbol{\beta}\right), \tag{5.1}$$

where $\boldsymbol{x}_k^c$ is the transmitted signals after the power normalization, $S\left(\cdot; \boldsymbol{\alpha}\right)$ and $C\left(\cdot; \boldsymbol{\beta}\right)$ are denoted as the semantic encoder with parameter $\boldsymbol{\alpha}$ and channel encoder with parameter $\boldsymbol{\beta}$, respectively.

Transmitting the signals over the channels, the received signal can be presented as

$$\boldsymbol{y}_k^c = \boldsymbol{h} \odot \boldsymbol{x}_k^c + \boldsymbol{n}, \tag{5.2}$$

where $\boldsymbol{h}$ is the channel coefficients and $\boldsymbol{n}$ is the AWGN, in which $\boldsymbol{n} \sim \mathcal{CN}\left(0, \sigma_n^2 \mathbf{I}_L\right)$. For the Rayleigh fading channel, the channel coefficient follows $\boldsymbol{h} \sim \mathcal{CN}\left(0, \mathbf{I}_L\right)$; for the Rician fading channel, it follows $\boldsymbol{h} \sim \mathcal{CN}\left(\mu_h \mathbf{I}_{L\times 1}, \sigma_h^2 \mathbf{I}_L\right)$ with $\mu_h = \sqrt{r/(r+1)}$ and $\sigma_h = \sqrt{1/(r+1)}$, where r is the Rician coefficient. The SNR is defined as $\mathbb{E}(\|\boldsymbol{h} \odot \boldsymbol{x}_k^c\|^2)/\mathbb{E}(\|\boldsymbol{n}\|^2)$.

With the estimated CSI, $\hat{\boldsymbol{h}}$, the transmitted signals, $\hat{\boldsymbol{x}}_k$, can be detected by

$$\hat{\boldsymbol{x}}_k^c = \hat{\boldsymbol{h}}^{\mathtt{H}} \odot \boldsymbol{y}_k^c \oslash \left(\hat{\boldsymbol{h}} \odot \hat{\boldsymbol{h}}^{\mathtt{H}}\right). \tag{5.3}$$

After signal detection, the semantic features can be recovered by

$$\hat{\boldsymbol{z}}_k^c = C^{-1}\left(\hat{\boldsymbol{x}}_k^c; \boldsymbol{\gamma}\right), \tag{5.4}$$

where $C^{-1}\left(\cdot; \boldsymbol{\gamma}\right)$ is denoted as the channel decoder with parameter $\boldsymbol{\gamma}$. Then, the recovered semantic features will be inputted into the memory module.

We model the memory module with the concept of short-term memory as the queue with length K. The memory module at the k-th time slot is represented by

$$\mathbf{M}^{(k)} = [\hat{\boldsymbol{z}}_{k-K+1}^c, \hat{\boldsymbol{z}}_{k-K+2}^c, \cdots, \hat{\boldsymbol{z}}_k^c]. \tag{5.5}$$

From (5.5), the memory queue is updated with the incoming received latest semantic features and pop the oldest features out of the queue.

5.2.2 Task Execution

In the task execution phase, the transmitter sends the question sentence, $\boldsymbol{s}^q$, to the receiver to perform the task. Specially, $\boldsymbol{s}^q$ is encoded into $\boldsymbol{x}^q$ by (5.1), transmitted over the air, and decoded into $\hat{\boldsymbol{z}}^q$ by (5.4). In the scenario question answer task, the question is not only related to only one context sentence but also multiple context sentences. Therefore, the answer is predicted with the question and memory together, which is represented as

$$\hat{a} = S^{-1}\left(\left[\hat{\boldsymbol{z}}^q, \mathbf{M}^{(k)}\right]; \boldsymbol{\varphi}\right), \tag{5.6}$$

where $S^{-1}(\cdot; \boldsymbol{\varphi})$ is the semantic decoder with parameters $\boldsymbol{\varphi}$.

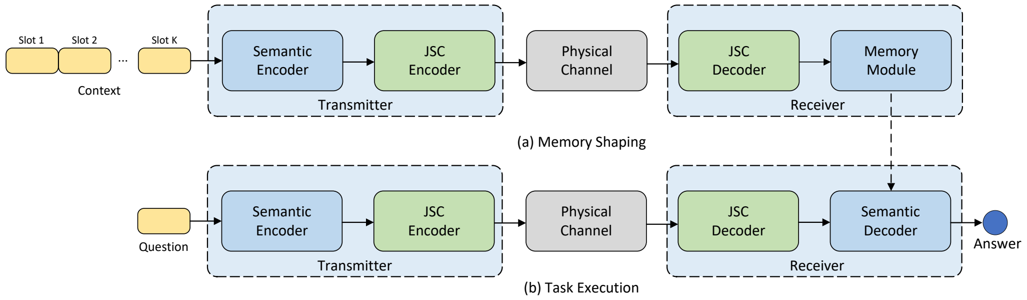


Figure 5.1: The proposed framework for memory semantic communication systems.

5.3 Semantic Communication System with Memory

In this section, we design a semantic communication system with memory, named Mem-DeepSC, to perform scenario question answer task, in which the universal Transformer is employed for text understanding.

5.3.1 Model Description

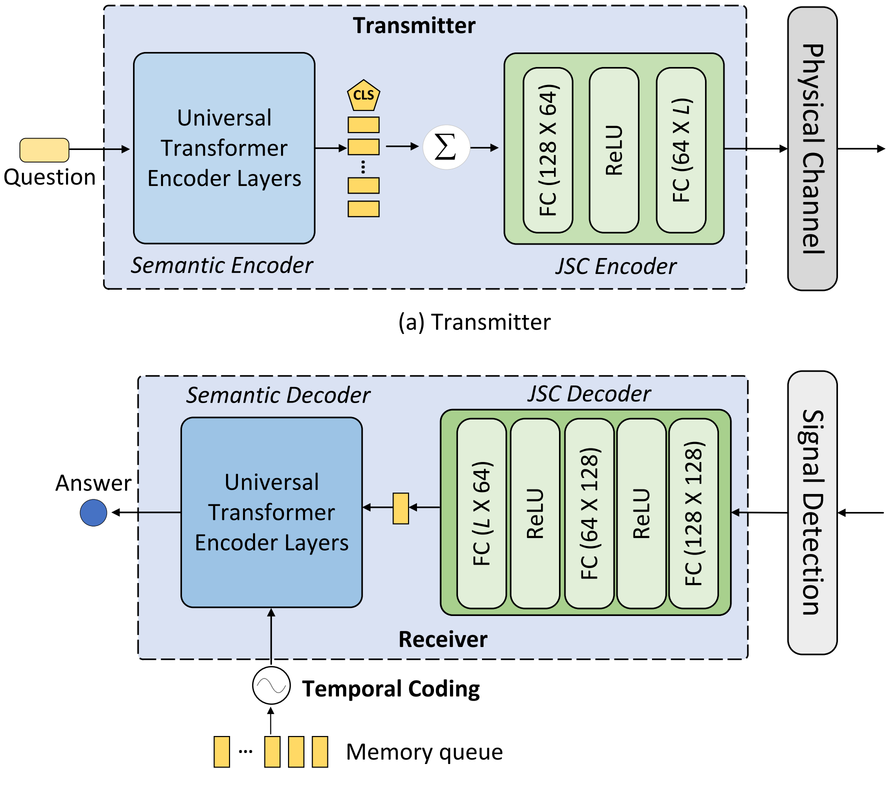
The proposed Mem-DeepSC is shown in Fig. 5.2. The semantic encoder consists of universal Transformer encoder layer with variable steps to extract the semantic feature of each word. In order to reduce the transmission overheads, *the summation operation* is taken here, in which these semantic features at the word level are merged to get one semantic feature at the sentence level. The reason that we choose universal Transformer in the semantic codec can be summarized as follows:

1. The universal Transformer can be trained and tested much faster than the architectures based on recurrent layers due to the parallel computation [99].

2. Compared with the classic Transformer, the universal Transformer shares the parameters, which can reduce the model size.

With the sentence semantic feature, the JSC encoder employs multiple dense layers to compress the sentence semantic feature. The reasons that we mainly use dense layer in the channel codec can be summarized as follows:

1. The universal Transformer consists of dense layers to capture the text features.



(a) Transmitter

(b) Receiver

Figure 5.2: The proposed Mem-DeepSC.

The use of dense is consistent with the design of the universal Transformer.

2. The JSC codec aims to recover the whole semantic features. Compared with the CNN layer to capture the local information, the dense layer is good at capturing the global information and preserving the entire attributes, which follows the target of the JSC codec.

At the receiver, the JSC decoder correspondingly includes multiple dense layers to decompress sentence semantic feature and reduce the distortion from channels. The semantic decoder also contains the universal Transformer encoder layer with variable steps to find the relationship between the memory queue and the query feature to get the answer. Especially, the memory queue does not contain the temporal information inside. Therefore, *temporal coding* is employed to add temporal information to the memory queue, in which we adopt the positional coding [62] as the temporal coding.

5.3.2 Training Details

As shown in Algorithm 5.1, the training of Mem-DeepSC includes three steps, which is similar to the training algorithm proposed in [89]. The first step is to train the semantic codec. In order to improve the accuracy of answers, we choose the CE as the loss function instead of the answer accuracy. The cross-entropy is given by

$$\mathcal{L}_{\texttt{CE}} = -p\left(a\right)\log\left(p\left(\hat{a}\right)\right), \tag{5.7}$$

where $p\left(a\right)$ is the real probability of answer and $p\left(\hat{a}\right)$ is the predicted probability. After convergence, the model learns to extract the semantic features and predict the answers. The following proposition proved in Appendix A reveals the relationship between cross-entropy and the answer accuracy.

Proposition 1. Cross entropy loss function is the refined function of answer accuracy and is more stable during training.

With the trained semantic codec, the second step is to ensure the semantic features transmitted over the air effectively. Thus, the JSC codec is trained to learn the compression and decompression of the semantic features as well as to deal with the channel distortion with the MSE loss function,

$$\mathcal{L}_{\texttt{MSE}} = \left\|\boldsymbol{z}_k^c - \hat{\boldsymbol{z}}_k^c\right\|^2, \tag{5.8}$$

where $\boldsymbol{z}_k^c$ and $\hat{\boldsymbol{z}}_k^c$ are the original semantic features and the recovered semantic features, respectively.

Finally, the third step is to optimize the entire system jointly to achieve the global optimization. The semantic codec and JSC codec are trained jointly with the CE loss function to reduce the error propagation between each modules.

Algorithm 5.1: *Mem-DeepSC Training Algorithm.*

```
1: Function: Train Semantic Codec{}
2:   Choose $\{(\boldsymbol{s}_1^c, \boldsymbol{s}_2^c, \cdots, \boldsymbol{s}_K^c), \boldsymbol{s}^q, a\}$ from dataset.
3:   For $k = 1 \rightarrow K$
4:     $S(\boldsymbol{s}_k^c; \boldsymbol{\alpha}) \rightarrow \mathbf{W}_k^c$,
5:     Take the summation operation, $\boldsymbol{z}_k^c = \sum_j \mathbf{W}_{k,j}^c$,
6:   End For
7:   $S(\boldsymbol{s}^q; \boldsymbol{\alpha}) \rightarrow \mathbf{W}^q$, and $\boldsymbol{z}^q = \sum_j \mathbf{W}_j^q$,
8:   Form the memory queue $\mathbf{M}^{(K)}$ by (5.5),
9:   Take the temporal coding for $\mathbf{M}^{(K)}$,
10:  $S^{-1}\left(\left[\boldsymbol{z}^q, \mathbf{M}^{(K)}\right]; \boldsymbol{\varphi}\right) \rightarrow \hat{a}$,
11:  Compute CE loss with $a$ and $\hat{a}$.
12:  Train $\boldsymbol{\alpha}, \boldsymbol{\varphi} \rightarrow$ Gradient descent with CE loss.
13: Return $S(\cdot; \boldsymbol{\alpha})$ and $S^{-1}(\cdot; \boldsymbol{\varphi})$.

1: Function: Train JSC Codec
2:   Choose Semantic features $\boldsymbol{z}_k^c$.
3:   Transmitter:
4:     $C(\boldsymbol{z}_k^c; \boldsymbol{\beta}) \rightarrow \boldsymbol{x}_k^c$,
5:     Power Normalization,
6:     Transmit $\boldsymbol{x}_k^c$ over the air.
7:   Receiver:
8:     Receive $\boldsymbol{y}_k^c$,
9:     Signal detection by (5.3) to get $\hat{\boldsymbol{x}}_k^c$,
10:    $C^{-1}(\hat{\boldsymbol{x}}_k^c; \boldsymbol{\gamma}) \rightarrow \hat{\boldsymbol{z}}_k^c$,
11:  Compute MSE loss with $\boldsymbol{z}_k^c$ and $\hat{\boldsymbol{z}}_k^c$.
12:  Train $\boldsymbol{\beta}, \boldsymbol{\gamma} \rightarrow$ Gradient descent with MSE loss.
13: Return: $C(\cdot; \boldsymbol{\beta})$ and $C^{-1}(\cdot; \boldsymbol{\gamma})$.

1: Function: Train Whole Network
2:   Choose $\{(\boldsymbol{s}_1^c, \boldsymbol{s}_2^c, \cdots, \boldsymbol{s}_K^c), \boldsymbol{s}^q, a\}$ from dataset.
3:   Repeat line 2-5, 12-19, and 6-8 to get $\hat{a}$,
4:   Compute CE loss with $\hat{a}$ and $a$.
5:   Train $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\varphi} \rightarrow$ Gradient descent with CE loss.
6: Return: $S(\cdot; \boldsymbol{\alpha})$, $S^{-1}(\cdot; \boldsymbol{\varphi})$, $C(\cdot; \boldsymbol{\beta})$, and $C^{-1}(\cdot; \boldsymbol{\gamma})$.
```

With the Mem-DeepSC, the memory-related tasks can be performed. However, the context is transmitted via multiple time slots. If each time slot has different channel conditions, the damage to the semantic information is inevitable at the worse channel conditions, which affects the prediction accuracy. Therefore, in order to preserve the semantic information and save the communication overheads over multiple time slots, we further develop an adaptive rate transmission method.

5.4 Adaptive Rate Transmission

In this section, we derive the relationship between the length of semantic signal and channel noise, which inspires us to transmit different length signals according to SNRs. We develop two dynamic transmission methods, importance mask and consecutive mask for saving the communication resources and preventing the outage for memory transmission to different SNRs.

5.4.1 The Relationship between the Length of Semantic Signal and Channel Noise

Adaptive modulation has been developed for conventional communications [10], where the modulation order and code rate change according to SNRs. The same spirit can be used in semantic communications if there exists the relationship between the length of semantic signal and channel noise. In this situation, we can achieve such adaptive operation by masking some elements, i.e., masking less at low SNR regimes to ensure the reliability of performing tasks and masking more elements at high SNR regimes to achieve a higher transmission rate.

How many semantic elements should be transmitted? The existing works [77, 144] employ neural networks to learn how to determine the number of transmitted semantic elements dynamically, which lacks of interpretability. Therefore, we provide a theoretical analysis of semantic-aware channel capacity to guide us to determine the number of semantic elements at certain SNR.

The key is to find the relationship between the noise level and the number of elements that can be transmitted correctly. Firstly, we model $\boldsymbol{x}_k^c$ into

$$\boldsymbol{x}_k^c = \boldsymbol{r}_k^c + \boldsymbol{n}_{\texttt{model}}, \tag{5.9}$$

where $\boldsymbol{r}_k^c$ is the semantic information selected from the latent semantic codewords, $\boldsymbol{n}_{\texttt{model}} \sim \mathcal{CN}\left(0, \sigma_m^2 \mathbf{I}\right)$ is the model noise. We generally initialize the model weights with Gaussian distribution and apply the batch normalization/layer normalization to normalize the outputs following $\mathcal{N}(0,1)$ [145]. Therefore, we model the model noise with Gaussian distribution. In deep learning, the model noise is caused by the unstable gradients descending, the training data noise, and so on. The model noise can be alleviated by the larger dataset, the refined optimizer, and the re-designed loss function but cannot be removed.

Assume the length of $\boldsymbol{x}_k^c$ is L. By applying the packing sphere theory [146], $\boldsymbol{x}_k^c$ can be mapped to the L-dimension sphere space as shown in Fig. 5.3(a). In the Fig. 5.3(a), the smaller sphere represents the noise sphere with radius $\sqrt{L}\sigma_m$ and the larger sphere is the signal sphere with radius $\sqrt{L(\mu_{\max}^2+\sigma_m^2)}$, where $\mu_{\max}$ is the maximum value in the latent semantic codewords. The reason that noise spheres spread the signal sphere is that the latent semantic codewords have different constellation points. Communication is reliable as long as the noise spheres do not overlap. Therefore, there exists a minimum length of L to prevent the overlap from the model noise. In other words, the number of semantic codewords that can be packed with non-overlapping noise sphere over the model noise is

$$N = \frac{\left(\sqrt{L\left(\mu_{\max}^2+\sigma_m^2\right)}\right)^L}{\left(\sqrt{L\left(\sigma_m^2\right)}\right)^L} = \left(1+\frac{\mu_{\max}^2}{\sigma_m^2}\right)^{\frac{L}{2}}. \tag{5.10}$$

After transmitting $\boldsymbol{x}_k^c$ over the AWGN channels, the received signals can be represented by submitting (5.9) into (5.2),

$$\boldsymbol{y}_k^c = \boldsymbol{r}_k^c + \boldsymbol{n}_{\texttt{model}} + \boldsymbol{n}_{\text{channel}}, \tag{5.11}$$

where $\boldsymbol{n}$ in (5.2) is re-denoted to $\boldsymbol{n}_{\text{channel}}$. The $\boldsymbol{y}_k^c$ can also be mapped to the L-dimension sphere space shown in Fig. 5.3(b). Because of the channel noise, the radius of noise sphere increases from $\sqrt{L(\sigma_m^2)}$ to $\sqrt{L(\sigma_n^2+\sigma_m^2)}$, which makes the noise spheres overlap.

Since the channel noise and model noise are independent, we can view the $\mathbf{n}_{\texttt{channel}}$

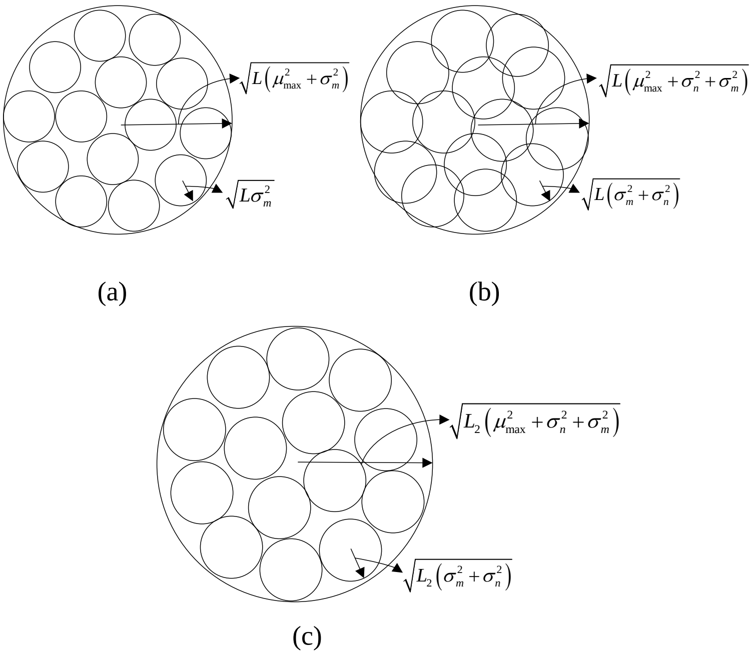

$\sqrt{L\left(\mu_{\max}^2+\sigma_m^2\right)}$
$\sqrt{L\sigma_m^2}$
(a)
$\sqrt{L\left(\mu_{\max}^2+\sigma_n^2+\sigma_m^2\right)}$
$\sqrt{L\left(\sigma_m^2+\sigma_n^2\right)}$
(b)
$\sqrt{L_2\left(\mu_{\max}^2+\sigma_n^2+\sigma_m^2\right)}$
$\sqrt{L_2\left(\sigma_m^2+\sigma_n^2\right)}$
(c)

Figure 5.3: The example of semantic-aware channel capacity.

and $\mathbf{n}_{\texttt{model}}$ as new channel noise, $\tilde{\mathbf{n}}_{\texttt{channel}} = \mathbf{n}_{\texttt{channel}} + \mathbf{n}_{\texttt{model}}$, which is given by

$$\boldsymbol{y}_k^c = \boldsymbol{r}_k^c + \tilde{\mathbf{n}}_{\texttt{channel}}. \tag{5.12}$$

Similar to the traditional wireless communication, we can capture the relation between the length of the signal and the new channel noise, i.e., the adaptive rate transmission scheme, by using Shannon's theory. In order to eliminate the overlapping, one way is to increase the length of $\boldsymbol{x}_k^c$ from L to L_1 to enlarge the volume of the signal sphere so that the enlarged noise spheres do not overlap. Then, the number of semantic codewords that can be packed with non-overlapping noise sphere over the model noise and the channel noise is

$$N = \frac{\left(\sqrt{L_1\left(\mu_{\max}^2+\sigma_m^2+\sigma_n^2\right)}\right)^{L_1}}{\left(\sqrt{L_1\left(\sigma_m^2+\sigma_n^2\right)}\right)^{L_1}} = \left(1+\frac{\mu_{\max}^2}{\sigma_m^2+\sigma_n^2}\right)^{\frac{L_1}{2}}. \tag{5.13}$$

The semantic codewords only describe the semantic information of the source and are unrelated to the channel noise, which means that the numbers of semantic codewords in (5.10) and (5.13) are the same. Therefore, the relationship between L and L_1 can be derived as shown in proposition 2.

Proposition 2. Given the minimum length L to prevent from model noise, the minimum length for reliable communication over AWGN channels is

$$L_1 = L \times \frac{\log\left(1 + \frac{\mu_{\max}^2}{\sigma_m^2}\right)}{\log\left(1 + \frac{\mu_{\max}^2}{\sigma_m^2 + \sigma_n^2}\right)}. \tag{5.14}$$

With proposition 2, the masked ratio to different SNRs can be computed theoretically.

5.4.1.1 Asymptotic Analysis

With (5.14), the asymptotic analysis can be derived into four cases listed below.

- **Case 1**: When $\sigma_n^2 \to 0$, then $L_1 \to L$. The number of transmitted symbols will converge to minimum L. In this case, the semantic communication system can be viewed as the compressor and decompressor.

- **Case 2**: When $\sigma_n^2 \to \infty$, then $L_1 \to \infty$. The number of transmitted symbols will lead to infinity. In this case, the semantic communication system experiences an outage.

- **Case 3**: When $\sigma_m^2 \to 0$, then $L \to 0$. L_1 only depends on the channel noise and can be computed by
$$L_1 = \frac{2\log(N)}{\log\left(1 + \frac{\mu_{\max}^2}{\sigma_n^2}\right)}. \tag{5.15}$$
In this situation, L_1 is computed by the traditional channel capacity and the number of semantic codewords. In this case, the relationship between the length of semantic signal and channel noise is the same as the traditional channel capacity.

- **Case 4**: When $\sigma_m^2 \to \infty$, then $L \to \infty$. The semantic communication system experiences an outage, similar to case 2.

The key differences between semantic-aware channel capacity and traditional channel capacity can be summarized in the following,

1. The relationship between the length of semantic signal and channel noise indicates how much semantic information can be transmitted error-free while the traditional channel capacity indicates how many bits can be transmitted error-free.

2. The relationship between the length of semantic signal and channel noise is affected by three points, 1) the number of semantic codewords, 2) the model noise, and 3) the channel noise. But the channel capacity only depends on the channel noise.

3. When channel noise disappears, the relationship between the length of semantic signal and channel noise has the lower bound, L. The traditional channel capacity does not have such a lower bound.

With the relationship between the length of semantic signal and channel noise, it is possible to achieve dynamic transmission. The key to achieving such a dynamic transmission in semantic communication systems is to identify which elements are more important than the others and mask the unimportant ones. For different noise levels, we can adjust the length of the semantic signal according to the proposed relationship between the length of the semantic signal and the new channel noise, in which the length of the semantic signal is decided by the number of neurons in the neural networks. In other words, we can adjust the number of neurons, i.e., the neural network architecture, based on the new channel noise. In this chapter, we propose the dynamic neural network to achieve the different number of neurons for different noise levels. As shown in Fig. 5.4, we propose two mask methods subsequently, importance mask method and consecutive mask method.

5.4.2 Importance Mask

As shown in Fig. 5.4(a), the importance mask method introduces the importance-aware model to identify the importance order among the elements of $\boldsymbol{x}_k^c$, which can be expressed as

$$\boldsymbol{r}_k^c = F\left(\boldsymbol{x}_k^c; \boldsymbol{\theta}\right), \tag{5.16}$$

where $F\left(\cdot; \boldsymbol{\theta}\right)$ is the importance-aware model with learnable parameter $\boldsymbol{\theta}$, $\boldsymbol{r}_k^c$ is the importance rank of $\boldsymbol{x}_k^c$, in which the bigger value means that the corresponding element is more important.

By setting the threshold, γ, the mask, $\boldsymbol{m}_k^c$, can be computed with the $\boldsymbol{r}_k^c$ by

$$m_{k,i}^c = \begin{cases} 1, \, r_{k,i}^c > \gamma, \\ 0, \, r_{k,i}^c \leq \gamma. \end{cases} \tag{5.17}$$

Then, the masked transmitted signal can be generated by

$$\tilde{\boldsymbol{x}}_k^c = \boldsymbol{x}_k^c \odot \boldsymbol{m}_k^c. \tag{5.18}$$

With $\tilde{\boldsymbol{x}}_k^c$, the transmitter can send the only non-zero elements and the position information of zero elements to reduce the communication overheads.

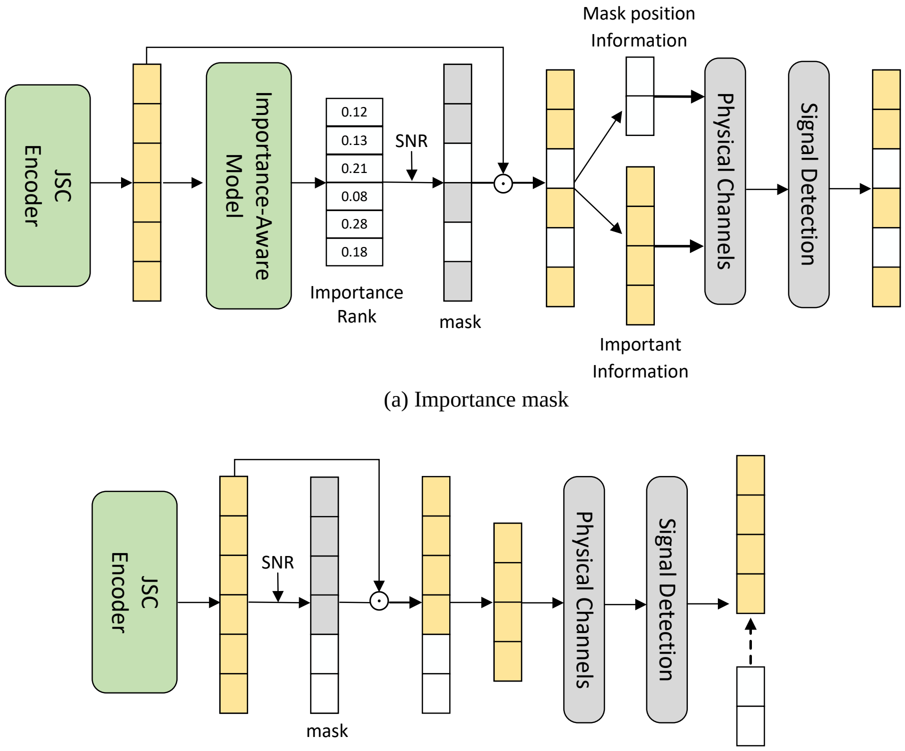
After transmitting $\tilde{\boldsymbol{x}}_k^c$ over the air, the receiver follows the same processing to perform signal detection, JSC decoding, and semantic decoding.

5.4.2.1 Loss Function Design

In order to train the importance model, the optimization goal is to keep more information related the task in the masked signals to prevent performance degradation. Therefore, the mutual information between $\tilde{\boldsymbol{x}}_k^c$ and the goal a is employed as the loss function,

$$\mathcal{L}_{\mathtt{MI}} = -I\left(\tilde{\boldsymbol{x}}_k^c; a\right). \tag{5.19}$$

However, minimizing (5.19) with gradients descending algorithm is hard since $\mathcal{L}_{\mathtt{MI}}$



(a) Importance mask

(b) Consecutive mask

Figure 5.4: Two proposed dynamic transmission methods.

is undifferentiable and difficult to compute. There are several methods to alleviate the problem, e.g., employing the mutual information estimator and the numerical approximation. Even if these methods solve the undifferentiable problem, it is still unstable in estimating the mutual information. In order to achieve stable optimization, an approximate bound-optimization (or Majorize-Minimize) algorithm is employed. The bound-optimization aims to construct the desired majorized/minorized version of the objective function. Following the idea, two propositions are proposed for the bound-optimization of mutual information, which are proved in Appendices B and C, respectively.

Proposition 3. For classification tasks, alternately maximizing the mutual information can be viewed as a bound optimization of the cross entropy [147].

Proposition 4. For regression tasks, alternately maximizing the mutual information can be viewed as a bound optimization of the mean absolute error [147].

With Propositions 3 and 4, the mutual information loss function in (5.19) can be changed to the cross-entropy loss function in (5.7).

5.4.2.2 Training Details

As shown in Algorithm 5.2, the importance model is trained by the CE loss function and the frozen Mem-DeepSC model. The training importance model takes the backpropagations from the semantic decoder to guide the importance model, in which the SoftKMax activation function is employed to bridge the backpropagation from mask to importance model. In other words, the importance model can learn which elements have more contributions/importance to the task performance by minimizing the CE loss function.

5.4.3 Consecutive Mask

As shown in Fig. 5.4(b), the consecutive mask method masks the last consecutive elements in the $\boldsymbol{x}_k^c$ to zero, so that the transmitter only sends the non-zero elements and the receiver pads the received signals with zeros to the same length of $\boldsymbol{x}_k^c$. The consecutive mask method does not need to transmit the additional mask position information but to re-train the Mem-DeepSC model. Since the importance rank of the elements of $\boldsymbol{x}_k^c$ is not consecutive, directly masking these consecutive elements may experience performance degradation. The Mem-DeepSC needs to be re-trained with the consecutive mask so that it can learn to re-organize the elements of $\boldsymbol{x}_k^c$ following the order of decreasing importance.

The training of the consecutive mask method only includes one step, which is similar to the `Train Whole Network` in Algorithm 5.1 but with two additional operations, i.e., masking operation before transmitting and padding operation after signal detection. The loss function during the training is the CE loss function.

Algorithm 5.2: *Importance Mask Training Algorithm.*

```
Function: Train Importance Model
    Choose $\{\boldsymbol{x}_k^c, \boldsymbol{x}^q, a\}$. Freeze Mem-DeepSC.
    Transmitter:
        $F\left(\boldsymbol{x}_k^c; \boldsymbol{\theta}\right) \to \boldsymbol{r}_k^c$,
        Compute the mask, $\boldsymbol{m}_k^c$, by (5.17)
        Compute the mask signal, $\tilde{\boldsymbol{x}}_k^c$, by (5.18),
        Transmit $\tilde{\boldsymbol{x}}_k^c$ and $\boldsymbol{x}^q$ over the air,
    Receiver:
        Receive signal and perform signal detection,
        $C^{-1}\left(\tilde{\boldsymbol{x}}_k^c; \boldsymbol{\gamma}\right) \to \hat{\boldsymbol{z}}_k^c$, and $C^{-1}\left(\hat{\boldsymbol{x}}^q; \boldsymbol{\gamma}\right) \to \hat{\boldsymbol{z}}^q$
        Update the memory queue, $\mathbf{M}^{(k)}$, with $\hat{\boldsymbol{z}}_k^c$,
        Take the temporal coding for $\mathbf{M}^{(k)}$,
        $S^{-1}\left(\left[\hat{\boldsymbol{z}}^q, \mathbf{M}^{(k)}\right]; \boldsymbol{\varphi}\right) \to \hat{a}$,
    Compute CE loss with $a$ and $\hat{a}$.
    Train $\boldsymbol{\theta} \to$ Gradient descent with CE loss.
Return: $F\left(\cdot; \boldsymbol{\theta}\right)$.
```

5.5 Simulation Results

In this section, we compare the proposed semantic communication systems with memory with the traditional source coding and channel coding method over various channels, in which the proposed mask methods are compared with different benchmarks.

5.5.1 Implementation Details

5.5.1.1 The Dataset

We choose the *bAbI-10k* dataset [148], including 20 different types of scenario tasks. Each example is composed of a set of facts, a question, the answer, and the supporting facts that lead to the answer. We split the 10k examples into 8k examples for training, 1k examples for validation, and 1k examples for testing.

5.5.1.2 Traing Settings

The semantic encoder and decoder consist of the universal Transformer encoder layer with 3 steps and with 6 steps, respectively, in which the width of the layer is 128. The importance model is composed of one Transformer encoder layer with the width of 64.

The other training settings are listed in Table 5-A.

5.5.1.3 Benchmarks and Performance Metrics

We adopt the typical source and channel coding method as the benchmark of the proposed Mem-DeepSC, and the random mask method as the counterpart of the proposed two mask methods.

- Separate Mem-DeepSC: The semantic codec and channel codec are trained separately.
- Conventional methods: To perform the source and channel coding separately, we use the following technologies, respectively:
  - UTF-8 encoding for text source coding, a commonly used method in text compression;
  - Turbo coding for text channel coding, popular channel coding for a small size file;
  - 16-QAM as the modulation.
- Random Mask: Mask the elements in the transmitted signal randomly.

In the simulation, the coding rate is 1/3 and the block length is 256 for the Turbo codes. The coherent time is set as the transmission time for each context in the simulation. We set $r = 2$ for the Rician channels and $\mathbf{h} = \mathbf{1}$ for the AWGN channels. In order to compute the relationship between the length of semantic signal and channel noise, we train multiple Mem-DeepSC with different sizes to find the values of $\mu_{\max}$ and σ_m^2. For Mem-DeepSC, $\mu_{\max} = 1$ and $\sigma_m^2 = 1.44$. Answer accuracy is used as the metric to compute the ratio between the number of correct answers and that of all generated answers.

Table 5-A: The Training Settings.

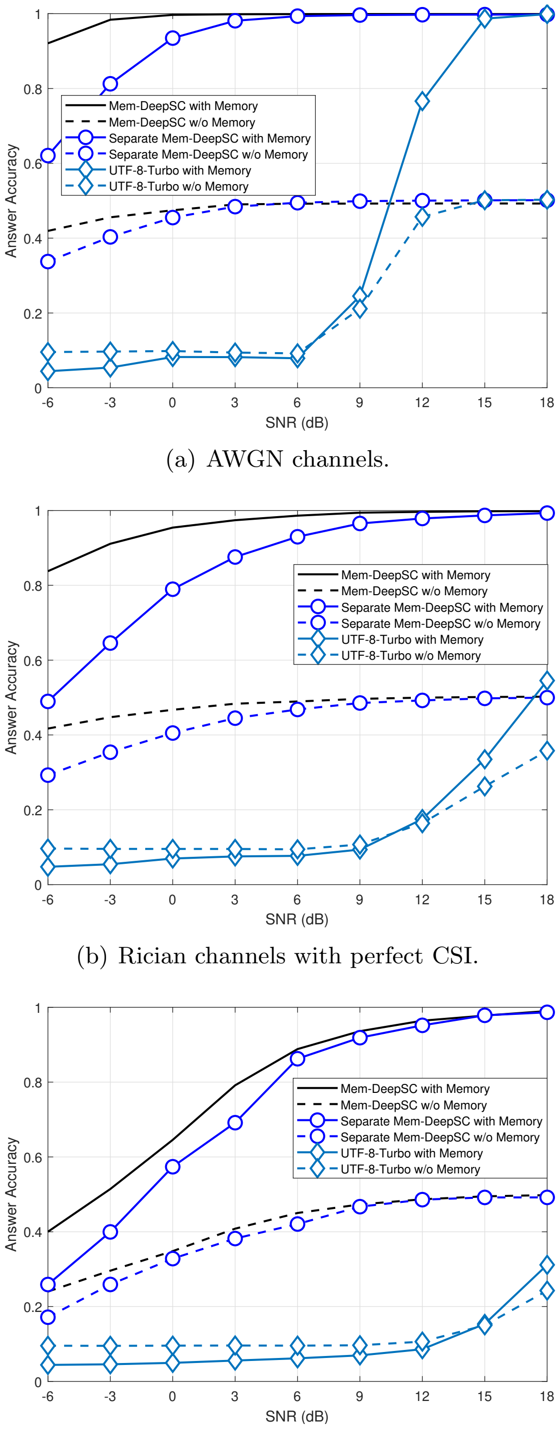
	Batch Size	Learning Rate	Epoch
`Train Semantic Codec`	200	5×10^{-4}	250
`Train Channel Codec`	100	1×10^{-4}	50
`Train Whole Network`	200	5×10^{-4}	30
`Train Importance Mask`	200	5×10^{-4}	10
`Train Consecutive Mask`	200	1×10^{-4}	50

5.5.2 Memory Semantic Communication Systems

Fig. 5.5 compares the answer accuracies over different channels, in which the Mem-DeepSC and the UTF-8-Turbo transmit 32 symbols per sentence and 190 symbols per sentence, respectively. The proposed Mem-DeepSC with memory outperforms all the benchmarks at the answer accuracy in all SNR regimes by the margin of 0.8. Compared the Mem-DeepSC with memory and without memory, the memory module can significantly improve the answer accuracy, which validates the effectiveness of the memory module in memory-related transmission tasks. Besides, the Mem-DeepSC outperforms the separate Mem-DeepSC in low SNR regimes, which means that the three stage training algorithm can help improve the robustness to channel noise. From the AWGN channels to the Rician channels, the proposed Mem-DeepSC with memory experiences slight answer accuracy degradation in the low SNR regimes but the UTF-8-Turbo has an obvious performance loss in all SNR regimes. The inaccurate CSI deteriorates the answer accuracy for both methods, however, the proposed Mem-DeepSC can keep a similar answer accuracy in high SNR regimes, which shows the robustness of the proposed Mem-DeepSC.

5.5.3 The Proposed Mask Methods

Table 5-B compares the number of transmitted symbols for different methods. Compared to the UTF-8-Turbo with the adaptive modulation and channel coding (AMC), the proposed Mem-DeepSC decreases the number of the transmitted symbols significantly with only 4%-16.8% symbols. The reason is that the Mem-DeepSC transmits the semantic

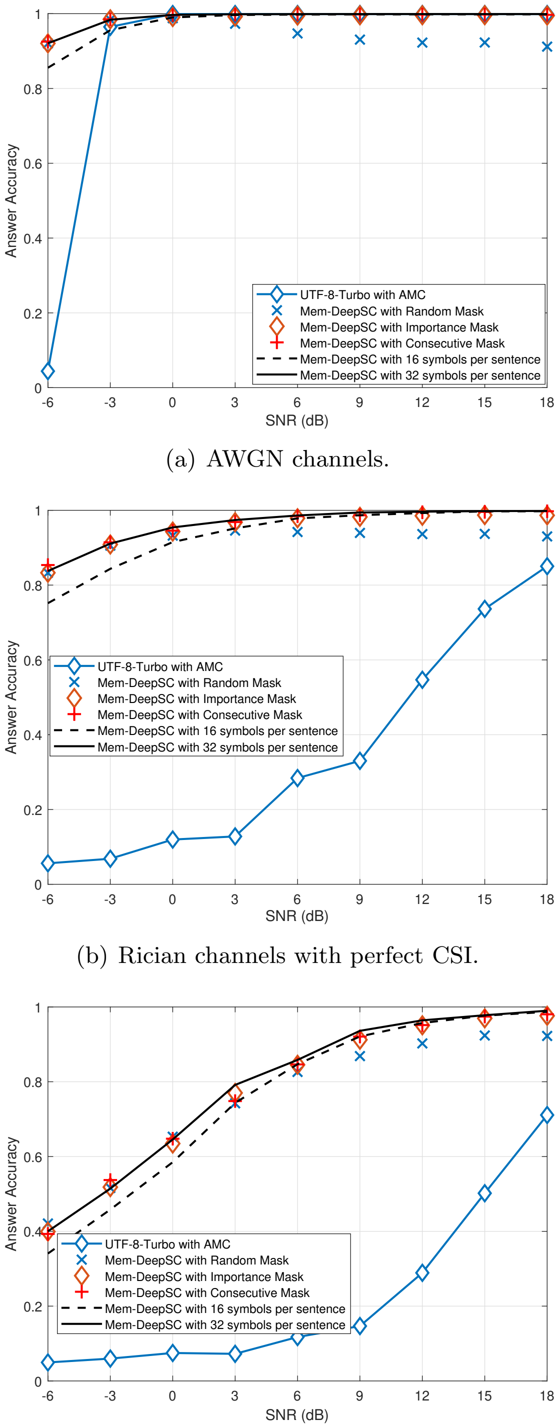


(a) AWGN channels.

(b) Rician channels with perfect CSI.

(c) Rician channels with imperfect CSI.

Figure 5.5: Answer accuracy comparison between Mem-DeepSC and UTF-8-Turbo with 16-QAM over different channels.



(a) AWGN channels.

(b) Rician channels with perfect CSI.

(c) Rician channels with imperfect CSI.

Figure 5.6: Answer accuracy comparison between Mem-DeepSC for different number of transmitted symbols over different channels.

Table 5-B: The number of transmitted symbols comparison between different methods.

	Number of Transmitted Symbols				
	-6dB	0dB	6dB	12dB	18dB
Mem-DeepSC	32				
Dynamic Transmission	32	25	18	16	16
UTF-8-Turbo	190				
UTF-8-Turbo with AMC (AWGN Channels)	760 (BPSK)	760 (BPSK)	380 (4QAM)	253 (8QAM)	190 (16QAM)
UTF-8-Turbo with AMC (Rician Fading Channels)	760 (BPSK)	760 (BPSK)	380 (4QAM)	253 (8QAM)	253 (8QAM)

information at the sentence level instead of at the letter/word level. Besides, applying the dynamic methods can further reduce the number of transmitted symbols from 32 symbols to 16 symbols per sentence as the SNR increases, especially saving an additional 50% symbols in the high SNR regimes. Then, the effectiveness of (5.14) is validated by the following simulation in Fig. 5.6.

Fig. 5.6 verifies the effectiveness of the proposed mask strategy. For Mem-DeepSC with no mask, we provided two cases with 16 symbols and 32 symbols per sentence, respectively. Utilizing adaptive modulation and channel coding (AMC) on UTF-8-Turbo can yield comparable answer accuracy to that of Mem-DeepSC over AWGN channels. However, this comes at the expense of a reduced transmission rate. Then comprising the no mask cases with different numbers of symbols per sentence, increasing the number of symbols per sentence leads to higher answer accuracy in low SNR regimes but the gain disappears as the SNR increases. This suggested that the semantic communication systems can employ more symbols in low SNR to improve the robustness and transmit fewer symbols in the high SNR regimes to improve transmission efficiency. The proposed importance mask and consecutive mask keep a similar answer accuracy as the Mem-DeepSC with 32 symbols per sentence in all SNR regimes over the AWGN and the Rician channels.

5.6 Summary

In this chapter, we have proposed a semantic communication system with memory, named Mem-DeepSC, for the scenario question answer task. The Mem-DeepSC can extract the semantic information at the sentence level to reduce the number of the transmitted symbols and deal with the context information at the receiver by introducing the memory queue. Moreover, with the memory module, the Mem-DeepSC can deal with the memory-related tasks compared to that without the memory module, which is closer to human-like communication. Besides, the relationship between the length of semantic signal and channel noise is derived to decide how many symbols are required to be transmitted at different SNRs. Two dynamic transmission methods are proposed to mask the unimportant elements in the transmitted signals, which can employ more symbols in the low SNR to improve the robustness and transmit fewer symbols in the high SNR regimes to improve the transmission efficiency. In particular, the dynamic transmission methods can save an additional 50% transmitted symbols. Therefore, the semantic communication system with memory is an attractive alternative to intelligent communication systems. In the next chapter, we will investigate the low-complexity semantic communication for IoT devices.

Chapter 6

Low-Complexity Semantic Communication Systems

In this chapter, the contributions are introduced in Section 6.1. The distributed semantic communication system model is introduced and the corresponding problems are identified in Section 6.2. Section 6.3 presents the proposed L-DeepSC. Numerical results are used to verify the performance of the proposed L-DeepSC in Section 6.4. Finally, Section 6.5 concludes this chapter.

6.1 Introduction

The rapid development of DL and widespread applications of IoT have made the devices smarter than before, and enabled them to perform more intelligent tasks. However, it is challenging for any IoT device to train and run a DL model independently due to its limited computing capability. In this chapter, we consider an IoT network where the cloud/edge platform performs the DeepSC model training and updating while IoT devices perform data collection and transmission based on the trained model. To make it affordable for IoT devices, we are facing the following three questions,

- *Question 1: How to design semantic communication systems over wireless fading channels?*
- *Question 2: How to form the constellation to make it affordable for capacity-limited IoT devices?*
- *Question 3: How to compress semantic models for fast-model transmission and low-cost implementation on IoT devices?*

For the above questions, we propose a lite distributed semantic communication system based on DL, named L-DeepSC, for text transmission with low complexity, where the data transmission from the IoT devices to the cloud/edge works at the semantic level to improve transmission efficiency. The main contributions of this chapter are summarized as follows.

- We design a distributed semantic communication network under power and latency constraints, in which the receiver and feature extractor networks are jointly optimized by overcoming fading channels.
- By identifying the impacts of CSI on DL model training over fading channels, we propose a CSI-aided semantic communication system to speed up convergence, where the CSI is refined by a de-noise neural network. This addresses the aforementioned *Question 1*.
- To make data transmission and receiving affordable for capacity constrained devices, we design a finite-bits constellation to solve *Question 2*.
- Due to over-parametrization, we propose a model compression algorithm, including network sparsification and quantization, to reduce the size of DL models by pruning the redundancy connections and quantizing the weights, which addresses the aforementioned *Question 3*.

6.2 System Model and Problem Formulation

Text is an important type of source data, which can be sensed from speaking and typing, environmental monitoring, etc. By training DL models with these text data at cloud/edge platform, the DL models based IoT devices have the capability to understand text data and generate semantic feature to be transmitted to the center to perform intelligent tasks, i.e., intelligent assistants, human emotion understanding, and environment humid and temperature adjustment based on human preference [149].

As shown in Fig. 6.1(a), we focus on distributed semantic communications for IoT networks. The considered system is consisted of various IoT networks with two layers, the cloud/edge platform and distributed IoT devices. The cloud/edge platform is equipped with huge computation power and big memory, which can be used to train the DL model by the received semantic features. The semantic communication enabled IoT devices to perform intelligent tasks by understanding sensed texts, which are with limited memory and power but expected long lifetime, i.e., up to 10 years. Particularly, our considered distributed semantic communication system consists of the following three steps:

1) **Model Initialization/Update**: The cloud/edge platform first trains the semantic communication model by initial dataset. The trained model is updated in the subsequent iterations by the received semantic features from IoT devices.

2) **Model Broadcasting**: The cloud/edge platform broadcasts the trained DL model to each IoT device.

3) **Semantic Features Upload**: The IoT devices constantly capture the text data, which are encoded by the proposed semantic transmitter shown in Fig. 6.1(b). The extracted semantic features are then transmitted to the cloud/edge for model update and subsequent processing.

The aforementioned *Questions 1-3* correspond to model initialization/update, semantic features uploading, and model broadcasting, respectively. Different from the traditional information transmission, semantic features can be not only used for recovering the text at the semantic level accurately, but also exploited as the input of other modules, i.e., emotion classification, dialog system, and human-robot interaction, for training effect networks and perform various intelligent tasks directly. The devices can also exchange semantic features, which has been previously discussed in our work in Chapter 2. We focus on the communication between cloud/edge platforms and local IoT devices to make the semantic communication model affordable.

6.2.1 Semantic Communication System

The DeepSC shown in Fig. 6.1(b) can be divided into three parts mainly, transmitter network, physical channel, and receiver network, where the transmitter network includes semantic encoder and channel encoder, and the receiver network consists of semantic decoder and channel decoder.

We assume that the input of the DeepSC is a sentence, $\mathbf{s} = [w_1, w_2, \cdots, w_L]$, where w_l represents the l-th word in the sentence. The encoded symbol stream can be represented as

$$\mathbf{X} = C\left(S\left(\boldsymbol{s}; \boldsymbol{\alpha}\right); \boldsymbol{\beta}\right), \tag{6.1}$$

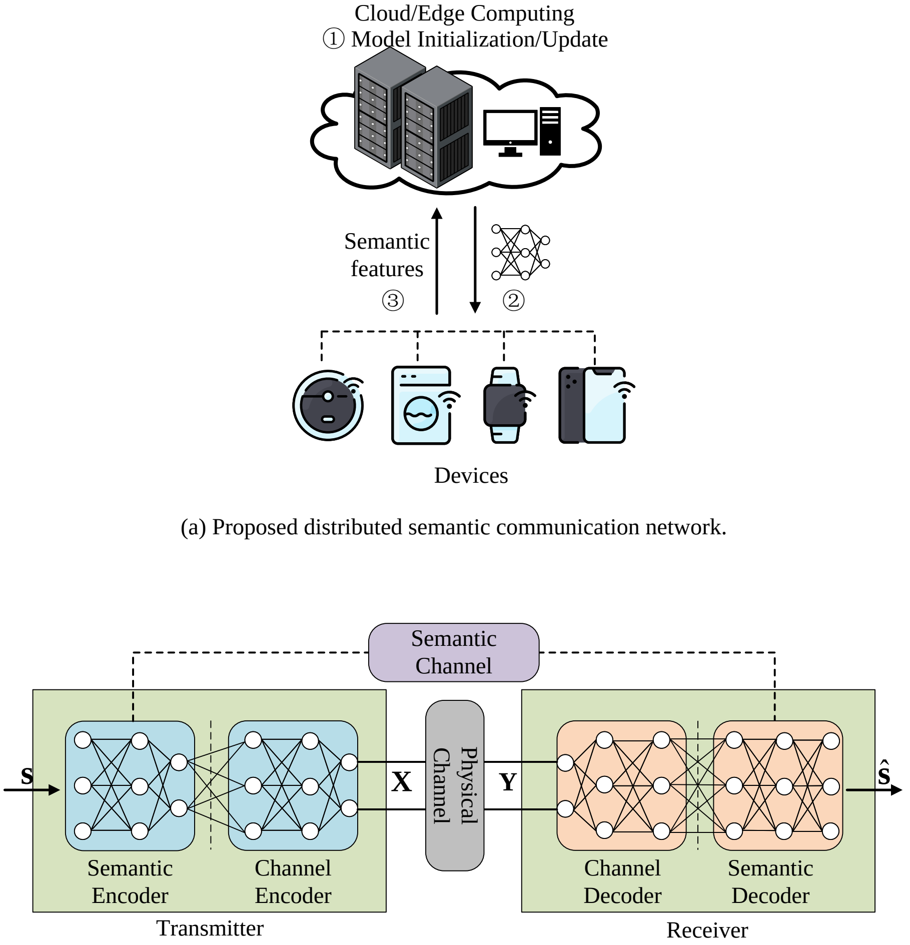
where $S\left(\cdot; \boldsymbol{\alpha}\right)$ is the semantic encoder network with parameter set $\boldsymbol{\alpha}$ and $C\left(\cdot; \boldsymbol{\beta}\right)$ is the channel encoder with parameter set $\boldsymbol{\beta}$.

If $\mathbf{X}$ is sent through a wireless fading channel, the signal received at the receiver can be given by

$$\mathbf{Y} = f_{\mathbf{H}}(\mathbf{X}) = \mathbf{H}\mathbf{X} + \mathbf{N}, \tag{6.2}$$

where $\mathbf{H}$[1] represents the channel gain between the transmitter and the receiver, and $\mathbf{N} \sim \mathcal{CN}\left(0, \sigma_n^2\right)$ is AWGN.

[1] Here, we have omitted discussion of complex channels. If the complex channel is $\bar{\mathbf{H}}$, then $\bar{\mathbf{H}} = \left[\Re\left(\mathbf{H}\right), -\Im\left(\mathbf{H}\right); \Im\left(\mathbf{H}\right), \Re\left(\mathbf{H}\right)\right]$.



(a) Proposed distributed semantic communication network.

(b) Semantic communication system

Figure 6.1: The framework of semantic communications for IoT networks.

Then, the decoded source signal can be represented as

$$\hat{\boldsymbol{s}} = S^{-1}\left(C^{-1}\left(\mathbf{Y};\boldsymbol{\gamma}\right);\boldsymbol{\varphi}\right), \tag{6.3}$$

where $\hat{\boldsymbol{s}}$ is the recovered sentence, $C^{-1}\left(\cdot;\boldsymbol{\gamma}\right)$ is the channel decoder with parameter set $\boldsymbol{\gamma}$ and $S^{-1}\left(\cdot;\boldsymbol{\varphi}\right)$ is the semantic decoder network with parameter set $\boldsymbol{\varphi}$, the superscript -1 represents the decoding operation.

The whole semantic communication can be trained by the CE loss function, which is given by

$$\mathcal{L}_{\mathtt{CE}}(\boldsymbol{s},\hat{\boldsymbol{s}}) = \sum_{l=1} \left(q\left(w_l\right) - 1\right) \log\left(1 - p\left(w_l\right)\right) - \sum_{l=1} q\left(w_l\right) \log\left(p\left(w_l\right)\right), \tag{6.4}$$

where $q(w_l)$ is the real probability that the l-th word, w_l, appears in source sentence $\boldsymbol{s}$, and $p(w_l)$ is the predicted probability that the l-th word, w_l, appears in $\hat{\boldsymbol{s}}$. CE can measure the difference between the two distributions. Through minimizing the CE loss, the network can learn the word distribution, $q(w_l)$, in the source sentence, $\boldsymbol{s}$. Consequently, the syntax, phrase, and the meaning of words in the context can be learnt by DNNs.

6.2.2 Problem Description

Instead of bits, the input sentence, $\boldsymbol{s}$, in the DeepSC, will cause that the learned constellation is no longer limited to a few points anymore. After transmitting $\mathbf{X}$, the fading channel increases the difficulty of model training compared with the AWGN channel. Meanwhile, the huge number of parameters, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}$, indicates the complexity of the whole model. These factors limit DeepSC for IoT networks and incur the aforementioned *Questions 1-3*, including feasible constellation design, training for fading channel, and model compression.

6.2.2.1 Training of fading channel

In DL, the training process can be divided forward-propagation to predict the target and back-propagation to converge the neural network, as stated in the following.

Forward-propagation: From the received signal to recover semantic information, the estimation sentence is given by

$$\hat{\boldsymbol{s}} = S^{-1}\left(C^{-1}\left(\mathbf{Y};\gamma\right);\boldsymbol{\varphi}\right), \tag{6.5}$$

Back-propagation: Taking semantic encoder as an example, the parameter vector at the t_{th} iteration are is updated by

$$\boldsymbol{\alpha}(t) = \boldsymbol{\alpha}(t-1) - \eta \frac{\partial \mathcal{L}_{\texttt{CE}}}{\partial \boldsymbol{\alpha}}, \tag{6.6}$$

where η is the learning rate and $\frac{\partial \mathcal{L}_{\texttt{CE}}}{\partial \boldsymbol{\alpha}}$ is the gradient, computed by

$$\frac{\partial \mathcal{L}_{\texttt{CE}}}{\partial \boldsymbol{\alpha}} = \frac{\partial \mathcal{L}_{\texttt{CE}}}{\partial \hat{\boldsymbol{s}}} \frac{\partial \hat{\boldsymbol{s}}}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \boldsymbol{\alpha}} = \frac{\partial \mathcal{L}_{\texttt{CE}}}{\partial \hat{\boldsymbol{s}}} \frac{\partial \hat{\boldsymbol{s}}}{\partial \mathbf{Y}} \mathbf{H} \frac{\partial \mathbf{X}}{\partial \boldsymbol{\alpha}}. \tag{6.7}$$

In (6.7), $\mathbf{H}$ will introduce stochasticity during weight updating. For an AWGN channel, $\mathbf{H} = \mathbf{I}$ will not affect it. However, for fading channels, $\mathbf{H}$ is random, which may lead to that $\boldsymbol{\beta}$ fails to converge to the global optimum while the forward-propagation in (6.5) is unable to recover semantic information accurately based on the local optimum. Thus, it is critical to design the training process to mitigate the effects of $\mathbf{H}$, which also makes the DeepSC applicable for fading channels.

6.2.2.2 Feasible constellation design

Generally, the DL models run on floating-point operations, which means that the input, output, and weights are in a large range of $\pm 1.40129 \times 10^{-45}$ to $\pm 3.40282 \times 10^{+38}$ [150]. Although DeepSC can learn the constellations from the source information and channel statistics, the learned constellation points, such as cluster constellation [151], are disordered in the range of $\pm 1.40129 \times 10^{-45}$ to $\pm 3.40282 \times 10^{+38}$, which brings additional burden to the hardware of IoT devices, for instance, the high-resolution phase-shift and amplitude-shift pose high requirements on the circuit. Therefore, it is desired to form feasible constellations with only finite points for the current radio frequency (RF) systems. In other words, we have to design a smaller constellation for the DeepSC.

6.2.2.3 Model communication

The more parameters DeepSC has, the stronger the signal processing ability, which however increases computational complexity and model size and results in high power consumption. In the distributed DeepSC system, the trained DeepSC model deployed at local IoT devices is frequently updated to perform intelligent tasks better. The IoT application limits the bandwidth and cost of distributing the DeepSC model. Furthermore, to extend the IoT network lifetime, especially the battery lifetime, most local devices are with finite storage and computation capability, which limits the size of DeepSC. Therefore, compressing DeepSC not only reduces the latency of model transmission between the cloud/edge platform and local devices but also makes it possible to run the DL model on local devices.

6.3 Proposed Lite Distributed Semantic Communication System

To address the identified challenges in Section II, we propose a lite distributed semantic communication system, named L-DeepSC. We analyze the effects of CSI in the model training under fading channels and design a CSI-aided training process to overcome the fading effects, which successfully deals with *Question 1*. Besides, the weight pruning and quantization are investigated to address *Question 2*. Finally, our finite-points constellation design solves *Question 3*, effectively.

6.3.1 Deep De-noise Network based CSI Refinement and Cancellation

The most common method to reduce the effects of fading channels in wireless communication is to use the known channel properties of a communication link, CSI. Similarly, CSI can also reduce the channel impacts in training L-DeepSC. Next, we will first analyze the role of CSI in L-DeepSC training.

In order to simplify the analysis, we assume the transmitter and the receiver are with one-layer dense with sigmoid activation, where transmitter has an additional untrainable embedding layer, and receiver also has an untrainable de-embedding layer. The IoT devices are with the trained transmitter model and the cloud/edge platform works as the receiver, as shown in the system model Fig. 6.1. The IoT devices and cloud/edge platform are equipped with the same number of antennas. After the embedding layer, the source message, $\mathbf{s}$, is embedded into, $\mathbf{S}$. Then, encode $\mathbf{S}$ into

$$\mathbf{X} = \sigma\left(\mathbf{W}_T \mathbf{S} + \mathbf{b}_T\right), \tag{6.8}$$

where $\mathbf{X}^2$ is the semantic features transmitted from the IoT devices to the cloud/edge platform. $\mathbf{W}_T$ and $\mathbf{b}_T$ are the trainable parameters to extract the features from source message $\mathbf{s}$, and $\sigma(\cdot)$ is the sigmoid activation function.

The received symbol at the cloud/edge platform is affected by channel $\mathbf{H}$ and AWGN as in (6.2). From the received symbol, the cloud/edge platform recovers the embedding matrix by

$$\hat{\mathbf{S}} = \sigma\left(\mathbf{W}_R \mathbf{Y} + \mathbf{b}_R\right), \tag{6.9}$$

where the estimated source message, $\hat{\mathbf{s}}$, can be obtained after de-embedding layer. $\mathbf{W}_R$ and $\mathbf{b}_R$ can learn to recover $\mathbf{s}$. The L-DeepSC can be optimized by the loss function in (6.4). The fading channels not only contaminates the gradients in the back-propagation, but also restricts the representation power in the forward-propagation.

Back-propagation: It updates parameter $\mathbf{W}_T$ by its gradient

$$\frac{\partial \mathcal{L}_{\texttt{CE}}\left(\hat{\boldsymbol{s}}, \boldsymbol{s}\right)}{\partial \mathbf{W}_T} = \left(\mathbf{F}_R \mathbf{W}_R \mathbf{H} \mathbf{F}_T\right)^T \nabla_{\hat{\boldsymbol{s}}} \mathcal{L}_{\texttt{CE}}\left(\hat{\boldsymbol{s}}, \boldsymbol{s}\right) \boldsymbol{s}^T, \tag{6.10}$$

where $\mathbf{F}_R \sim \operatorname{diag}\left(\sigma'\left(\mathbf{W}_R \mathbf{y} + \mathbf{b}_R\right)\right)$ and $\mathbf{F}_T \sim \operatorname{diag}\left(\sigma'\left(\mathbf{W}_T \mathbf{s} + \mathbf{b}_T\right)\right)$. In (6.10), the $\mathbf{H}$ is untrainable and random, therefore it will cause perturbation for the weight updating, i.e., the weight updating with higher variance. If the transmitter consists of very deep neural networks, the perturbation will affect the back-propagation of the whole transmitter network, where the perturbation will propagate to the whole transmitter network by the chain rule.

[2] Here, we have avoided discussion of complex signal. If the complex signal is $\bar{\mathbf{X}}$, then $\bar{\mathbf{X}} = \left[\Re\left(\mathbf{X}\right), \Im\left(\mathbf{X}\right)\right]$.

Forward-propagation: With the received signal $\mathbf{W}_R$, the source messages can be recovered by

$$\hat{\mathbf{S}} = \sigma\left(\mathbf{W}_R\mathbf{Y} + \mathbf{b}_R\right) = \sigma\left(\mathbf{W}_R\mathbf{H}\mathbf{X} + \mathbf{W}_R\mathbf{N} + \mathbf{b}_R\right). \tag{6.11}$$

In (6.11), $\mathbf{W}_R$ has to learn how to deal with the channel effects and decode at the same time, which increases training burden and reduces network expression capability. Meanwhile, the errors caused by channel effects also propagate to the subsequent layers for the L-DeepSC receiver with multiple layers.

The impacts of channel can be mitigated by exploiting CSI at the cloud/edge. If channel $\mathbf{H}$ is known, then the received symbol can be processed by

$$\tilde{\mathbf{Y}} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{Y} \quad = \mathbf{X} + \tilde{\mathbf{N}}, \tag{6.12}$$

where $\tilde{\mathbf{N}} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{N}$. In (6.12), the channel effect is transferred from multiplicative noise to additive noise, $\tilde{\mathbf{N}}$, which provides the possibility of stable back-propagation as well as the stronger capability of network representation. With (6.12), back-propagation and forward-propagation can be performed by setting $\mathbf{H} = \mathbf{I}$ in (6.10) and (6.11), respectively. Therefore, the channel effects can be completely removed.

The above discussion shows the importance of CSI in model training. However, CSI can be only estimated generally, i.e., LS, LMMSE, or MMSE estimators. Due to exploiting prior channel statistics, LMMSE and MMSE estimators usually perform better than the LS estimators. Thus, LMMSE and MMSE estimators are sensitive to the accuracy of channel statistic while the LS estimator requires no prior channel information. Meanwhile, DL techniques can also be used to improve the performance of channel estimation [152, 153].

For simplicity, we initially use the LS estimator. Then, we adopt the deep de-noise network to increase the resolution of the LS estimator as in [154] shown in Fig. 6.2. Particularly, the rough CSI estimated by the LS estimator with few pilots first denoted by

$$\mathbf{H}_{\texttt{rough}} = \mathbf{Y}_p\mathbf{X}_p^H = \mathbf{H} + \mathbf{N}\mathbf{X}_p^H, \tag{6.13}$$

where $\mathbf{Y}_p = \mathbf{H}\mathbf{X}_p+\mathbf{N}$, $\mathbf{Y}_p$ is the received pilot signal, $\mathbf{X}_p$ is the transmitted pilot signals. Then, (6.13) can be represented as

$$\mathbf{H}_{\texttt{rough}} = \mathbf{H} + \widehat{\mathbf{N}}, \tag{6.14}$$

where $\widehat{\mathbf{N}} = \mathbf{N}\mathbf{X}_p^H$.

From (6.14), $\mathbf{H}_{\texttt{rough}}$ consists of exact $\mathbf{H}$ and the noise, $\widehat{\mathbf{N}}$. De-noise neural networks are used to recover $\mathbf{H}$ more accurately from $\mathbf{H}_{\texttt{rough}}$ by considering $\mathbf{H}$ and $\mathbf{H}_{\texttt{rough}}$ as the original picture and noisy picture, respectively. Here, we exploit attention-guided denoising convolutional neural network (ADNet) [155] to refine CSI. ADNet includes four blocks, a sparse block, a feature enhancement block, an attention block, and a reconstruction block. After the input image, the sparse block is used to extract useful features from the given noisy image. Attention block can extract the noise information hidden in the complex background and is integrated into the feature enhancement block to reduce the complexity. Finally, the de-noised image is reconstructed by the reconstruction block.

The refined CSI, $\mathbf{H}_{\texttt{refine}}$ denoted by

$$\mathbf{H}_{\texttt{refine}} = \text{ADNet}\left(\mathbf{H}_{\texttt{rough}}\right). \tag{6.15}$$

In (6.15), the ADNet$(\cdot)$ is trained the the loss function, $\mathcal{L}\left(\mathbf{H}_{\texttt{refine}}, \mathbf{H}\right) = \frac{1}{2}\left\|\mathbf{H}_{\texttt{refine}} - \mathbf{H}\right\|_F^2$. Since the performance of the LS estimator is similar to that of LMMSE and MMSE estimators in the high SNR region, we pay more attention to the low SNR region when training ADNet. With proper training, ADNet can mitigate the impacts from noise but without any prior channel information, especially in the low SNR region. Such a design provides a good solution for *Question 1*.

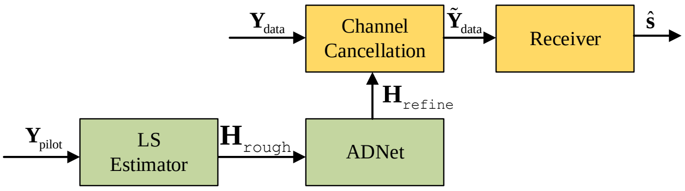


Figure 6.2: The proposed CSI refinement and cancellation based on de-noise neural networks.

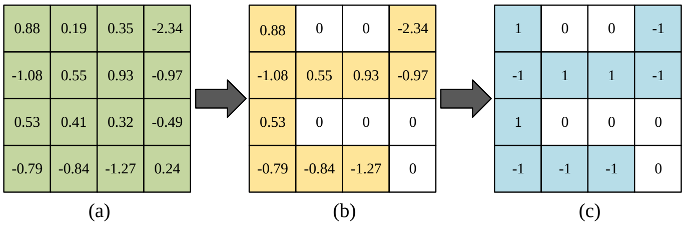
6.3.2 Model Compression

Through applying CSI into model training, the cloud/edge platform can extract the semantic features from L-DeepSC. However, the size and complexity of the trained L-DeepSC model are still very large, which causes high latency for the cloud/edge platform to broadcast updated L-DeepSC. Note that both weights pruning and quantization can reduce the model size and complexity, therefore, we compress the DeepSC model by a joint pruning-quantization scheme to make it affordable for IoT devices. As shown in Fig. 6.3, the original weights are first pruned at a high-precision level by identifying and removing the unnecessary weights, which makes the network sparse. Quantization is then used to convert the trained L-DeepSC model into a low-precision level. The proposed network sparsification and quantization can address *Question 3* and are introduced in detail in the following.

6.3.2.1 Network Sparsification

A proper criterion to disable neural connections is important. Obviously, the connections with small weight values can be pruned. Therefore, the pruning issue here turns into setting a proper pruning threshold.

As shown in Fig. 6.1(b), the DeepSC consists with neural networks, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}$, where each includes multiple layers. As the DeepSC mainly consists of dense layers, we choose unstructured pruning method in this chapter, where the computation workload of sparse model can be reduced by the sparsity algorithm and field-programmable gate array design [156, 157], i.e., sparse matrix-vector multiplication. Assume there are total N layers in



0.88	0.19	0.35	-2.34
-1.08	0.55	0.93	-0.97
0.53	0.41	0.32	-0.49
-0.79	-0.84	-1.27	0.24

(a)

0.88	0	0	-2.34
-1.08	0.55	0.93	-0.97
0.53	0	0	0
-0.79	-0.84	-1.27	0

(b)

1	0	0	-1
-1	1	1	-1
1	0	0	0
-1	-1	-1	0

(c)

Figure 6.3: Flowchart of the proposed joint pruning-quantization, (a) the original weights matrix; (b) the weights after pruning, where the example pruning function is $x = 0$ for $x < 0.5$; (c) the weights after quantization, where the example quantization function is $x = \text{sign}(x)$.

the pre-trained DeepSC model with $\mathbf{W}_{i,j}^{(n)}$ being the weight of connection between the i_{th} neuron of the $(n+1)_{th}$ layer and j_{th} neuron of n_{th} layer. With a pruning threshold $w_{\texttt{thre}}$, the model weights can be pruned by

$$\mathbf{W}_{i,j}^{(n)} = \begin{cases} \mathbf{W}_{i,j}^{(n)}, \text{ if } \left|\mathbf{W}_{i,j}^{(n)}\right| > w_{\texttt{thre}}, \\ 0, \qquad \text{otherwise}, \end{cases} \tag{6.16}$$

We determine the pruning threshold by

$$w_{\texttt{thre}} = \boldsymbol{w}_{N_c \times \gamma}, \tag{6.17}$$

where $\boldsymbol{w} = \text{sort}\left(\left[\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \cdots, \mathbf{W}^{(N)}\right]\right)$, is the sorted weights value from least important one to the most important one, N_c is the total number of connections, and γ, the sparsity ratio between 0 and 1, indicates the proportion of zero values in weights. The weight pruning can be divided into two steps, weight pruning to disable some neuron connections and fine-tine to recover the accuracy, as shown in Algorithm 6.1.

6.3.2.2 Network Quantization

The quantization includes weight quantization and activation quantization. The weights, $\mathbf{W}_{i,j}^{(n)}$, from a trained model, can be converted from 32-bit float point to m-bits integer

Algorithm 6.1: *Network Sparsification.*

Input: The pre-trained weights $\mathbf{W}$, the sparse ratio γ.

Output: The pruned weights $\mathbf{W}_{\texttt{pruned}}$.

```
1: Count the the total number of connections, $M$.
2: Sort the whole connections from small to large, $\boldsymbol{s}$.
3: Obtain the threshold by (6.17) with $N_c$ and $\gamma$, $w_{\texttt{thre}}$.
4: for $n = 1$ to $N$ do
5:     Prune the connections by (6.16), $\mathbf{W}^{(n)}_{\texttt{pruned}}$.
6: end for
7: Fine-tune the pruned model by loss function (6.4).
```

through applying the quantization function by

$$\tilde{\mathbf{W}}^{(n)}_{i,j} = \text{round}\left(q_w\left(\mathbf{W}^{(n)}_{i,j} - \min\left(\mathbf{W}^{(n)}\right)\right)\right), \tag{6.18}$$

where q_w is the scale-factor to map the dynamic range of float points to an m-bits integer, which is given by

$$q_w = \frac{2^m - 1}{\max\left(\mathbf{W}^{(n)}\right) - \min\left(\mathbf{W}^{(n)}\right)}. \tag{6.19}$$

For activation quantization, the results of matrix multiplication are stored in accumulators. Due to the limited dynamic range of integer formats, it is possible that the accumulator overflows quickly if the bit-width for the weights and activation is the same. Therefore, accumulators are usually implemented with higher bit-widths, for example, INT32 += INT8× INT8. Besides, the range of activations is dynamic and dependent on the input data. Therefore, the output of activations has to re-quantize into m-bits integer for the subsequent calculation. Unlike weights that are constant, the output of activations usually includes elements that are statistical outliers, which expand the actual dynamic range. For example, even if 99% of the data is distributed between -100 and 100, an outlier, 10,000, will extend the dynamic range into from -100 to 10,000, which significantly reduces the mapping resolution. In order to reduce the influence from the outliers, an exponential moving average is used by

$$x^{(n)}_{\min}(t+1) = (1-c)\,x^{(n)}_{\min}(t) + c\min\left(\mathbf{X}^{(n)}\left(t\right)\right), \tag{6.20}$$

Algorithm 6.2: *Network Quantization.*

```
Input: The pre-trained weights $\mathbf{W}$, the quantization level $m$,
    the correlation coefficient $c$, and the calibration data $\mathcal{K}$.
Output: The pre-trained weights $\mathbf{W}_{\texttt{quantized}}$ and the range of
    activation $x_{\min}$ and $x_{\max}$.
Phase 1: Weights Quantization.
for $n = 1$ to $N$ do
    Compute the range of weights, $\max\left(\mathbf{W}^{(n)}\right)$ and $\min\left(\mathbf{W}^{(n)}\right)$.
    Quantize the weights by (6.18), $\tilde{\mathbf{W}}^{(n)}$.
end for
Phase 2: Activations Quantization.
for $t = 1$ to $\mathcal{K}$ do
    for $n = 1$ to $N$ do
        Update the dynamic range of activation by (6.20) and (6.21), $x_{\min}^{(n)}(t)$ and $x_{\max}^{(n)}(t)$.
    end for
end for
Quantize the activations by (6.22).
Fine-tune the quantized model by STE and loss function (6.4).
```

and

$$x_{\max}^{(n)}(t+1) = (1-c)\, x_{\max}^{(n)}(t) + c \max\left(\mathbf{X}^{(n)}\left(t\right)\right), \tag{6.21}$$

where $x_{\min}^{(n)}(t+1)$ and $x_{\max}^{(n)}(t+1)$ are used for the range of activation quantization, and $x_{\min}^{(n)}(1) = \min\left(\mathbf{X}^{(n)}\left(1\right)\right)$, $x_{\max}^{(n)}(1) = \max\left(\mathbf{X}^{(n)}\left(1\right)\right)$, $\mathbf{X}^{(n)}(t)$ is the output of activations at n_{th} layer with t_{th} batch data, $c \in [0, 1)$ represents the correlation between the current $x_{\min}^{(n)}/x_{\max}^{(n)}$ with its past value. The effects from outliers can be mitigated by the past normal values. After $t+1$ epochs, the $x_{\min}^{(n)}$ and $x_{\max}^{(n)}$ are fixed based on $x_{\min}^{(n)}(t+1)$ and $x_{\max}^{(n)}(t+1)$. Then, the output of the activations can be quantized by

$$\tilde{\mathbf{X}}^{(n)} = \operatorname{clamp}\left(\operatorname{round}\left(q_x\left(\mathbf{X}^{(n)} - x_{\min}^{(n)}\right)\right); -(2^m-1), (2^m-1)\right), \tag{6.22}$$

where $q_a = (2^m - 1)/(x_{\max}^{(n)} - x_{\min}^{(n)})$ is the scale-factor and clamp $(\cdot)$ is used to eliminate the quantized outliers, which is given by

$$\text{clamp}\left(\mathbf{X}^{(n)}; -(2^m-1), (2^m-1)\right) = \min\left(\max\left(\mathbf{X}^{(n)}, -(2^m-1)\right), (2^m-1)\right), \quad (6.23)$$

where $2^m - 1$ is the border of the m-bits integer format.

As shown in Algorithm 6.2, the network quantization includes two phases: i) weight quantization; ii) activations quantization. In phase 1, the weights of each layer can be quantized by (6.18) directly. In phase 2, the calibration process is applied by running a few calibration batches in order to get the activations statistics. In each batch, $x_{min}^{(n)}(t)$ and $x_{max}^{(n)}(t)$ will be updated based on the activations statistics from the previous batches. These quantization processes might lead to slight accuracy degradation. The quantization-aware training (QAT) is required to re-train for minimizing the loss of accuracy. Since the rounding operation is not derivable, a straight-through estimator (STE) is used to estimate the gradient of quantized weights in the back-propagation [158].

6.3.3 Constellation Design with Fewer Quantization Bits

The cloud/edge platform can further reduce the size of L-DeepSC with model compression after the model is trained, which not only reduces the latency significantly for broadcasting the updated DeepSC to IoT devices, but also changes DeepSC to L-DeepSC with low complexity. However, high-resolution waveform poses high requirements cost-sensitive IoT devices. In other words, the cost-sensitive IoT devices are usually capacity-limited and cannot afford a large number of constellation points which are with phase and amplitude close to each other.

Different from bits, the source message, $\mathbf{s}$, is more complicated and the learned constellation will not be limited to a few points, which brings additional burden on hardware. Besides, the DL models generally run in FP32, which also expands the range of constellation. Thus, we aim to reduce the size of learned constellation without degrading performance, where the output of $\mathbf{X}$ is the learned constellation while $\mathbf{X}$ is also the output of activation of last layer at the local IoT devices. Inspired from the network quantization, we convert the learned high-resolution constellation into low-resolution one with few points. Thus, we use two-stage quantization to narrow the range of constellations, which is represented by

$$\mathbf{X}_{\texttt{dequantize}} = \frac{\mathbf{X}_{\texttt{quantize}}}{q_x} + x_{\min}, \tag{6.24}$$

where $\mathbf{X}_{\texttt{quantize}}$ is the quantized $\mathbf{X}$ from (6.22), q_x is the scale-factor and $x_{\min}$ is the obtained by (6.20) and $\mathbf{X}_{\texttt{dequantize}}$ is the dequantized $\mathbf{X}$.

First, we quantize the $\mathbf{X}$ into m-bits integer so that the range of $\mathbf{X}$ is narrowed to the size of 2^m. For example, when $m = 8$, the size of the constellation is reduced to 256. Then, $\mathbf{X}_{\texttt{quantize}}$ is dequantize to restore $\mathbf{X}$. Such an $\mathbf{X}_{\texttt{dequantize}}$ has a similar distribution as $\mathbf{X}$ but is with fewer constellation points, which is helpful to lower the hardware cost at transmitter and preserves the performance as much as possible and therefore provides the solution for *Question 2*.

In summary, by exploiting the solutions for the aforementioned *Questions*, we develop a lite distributed semantic communication system, named L-DeepSC, which could reduce the latency for model exchange under limited bandwidth, run the models at IoT devices with low power consumption, and deal with the distortion from fading channels when uploading semantic features. As a result, the proposed L-DeepSC becomes a good candidate for the IoT networks.

6.4 Numerical Results

In this section, we compare the proposed L-DeepSC with traditional methods under different fading channels, including Rayleigh and Rician fading channels. The weights pruning and quantization are also verified under fading channels. For the Rayleigh fading channel, the channel coefficient follows $\mathcal{CN}(0, 1)$; for the Rician fading channel. it follows $\mathcal{CN}(\mu, \sigma^2)$ with $\mu = \sqrt{k/(k+1)}$ and $\sigma = \sqrt{1/(k+1)}$. where k is the Rician coefficient and we use $k = 2$ in our simulation.

The transmitter of L-DeepSC is the same as that of DeepSC in Chapter 2. The parameters for the decoding network at the receiver are shown in Table 6-A for the fading channels, where the sum of the outputs of Dense 3 and Dense 5 is the input of the LayerNorm layer. The Transformer encoder and decoder are semantic encoder and decoder Chapter 2, respectively, which enables the systems to understand text and extract semantic information. We also prune the whole network since we consider the communications between cloud/edge platform and each IoT devices as well as the communications between IoT devices.

Table 6-A: The setting of L-DeepSC transceiver.

	Layer Name	Units	Activation
Transmitter	Embedding layer	128	None
	4×Transformer Encoder	128 (8 heads)	None
	Dense 1	256	Relu
	Dense 2	16	None
Receiver	Dense 3	128	Relu
	Dense 4	512	Relu
	Dense 5	128	None
	LayerNorm	None	None
	4×Transformer Decoder	128 (8 heads)	None
	Prediction Layer	Dictionary Size	Softmax

The output features are with 8 symbols per word. We initialize the learnable embedding matrix from $\mathcal{N}(0,1)$ with shape (vocab size, embedding-dim). The embedding dim is set to 128 in our program and the vocab size depends on the training dataset. The batch size is 64, learning rate is $128^{-0.5}\min\left(step^{-0.5}, step\times 4000^{-1.5}\right)$, where *step* is the counting number in the back-propagation. This corresponds to increasing the learning rate linearly for the first 4000 training steps and decreasing it thereafter proportionally to the inverse square root of the step number. We also adopt the L_2 regularization and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\varepsilon = 10^{-8}$.

The adopted dataset is the proceedings of the European Parliament [131], which consists of around 2.0 million sentences and 53 million words. The dataset is pre-processed into lengths of sentences with 4 to 30 words and is split into training data and testing data with 0.1 ratio. The benchmark approach is based on separate source coding and channel coding technologies, which adopt variable-length coding (Huffman coding) for source coding, where we build the Huffman codes by counting the frequency of letters and punctuation so that the look-up table is not large. Turbo coding and RS coding [133] for channel coding, where turbo decoding method is log-MAP algorithm with 5 iterations, and QAM. The BLEU score is used to measure the performance [128].

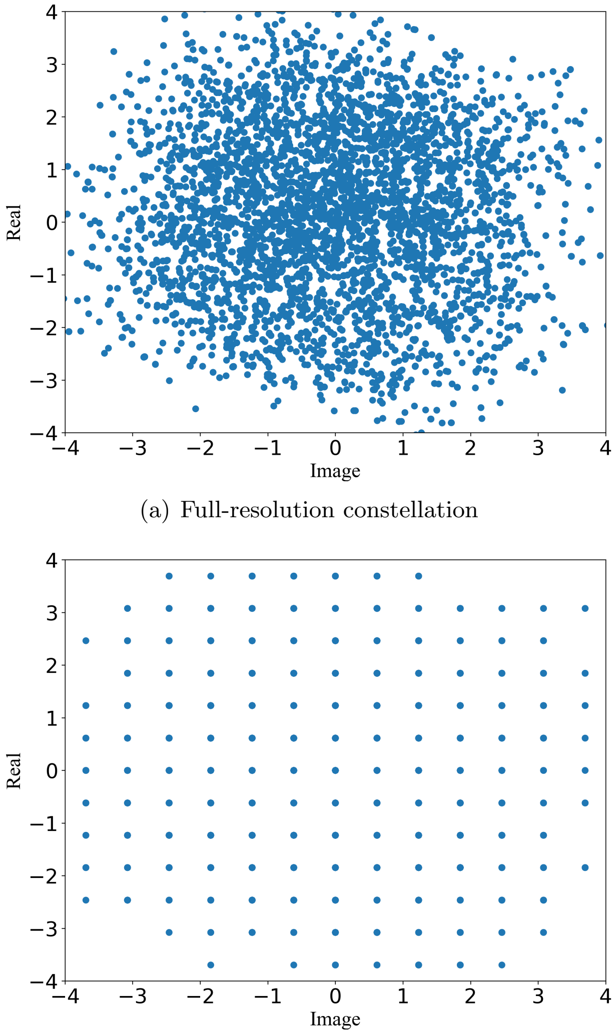
6.4.1 Constellation Design

Fig. 6.4 compares the full-resolution constellation and the 4-bits constellation. The full-resolution constellation points in Fig. 6.4(a) contain more information due to the higher resolution, but require complicated hardware, which is almost impossible to design. Through mapping the full-resolution constellation into a finite space, the 4-bits constellation points in Fig. 6.4(b) become simplified, which makes it possible to implement in the existing RF system. Note that the 4-bits constellation keeps a similar distribution with the full-resolution constellation. For example, there exist certain blank regions at the edge of the constellation in Fig. 6.4(a), while the 4-bits constellation shows a similar trend in Fig. 6.4(b). Such similar distribution prevents sharp performance degradation when the resolution of constellation decreases significantly.

Fig. 6.5 shows the BLEU scores versus SNR for different constellation sizes under AWGN, including 4-bits constellation, 8-bits constellation, and full-resolution constellation. All of them could achieve very similar performance when SNR $> 9\,\text{dB}$, which demonstrates the constellation design is effective and cause no significant performance degradation. Full resolution and 8-bits constellations perform slightly better than 4-bits constellation when SNR is low. This is because some weights information used for denoising is lost when the resolution of the constellation is small.

6.4.2 Performance over Fading Channels

Fig. 6.6 compares the channel estimation MSEs of LS, MMSE, and ADNet-aided LS estimator versus SNR under the Rayleigh fading channels. Note that MMSE equals to



(a) Full-resolution constellation

(b) 4-bits constellation

Figure 6.4: The comparison between the full-resolution constellation and 4-bits constellation.

LMMSE for the AWGN channels. The MMSE and LS estimators have similar accuracy in the high SNR region, thus the range of training SNRs for the ADNet is set from 0 dB to 10 dB to improve the performance of the LS estimator in the low SNR region. As a result, the MSE of ADNet based LS estimator is significantly lower than that of LS and MMSE estimators when SNR is low. With increasing SNR, the MSE of ADNet based LS estimator approaches to that of the LS and MMSE estimators. Therefore, the ADNet based LS estimator can be substituted by the LS estimator to reduce the complexity in the high SNR region.

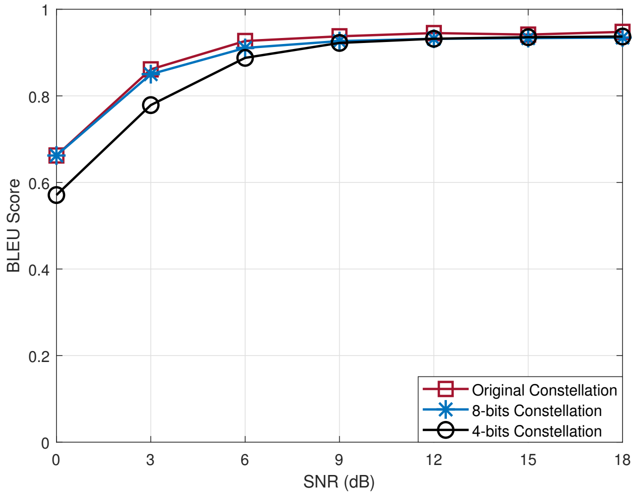


Figure 6.5: The BLEU scores of different constellation sizes versus SNR under AWGN.

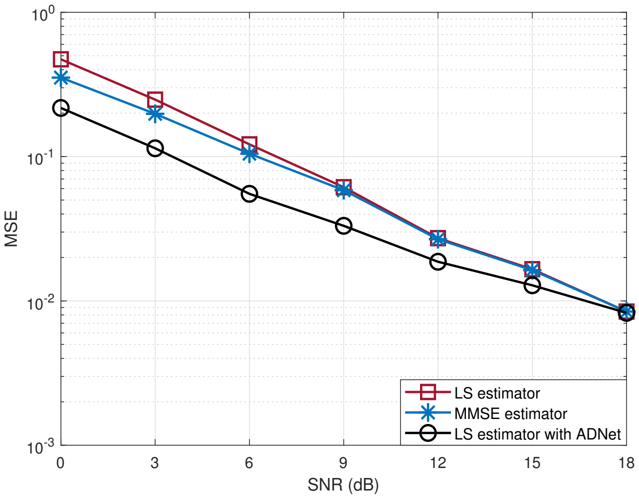


Figure 6.6: The MSE for MMSE estimator, LS estimator, and the proposed ADNet based LS estimator.

Fig. 6.7 and Fig. 6.8 illustrate the relationship between BLEU score and SNR with the 4-bits constellation over the Rician and the Rayleigh fading channels, respectively, where DeepSC is trained with perfect CSI and the L-DeepSC is trained with perfect CSI, rough CSI by (6.14), refined CSI by (6.15) and without CSI, respectively. The traditional approaches are Huffman coding with (5,7) RS and with turbo coding (rate 1/2), both with 64-QAM. We observe that all DL-enabled approaches are more competitive under the fading channels. RS coding is better than turbo coding in terms of BLEU score. This is because RS coding is linear block coding with long block-length, which can correct long

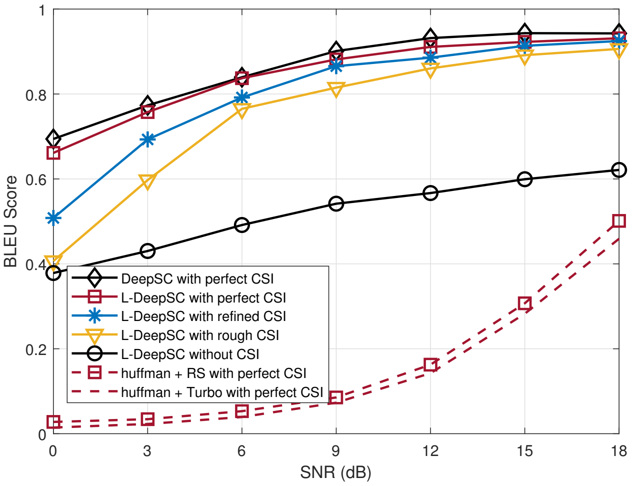


Figure 6.7: The BLEU scores versus SNR under Ricain fading channels, with perfect CSI, rough CSI, refined CSI, and no CSI.

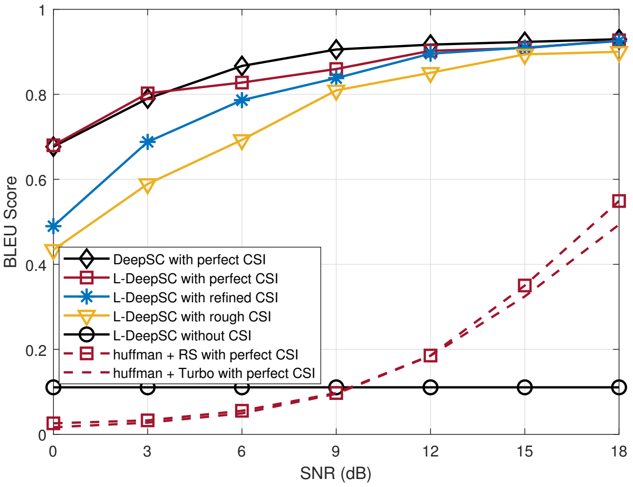


Figure 6.8: The BLEU scores versus SNR under Rayleigh fading channels, with perfect CSI, rough CSI, refined CSI, and no CSI.

bit sequences, however, turbo coding is convolution coding with short block-length, where the coded bits only are related with previous m bits, i.e., $m = 3$, so that the adjacent words result in higher error rate. The performance of L-DeepSC is very close to that of DeepSC in terms of BLEU score, but requires much less bandwidth for communications. The system trained without CSI performs worse than those trained with CSI, especially under the Rayleigh fading channels, which also confirms the analysis of (6.10) and (6.11). Without CSI, the performance difference between the Rayleigh channels and the Rician channels is caused by the line-of-sight, which can help the systems recognize the semantic information during training. Besides, with the aid of CSI, the effects of the fading channels are mitigated significantly, as we have analyzed before. When SNR is low, the system with perfect CSI or refined CSI outperforms that with rough CSI. As SNR increases, all these systems, L-DeepSC with perfect CSI, refined CSI, and rough CSI, converge to similar performance gradually.

6.4.3 Model Compression

In this experiment, we investigate the performance of network slimming, including network sparsification, network quantization, and the combination of both. The pre-trained model used for pruning and quantization is trained with 4-bits constellation under the Rician fading channels.

Fig. 6.9 shows the influences of network sparsity ratio, γ, on the BLEU scores with different SNRs under the Rician fading channels, where the system is pruned directly when γ increases from 0 to 0.9 and is pruned with fine-tuning when γ increases to 0.99 continually. The proposed L-DeepSC achieves almost the same BLEU scores when the γ increases from 0 to 0.9, which shows that there exists a mass of weights redundancy in the trained DeepSC model. When the γ increases to 0.99, the BLEU scores still drop slightly due to the processing of fine-tuning, where the performance loss at 0 dB and 6 dB is larger than that at 12 dB and 18 dB. Thus, for the high SNR cases, the model can be pruned directly with only slight performance degradation. For the low SNR region, it is possible to prune 99% weights without significant performance degradation when the system is sensitive to power consumption.

Fig. 6.10 demonstrates the relationship between the BLEU score and the quantization bit number, m, under the Rician fading channels, where m is defined in (6.19), and the system is quantized with QAT when the m is smaller than 2. The performance with $m = 8$ to $m = 20$ is similar, which indicates that the effectiveness of low-resolution neural networks. If the system is more sensitive to power consumption and can tolerant to certain performance degradation, the resolution of the neural networks can be further

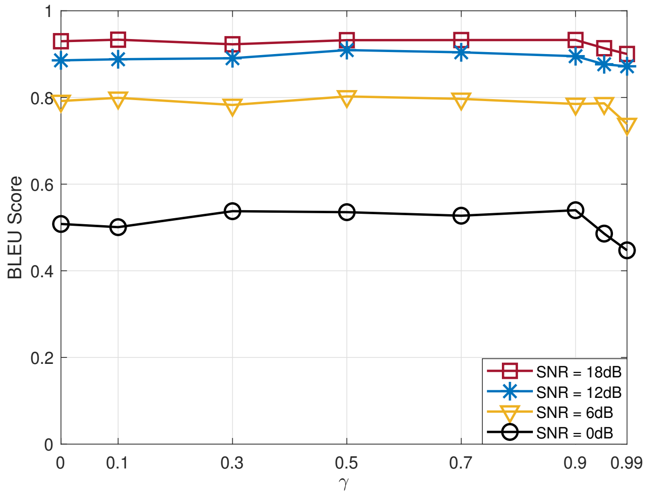


Figure 6.9: The BLEU scores of different SNRs versus sparsity ratio, γ, under Rician fadings channel with the refined CSI.

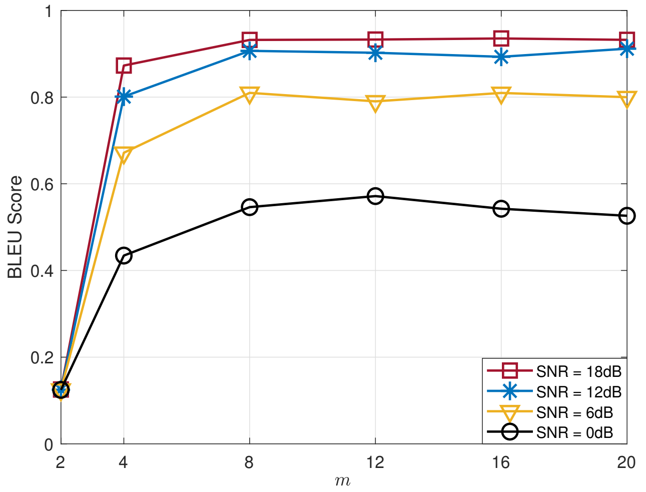


Figure 6.10: The BLEU scores of different SNRs versus quantization level, m, under Rician fading channels with the refined CSI.

reduced to 4-bits level. However, the BLEU score decreases dramatically from $m = 4$ to $m = 2$ over the whole SRN range since most of the key information is removed in the low-resolution neural network.

Table 6-B compares the BLEU scores and compression ratios under different combinations of weights pruning and weights quantization with SNR is 12 dB, where the

Table 6-B: The BLEU score and compression ratio, ψ, Comparisons versus different sparsity ratio, γ, and quantization level, m, in SNR $= 12dB$.

Pruned Model	BLEU score with $m = 4$	ψ	BLEU score with $m = 8$	ψ	BLEU score with $m = 12$	ψ
$\gamma = 0$	0.811194	8	0.906763	4	0.902354	2.667
$\gamma = 0.3$	0.838967	11.429	0.892745	5.714	0.908537	3.81
$\gamma = 0.6$	0.835863	20.0	0.897143	10.0	0.90815	6.667
$\gamma = 0.9$	0.810322	80.0	0.895306	40.0	0.898784	26.667
$\gamma = 0.95$	0.779685	160.0	0.875814	80.0	0.873426	53.333

Pruned Model	BLEU score with $m = 16$	ψ	BLEU score with $m = 32$	ψ
$\gamma = 0$	0.903089	2	0.895602	1
$\gamma = 0.3$	0.910184	2.857	0.89851	1.429
$\gamma = 0.6$	0.900468	5.0	0.9093	2.5
$\gamma = 0.9$	0.910554	20.0	0.89515	10
$\gamma = 0.95$	0.877221	40.0	0.87653	20

compression ratio is computed by

$$\psi = \frac{M \times 32}{M_{\texttt{pruned}} \times m}, \tag{6.25}$$

where M is the number of weights before pruning and $M_{\texttt{pruned}}$ is the number of weights remaining after pruning, 32 is the number of required bits for FP32 and m is the number of the required bits after quantization. The performance decreases when γ increases or m decreases, which are consistent with Fig. 6.9 and Fig. 6.10. From the table, different compression ratios could lead to similar performance. For example, the BLEU score with $\gamma = 30\%$ and $m = 8$ is similar to that with $\gamma = 90\%$ and $m = 12$, but the compression ratio is about five times different, i.e., 5.714 and 26.667. By properly choosing a suitable sparsity ratio and a quantization level, the same performance can be achieved but with a high compression ratio.

Table 6-C compares the DeepSC and L-DeepSC with 60% weights sparsity and 8-bit quantization when SNR is 12 dB, where we mainly consider the transmission of the weights. The simulation is performed in CPU by the computer with Intel Core i7-9700CPU@3.00GHz. After network slimming, the model size is reduced from 12.3 MB to 1.28 MB while achieving a similar BLEU score, which means the bandwidth resource can be saved significantly without degrading the performance. Besides, the runtime slightly decreases from 20ms to 18ms since the unstructured pruning method is employed, and there exists the communication time between flash memory and some operation that can not be optimized. If the model size is bigger, the L-DeepSC could save more runtime.

Table 6-C: The comparison between L-DeepSC and DeepSC transceiver in parameters, size, runtime, and BLEU score.

	Parameters	Size	Runtime	BLEU score
$\gamma = 0$, $m = 32$	3,333,120	12.3 MB	20ms	0.895602
$\gamma = 0.6$, $m = 8$	1,333,247	1.28 MB	18ms	0.897143

6.5 Summary

In this chapter, we proposed a lite distributed semantic communication system, named L-DeepSC, for the IoT networks, where the participating devices are usually with limited power and computing capabilities. Specially, the receiver and feature extractor were designed jointly for text transmission. Firstly, we analyzed the effectiveness of CSI in forward-propagation and back-propagation during system training over the fading channels. The analytical results reveal that the fading channels contaminate the weights update and restrict model representation capability. Thus, a refined LS estimator with fewer pilot overheads was developed to eliminate the effects of fading channels. Besides, we map the full-resolution original constellation into finite bits constellation to lower the cost of IoT devices, which was verified by simulation results. Finally, due to the limited narrow bandwidth and computational capability in IoT networks, two model compression approaches have been proposed: 1) the network sparsification to prune the unnecessary weights, and 2) network quantization to reduce the weights resolution. The simulation results validated that the proposed L-DeepSC outperforms the traditional methods, especially in the low SNR regime, and has provided insights into the balance among compression ratio, sparsity ratio, and quantization level. Therefore, the proposed L-DeepSC is a promising candidate for intelligent IoT networks, especially in the low SNR regime. In the next chapter, we will conclude these works and present the future works.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis presented the various design of deep learning based semantic communications (DeepSC) from single-user to multiple-user, from memoryless tasks to memory tasks, and from general complexity to low complexity. It has been demonstrated that semantic communication has shown a great potential to increase the reliability in performing intelligent tasks, reduce network traffic, and alleviate spectrum shortage. The following two aspects are represented in this thesis: 1) The basic DeepSC model to explore the possibility of semantic communication, and 2) the variants of DeepSC for different scenarios, i.e., multiple-user communication, energy-constrained device communication, and various task-oriented communication. The main contributions and insights are summarized as follows.

In chapter 3, the basic DeepSC was proposed for text transmission, in which the meanings behind sentences are extracted at the transmitter and sentences are reconstructed according to the received meanings at the receiver. Additionally, a new loss function was designed to improve the transmission rate as well as the sentence reconstructed accuracy. For applying DeepSC in dynamic environments and different knowledge, transfer learning was utilized to boost the training by replacing part modules in DeepSC. Furthermore, numerical results demonstrated that DeepSC is more robust than conventional communication systems, especially in the low SNR regimes.

In chapter 4, the multi-user DeepSC was proposed for multi-user communication scenario, in which the transmitters gather multimodal data from different users/devices, transmit over the air, and process/fuse multimodal data at the receiver. The unified framework was proposed to enable the transmitter reusing the same deep neural network (DNN) structure to support both the image user and text user semantic extraction. Based on the unified framework, two receivers named DeepSC-IR and DeepSC-MT were designed for the image retrieval and machine translation task in single model multi-user scenario. Considered the appearing multimodal multi-user scenario, the receiver named DeepSC-VQA was designed to fuse the image and text semantic information for the visual question answering (VQA) task. Numerical results verified the superiority of our proposed multi-user DeepSC.

In chapter 5, the DeepSC with memory was represented to support both memoryless and memory tasks, in which the transmitter transmits the context information and the receiver can perform the intelligent task related to both the current input (memoryless task) and the past inputs (memory task). By introducing the memory module, the Mem-DeepSC was designed to perform the context question answering task. Additionally, the semantic-aware channel capacity was derived to verify the possibility of dynamic transmission. Based on the semantic-aware channel capacity, two dynamic transmission methods were proposed to transmit the context over multiple time slots effectively by masking the unimportant elements. Numerical results demonstrated that Mem-DeepSC is capable of offering a graceful solution to perform both memoryless and memory tasks.

In chapter 6, the low-complexity DeepSC was proposed for capacity-constrained device communication scenario, in which the Internet-of-Things (IoT) devices need to encode/decode semantic information with capacity-limited hardware and update the model under the narrow band. The prune and quantization techniques were employed to remove the unimportant connections between neurons and run the model in INT32 instead of FLOAT32, which can reduce both the computational complexity and the model size. Furthermore, the finite-points constellation was derived to serve for the low-resolution phase-shift and amplitude-shift circuits. In order to reduce the transmission errors, the two-stage channel estimation was designed to refine the channel estimation information (CSI) obtained from the least-square channel estimation. Numerical results proved that the lite DeepSC can achieve lower complexity as well as similar performance compared to the DeepSC.

In a summary, in this thesis, various DeepSC designs were proposed for the basic single user communication scenario, the multi-user communication scenario, the capacity-constrained device communication scenario, and the task support communication scenario. Amount of simulations have been done to demonstrate the effectiveness and superiority of the proposed variants of DeepSC compared to the benchmark algorithms.

7.2 Future Work

The following three research issues have been identified and are to be addressed in future work, for the applications and implementations of semantic communications.

7.2.1 Hybrid Semantic Communication

The constellations in the proposed various semantic communication systems are changed with the different source data, which is suitable for analog communication. However, digital communication systems are mainly adopted in real life. In other words, the current transmission hardware is designed for digital communication systems and is not compatible with analog communication systems. Redesigning the hardware for semantic communication systems is a large cost. Therefore, the compatibility between semantic communication systems and digital communication systems is an important problem to apply semantic communication systems in real life.

7.2.2 One Model for All in Semantic Communication

The inborn characteristic of semantic communication systems can support various intelligent tasks. However, the semantic communication systems need to be redesigned the structure for different intelligent tasks, which requires strong prior knowledge to design such a network. Besides, when the environment changes, it also needs to redesign/retrain the model to fit the new communication scenario. Inspired by the large pre-trained model in computer science, the pre-trained model can support various downstream tasks with few/zero shot learning. In the semantic communication area, we can design a large pre-trained transceiver to support the different communication environments and various intelligent tasks so that reducing the design difficulty for researchers.

7.2.3 Semantic-Aware Channel Capacity

Semantic communication is in its infancy. The fundamental theory is still missing. Even if the author proposed the initial semantic-aware channel capacity for single-user communication, the unified semantic-aware channel capacity is required to unify different communication scenarios, e.g., multi-user communication, which can provide the insights to transmit the semantic information. Additionally, the application of semantic-aware channel capacity is another interesting direction, i.e., how to perform the power allocation with the semantic-aware channel capacity, and how to design the neural network with the guideline of semantic capacity.

Appendix A

Proof in Chapter 5

A.1 Proof of Proposition 1

Given the mini-batch, B, the question-answer accuracy can be computed by

$$Acc = \frac{1}{|B|} \sum_{B} \langle \mathbf{1}_i, \mathbf{1}_j \rangle, \tag{A.1}$$

where $|B|$ is the batch size, and $\mathbf{1}_i$ is the one-hot vector with one in the i-th position, $\mathbf{1}_i$ is the real answer with label i, and $\mathbf{1}_j$ represents the predicted answer with predicted label j, which is computed by

$$\mathbf{1}_j = \text{onehot}(\arg\max(\boldsymbol{l}))), \tag{A.2}$$

where $\boldsymbol{l}$ is the output logits before softmax activation.

Since softmax function is the soft function of onehot$(\arg\max(\cdot)))$, the $\mathbf{l}_j$ can be approximated by

$$\mathbf{1}_j \approx \boldsymbol{p} = \text{softmax}(\boldsymbol{l}), \tag{A.3}$$

where $\boldsymbol{p}$ is the predicted probabilities.

Submitting the (A.3) to (A.1), the answer accuracy can be approximated as

$$Acc \approx \frac{1}{|B|}\sum_{B}\langle \mathbf{1}_i, \boldsymbol{p}\rangle = \frac{1}{|B|}\sum_{B} p(a)p(\hat{a}). \tag{A.4}$$

where $p(a)$ is the real probability for label i and $p(\hat{a})$ is the i-th predicted probability at $\boldsymbol{p}$.

Based on (A.4), the loss function of answer accuracy can be designed as

$$\mathcal{L}_{\texttt{Acc}} = -\mathbb{E}\left[p(a)p(\hat{a})\right]. \tag{A.5}$$

The derivation of $\mathcal{L}_{\texttt{Acc}}$ for the parameters $\boldsymbol{\varphi}$ is

$$\nabla_{\boldsymbol{\varphi}}\mathcal{L}_{\texttt{Acc}} = p(\hat{a})\left(1 - p(\hat{a})\right)\nabla_{\boldsymbol{\varphi}}\boldsymbol{l}. \tag{A.6}$$

From (A.6), there exist two optimization directions when $\nabla_{\boldsymbol{\varphi}}\mathcal{L}_{\texttt{Acc}} \to 0$, i.e., $p(\hat{a}) \to 0$ and $p(\hat{a}) \to 1$. However, $p(\hat{a}) \to 0$ causes worse prediction results and should avoid. In order to make the optimization stable, the $\mathcal{L}_{\texttt{Acc}}$ should be refined. One refined loss function is the cross-entropy loss function given by

$$\mathcal{L}_{\texttt{CE}} = -\mathbb{E}\left[p(a)\log\left(p(\hat{a})\right)\right]. \tag{A.7}$$

The derivation of $\mathcal{L}_{\texttt{CE}}$ for the parameters $\boldsymbol{\varphi}$ is

$$\nabla_{\boldsymbol{\varphi}}\mathcal{L}_{\texttt{Acc}} = \left(1 - p(\hat{a})\right)\nabla_{\boldsymbol{\varphi}}\boldsymbol{l}. \tag{A.8}$$

Compared (A.6) and (A.8), the derivation of $\mathcal{L}_{\texttt{CE}}$ only has one correct optimization direction $p(\hat{a}) \to 1$, which is more stable during training. Therefore, the proposition 1 is derived.

A.2 Proof of Proposition 2

For the classification task, the mutual information, $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$, can be expressed as

$$I\left(\tilde{\boldsymbol{x}}_k^c; a\right) = H(a) - H(a|\tilde{\boldsymbol{x}}_k^c). \tag{A.9}$$

where $H(a)$ is the entropy of the real label, $H(a|\tilde{\boldsymbol{x}}_k^c)$ is the conditional entropy.

The cross-entropy between the real label and the predicted label given $\tilde{\boldsymbol{x}}_k^c$ is

$$H(a;\hat{a}|\tilde{\boldsymbol{x}}_k^c) = H(a|\tilde{\boldsymbol{x}}_k^c) + D_{\text{ KL}}\left(a||\hat{a}|\tilde{\boldsymbol{x}}_k^c\right), \tag{A.10}$$

where $D_{\text{KL}}\left(\cdot||\cdot\right)$ is the Kullback–Leibler divergence and is always non-negative. Therefore, we have the following inequality

$$H(a;\hat{a}|\tilde{\boldsymbol{x}}_k^c) \geqslant H(a|\tilde{\boldsymbol{x}}_k^c), \tag{A.11}$$

Submitting (A.11) into (A.9), the lower bound of $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$ can be obtained

$$I\left(\tilde{\boldsymbol{x}}_k^c; a\right) \geqslant H(a) - H(a;\hat{a}|\tilde{\boldsymbol{x}}_k^c). \tag{A.12}$$

From (A.12), since $H(a)$ is constant, maximizing the $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$ can be approximated to minimizing the $H(a;\hat{a}|\tilde{\boldsymbol{x}}_k^c)$. The lower bound will be closer to $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$ when the model is trained. Therefore, the proposition 3 is derived.

A.3 Proof of Proposition 3

For the regression task, the mutual information, $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$, can be expressed as (A.9).

Lemma 1. The conditional differential entropy yields a lower bound on the expected squared error of an estimator, for any random variable X, observation Y, and estimator

$\hat{X}$, the following holds

$$\mathbb{E}\left[\left(X-\hat{X}\left(Y\right)\right)^2\right] \geqslant \frac{1}{2\pi e}e^{2H(X|Y)}. \tag{A.13}$$

Applying the Lemma 1, the upper bound of conditional entropy, $H(a|\tilde{\boldsymbol{x}}_k^c)$, can be expressed as

$$H(a|\tilde{\boldsymbol{x}}_k^c) < \mathbb{E}\left[\ln\left|a-\hat{a}(\tilde{\boldsymbol{x}}_k^c)\right|\right], \tag{A.14}$$

where $\hat{a}(\tilde{\boldsymbol{x}}_k^c)$ means the model outputs $\hat{a}$ with the $\tilde{\boldsymbol{x}}_k^c$.

Submitting (A.14) into (A.9), the lower bound of $I\left(\tilde{\boldsymbol{x}}_k^c;a\right)$ can be obtained

$$I\left(\tilde{\boldsymbol{x}}_k^c;a\right) > H(a) - \mathbb{E}\left[\ln\left|a-\hat{a}\right|\right]. \tag{A.15}$$

From (A.15), since $H(a)$ is constant, maximizing the $I\left(\tilde{\boldsymbol{x}}_k^c;a\right)$ can be approximated to minimizing the $\mathbb{E}\left[\ln\left|a-\hat{a}\right|\right]$. However, directly minimizing the $\mathbb{E}\left[\ln\left|a-\hat{a}\right|\right]$ may cause the gradient explosion.

Given the derivation of $\ln\left|a-\hat{a}\right|$ for the parameters $\boldsymbol{\varphi}$,

$$\nabla_{\boldsymbol{\varphi}}\ln\left|a-\hat{a}\right| = \frac{1}{\left|a-\hat{a}\right|}\nabla_{\boldsymbol{\varphi}}\hat{a}. \tag{A.16}$$

From (A.16), when $\hat{a}\to a$, $\nabla_{\boldsymbol{\varphi}}\ln\left|a-\hat{a}\right|\to\infty$. In order to alleviate the gradient explosion, the approximation of $\ln\left|a-\hat{a}\right|$ is derived by applying the Taylor series expansion

$$\ln\left(\left|a-\hat{a}\right|-1+1\right) \approx \left|a-\hat{a}\right|-1. \tag{A.17}$$

The derivation of (A.17) for the parameters $\boldsymbol{\varphi}$ is

$$\nabla_{\boldsymbol{\varphi}}\left|a-\hat{a}\right| = \nabla_{\boldsymbol{\varphi}}\hat{a}. \tag{A.18}$$

Compared (A.18) and (A.16), the item, $\frac{1}{|a-\hat{a}|}$, is removed, therefore, the gradient explosion is eliminated. Then, the lower bound of $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$ can be expressed as

$$I\left(\tilde{\boldsymbol{x}}_k^c; a\right) > H(a) - \mathbb{E}\left[|a-\hat{a}|\right]. \tag{A.19}$$

From (A.19), maximizing the $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$ can be approximated to minimizing the $\mathbb{E}\left[|a-\hat{a}|\right]$. The lower bound will be closer to $I\left(\tilde{\boldsymbol{x}}_k^c; a\right)$ when the model is trained. Therefore, the proposition 4 is derived.

References

[1] "Introduction to question answering over knowledge graphs," Oct. 2019. [Online]. Available: https://yashuseth.wordpress.com/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/

[2] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[3] "Ericsson mobility report," Nov. 2021. [Online]. Available: https://www.ericsson.com/4ad7e9/assets/local/reports-papers/mobility-report/documents/2021/ericsson-mobility-report-november-2021.pdf

[4] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Commun. Surveys Tutorials*, vol. 21, no. 2, pp. 1275–1313, Sept. 2018.

[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Thingss: a survey," *Computer Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[6] M. J. Schuemie, P. Van Der Straaten, M. Krijn, and C. A. Van Der Mast, "Research on presence in virtual reality: A survey," *CyberPsychology & Behavior*, vol. 4, no. 2, pp. 183–201, Jul 2004.

[7] R. T. Azuma, "A survey of augmented reality," *Presence: teleoperators & virtual environments*, vol. 6, no. 4, pp. 355–385, Apr. 1997.

[8] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots.* MIT press, 2011.

[9] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication.* The University of Illinois Press, 1949.

[10] A. Goldsmith, *Wireless Communications.* Cambridge university press, 2005.

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

[12] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2923–2960, Jun. 2018.

[13] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[14] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 06, pp. 8–19, 1984.

[15] J. D. Allen, D. Anderson, J. Becker, R. Cook, M. Davis, P. Edberg, M. Everson, A. Freytag, L. Iancu, R. Ishida *et al.*, "The unicode standard," *Mountain view, CA*, pp. 660–664, 2012.

[16] D. Marpe, T. Wiegand, and G. J. Sullivan, "The h. 264/mpeg4 advanced video coding standard and its applications," *IEEE communications magazine*, vol. 44, no. 8, pp. 134–143, 2006.

[17] G. Parsons and J. Rafferty, "Tag image file format (tiff)-f profile for facsimile," Tech. Rep., 1998.

[18] D. Salomon and G. Motta, *Handbook of data compression.* Springer, 2010.

[19] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg 2000 still image compression standard," *IEEE Signal processing magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[20] S.-M. Lei, T.-C. Chen, and M.-T. Sun, "Video bridging based on h. 261 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 425–437, 1994.

[21] C. Poynton, *Digital video and HD: Algorithms and Interfaces.* Elsevier, 2012.

[22] D. O'Shaughnessy, "Linear predictive coding," *IEEE potentials*, vol. 7, no. 1, pp. 29–32, 1988.

[23] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications.* John Wiley & Sons, 1999.

[24] U. Kumar and B. Umashankar, "Improved hamming code for error detection and correction," in *2007 2nd International Symposium on Wireless Pervasive Computing.* IEEE, 2007.

[25] G. Forney, "On decoding bch codes," *IEEE Transactions on information theory*, vol. 11, no. 4, pp. 549–557, 1965.

[26] J. Chen, A. Dholakia, E. Eleftheriou, M. P. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of ldpc codes," *IEEE transactions on communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[27] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

[28] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding.* John Wiley & Sons, 2015.

[29] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of ICC'93-IEEE International Conference on Communications*, vol. 2. IEEE, 1993, pp. 1064–1070.

[30] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. Le Cun, "High quality document image compression with" djvu"," *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 410–425, 1998.

[31] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5435–5443.

[32] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.

[33] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS).* IEEE, 2018, pp. 253–257.

[34] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," in *Proc. Int'l. Conf. Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., Puerto Rico, May 2016.

[35] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *International Conference on Machine Learning.* PMLR, 2017, pp. 2922–2930.

[36] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *Proc. Int'l. Conf. on Information Sciences and Systems (CISS).* IEEE, 2017, pp. 1–6.

[37] X. Wu, M. Jiang, and C. Zhao, "Decoding optimization for 5g ldpc codes by machine learning," *IEEE Access*, vol. 6, pp. 50 179–50 186, 2018.

[38] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels," *Advances in neural information processing systems*, vol. 32, 2019.

[39] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *Advances in neural information processing systems*, vol. 31, 2018.

[40] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC).* IEEE, 2019, pp. 1–5.

[41] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Joint channel coding and modulation via deep learning," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC).* IEEE, 2020, pp. 1–5.

[42] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6g: Vision, enabling technologies, and applications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 5–36, 2021.

[43] J. Dewey, *Experience and nature.* Courier Corporation, 1958, vol. 471.

[44] L. Wittgenstein, *Philosophical investigations.* John Wiley & Sons, 2010.

[45] R. Carnap, Y. Bar-Hillel *et al.*, *An Outline of A Theory of Semantic Information.* RLE Technical Reports 247, Research Laboratory of Electronics, Massachusetts Institute of Technology., Cambridge MA, Oct. 1952.

[46] L. Floridi, "Outline of a theory of strongly semantic information," *Minds and machines*, vol. 14, no. 2, pp. 197–221, 2004.

[47] S. D'Alfonso, "On quantifying semantic information," *Information*, vol. 2, no. 1, pp. 61–101, 2011.

[48] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *Proc. in IEEE Network Science Workshop*, West Point, NY, USA, Jun. 2011, pp. 110–117.

[49] P. Basu, J. Bao, M. Dean, and J. Hendler, "Preserving quality of information by using semantic relationships," *Pervasive Mob. Comput.*, vol. 11, pp. 188–202, Apr. 2014.

[50] R. C. Schank and R. P. Abelson, *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.* Psychology Press, 2013.

[51] A. Newell, J. C. Shaw, and H. A. Simon, "Report on a general problem solving program," in *IFIP congress*, vol. 256. Pittsburgh, PA, 1959, p. 64.

[52] M. Jeong, B. Kim, and G. G. Lee, "Semantic-oriented error correction for spoken query processing," in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721).* IEEE, 2003, pp. 156–161.

[53] B. Guler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Trans. Cogn. Comm. Networking*, vol. 4, no. 4, pp. 787–802, Sep. 2018.

[54] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," *arXiv preprint cmp-lg/9511007*, 1995.

[55] Y. Wang, M. Chen, T. Luo, W. Saad, D. Niyato, H. V. Poor, and S. Cui, "Performance optimization for semantic communications: An attention-based reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2598–2613, 2022.

[56] F. Zhou, Y. Li, X. Zhang, Q. Wu, X. Lei, and R. Q. Hu, "Cognitive semantic communication systems driven by knowledge graph," *arXiv preprint arXiv:2202.11958*,

2022.

[57] Z. Wang, Y. Li, D. Huang, Y. Luo, N. Ge, and J. Lu, “Deformable geometry based semantic reconstruction from scene graphs,” in *Proc. Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[58] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. Int’l. Conf. Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[59] E. Cambria and B. White, “Jumping NLP curves: A review of natural language processing research [review article],” *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 48–57, Apr. 2014.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances Neural Info. Process. Systems (NIPS)*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Lake Tahoe, Nevada, United States, Dec. 2012, pp. 1106–1114.

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, Nevada, USA, Jun. 2016, pp. 770–778.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. Advances Neural Info. Process. Systems (NIPS)*, Long Beach, CA, USA. Dec. 2017, pp. 5998–6008.

[63] N. Farsad, M. Rao, and A. Goldsmith, “Deep learning for joint source-channel coding of text,” in *Proc. Int. Conf. Acoust., Speech Signal Processing (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 2326–2330.

[64] H. Xie, Z. Qin, G. Y. Li, and B. Juang, “Deep learning enabled semantic communication systems,” *IEEE Trans. Signal Processing*, vol. 69, pp. 2663–2675, Apr. 2021.

[65] H. Xie and Z. Qin, “A lite distributed semantic communication system for internet of things,” *IEEE J. Select. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[66] X. Peng, Z. Qin, D. Huang, X. Tao, J. Lu, G. Liu, and C. Pan, “A robust

deep learning enabled semantic communication system for text," *arXiv preprint arXiv:2206.02596*, Jun. 2022.

[67] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep source-channel coding for sentence semantic transmission with HARQ," *IEEE Trans. Commun.*, Jun. 2022.

[68] Y. Zhang, H. Zhao, J. Wei, J. Zhang, M. F. Flanagan, and J. Xiong, "Context-based semantic communication via dynamic programming," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, Early Access, 2022.

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

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

[71] ——, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 12, pp. 8081–8095, Jun. 2021.

[72] M. Yang, C. Bian, and H. Kim, "Deep joint source channel coding for wireless image transmission with OFDM," in *Proc. Int'l. Conf. on Comm. (ICC)*. IEEE, Montreal, QC, Canada, Jun. 2021, pp. 1–6.

[73] D. Huang, X. Tao, F. Gao, and J. Lu, "Deep learning-based image semantic coding for semantic communications," in *Proc. Global Commun. Conf. (GLOBECOM)*. IEEE, Madrid, Spain, Dec. 2021, pp. 1–6.

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

[75] H. Tong, Z. Yang, S. Wang, Y. Hu, W. Saad, and C. Yin, "Federated learning based audio semantic communication over wireless networks," in *Proc. Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[76] T. Han, Q. Yang, Z. Shi, S. He, and Z. Zhang, "Semantic-preserved communication system for highly efficient speech transmission," *arXiv preprint arXiv:2205.12727*, May 2022.

[77] T. Tung and D. Gündüz, "DeepWiVe: Deep-learning-aided wireless video trans-

mission," *IEEE J. Select. Areas Commun.*, Accept to be appeared, 2022.

[78] S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, and P. Zhang, "Wireless deep video semantic transmission," *arXiv preprint arXiv:2205.13129*, May 2022.

[79] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless semantic communications for video conferencing," *arXiv preprint arXiv:2204.07790*, Apr. 2022.

[80] P. Tandon, S. Chandak, P. Pataranutaporn, Y. Liu, A. M. Mapuranga, P. Maes, T. Weissman, and M. Sra, "Txt2Vid: Ultra-low bitrate compression of talking-head videos via text," *arXiv preprint arXiv:2106.14014*, Jun. 2021.

[81] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE J. Select. Areas Commun.*, Accept to be appeared, 2022.

[82] M. Sana and E. C. Strinati, "Learning semantics: An opportunity for effective 6g communications," in *Proc. Consumer Commun. & Networking Conference*, Las Vegas, NV, USA, Jan. 2022, pp. 631–636.

[83] Z. Weng, Z. Qin, and G. Y. Li, "Semantic communications for speech recognition," *arXiv preprint arXiv:2107.11190*, Jul. 2021.

[84] T. Han, Q. Yang, Z. Shi, S. He, and Z. Zhang, "Semantic-aware speech to text transmission with redundancy removal," in *Proc. Int'l. Conf. on Comm. (ICC)*, Seoul, South Korea, May 2022, pp. 1–6.

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

[86] Q. Hu, G. Zhang, Z. Qin, Y. Cai, G. Yu, and G. Y. Li, "Robust semantic communications with masked VQ-VAE enabled codebook," *arXiv preprint arXiv:2206.04011*, Jun. 2022.

[87] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Select. Areas Commun.*, vol. 39, no. 1, pp. 89–100, Jan. 2021.

[88] X. Kang, B. Song, J. Guo, Z. Qin, and F. R. Yu, "Task-oriented image transmission for scene classification in unmanned aerial systems," *IEEE Trans. Commun.*, Early Access, 2022.

[89] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented semantic communications for multi-

modal data," *arXiv preprint arXiv:2108.07357*, Aug. 2021.

[90] G. Zhang, Q. Hu, Z. Qin, Y. Cai, and G. Yu, "A unified multi-task semantic communication system with domain adaptation," *arXiv preprint arXiv:2206.00254*, Jun. 2022.

[91] J. Yen, "On nonuniform sampling of bandwidth-limited signals," *IRE Transactions on circuit theory*, vol. 3, no. 4, pp. 251–257, 1956.

[92] B. S. Melton and L. F. Bailey, "Multiple signal correlators," *Geophysics*, vol. 22, no. 3, pp. 565–588, 1957.

[93] J. H. Horne and S. L. Baliunas, "A prescription for period analysis of unevenly sampled time series," *The Astrophysical Journal*, vol. 302, pp. 757–763, 1986.

[94] E. Candes and M. Wakin, "People hearing without listening: An introduction to compressive sampling," *IEEE Signal Processing Magazine*, pp. 21–30, 2007.

[95] A. C. Gilbert, M. J. Strauss, and J. A. Tropp, "A tutorial on fast fourier sampling," *IEEE Signal processing magazine*, vol. 25, no. 2, pp. 57–66, 2008.

[96] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[97] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 96–102, Jul. 2021.

[98] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, T. Soleymani *et al.*, "Semantic communications in networked systems," *arXiv preprint arXiv:2103.05391*, 2021.

[99] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, "Universal transformers," in *Proc. Int'l. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019.

[100] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int'l. Conf. Learning Representations (ICLR)*, Austria, May 2021.

[101] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an

invariant mapping," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, vol. 2, New York, NY, USA, Jun. 2006, pp. 1735–1742.

[102] W. Chen, Y. Liu, W. Wang, E. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, "Deep image retrieval: A survey," *arXiv preprint arXiv:2101.11282*, Jan. 2021.

[103] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," in *Proc. Int'l. Conf. Learn. Representations (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–12.

[104] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 241–257.

[105] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1269–1277.

[106] C. Huang, S. Yang, Y. Pan, and H. Lai, "Object-location-aware hashing for multi-label image retrieval via automatic mask learning," *IEEE Trans. Image Processing*, vol. 27, no. 9, pp. 4490–4502, May 2018.

[107] E. W. Teh, T. DeVries, and G. W. Taylor, "ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, Glasgow, UK, Aug. 2020, pp. 448–464.

[108] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, "Training vision transformers for image retrieval," *arXiv preprint arXiv:2102.05644*, Feb. 2021.

[109] Y. Gu, S. Wang, H. Zhang, Y. Yao, W. Yang, and L. Liu, "Clustering-driven unsupervised deep hashing for image retrieval," *Neurocomputing*, vol. 368, pp. 114–123, Nov. 2019.

[110] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. Conf. Empir. Methods Natural Language Processing (EMNLP)*, Seattle, Washington, USA, Oct. 2013, pp. 1700–1709.

[111] F. Meng, Z. Lu, M. Wang, H. Li, W. Jiang, and Q. Liu, "Encoding source language with convolutional neural network for machine translation," in *Proc. of Annual*

Meeting of the Assoc. for Computational Linguistics and Int'l Joint Conf. Natural Language Processing of the Asian Fed. of Natural Language Processing, Beijing, China, Jul. 2015, pp. 20–30.

[112] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Aug. 1997.

[113] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[114] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, Dec. 2015.

[115] J. Kim, K. W. On, W. Lim, J. Kim, J. Ha, and B. Zhang, "Hadamard product for low-rank bilinear pooling," in *Proc. Int'l Conf. Learn. Representations (ICLR)*, Toulon, France, Apr. 2017.

[116] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, "Stacked attention networks for image question answering," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 21–29.

[117] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 6077–6086.

[118] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6281–6290.

[119] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *Proc. Int'l. Conf. Machine Learning (ICML)*, New York City, NY, USA, Jun. 2016, pp. 2397–2406.

[120] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, "End-to-end memory networks," in *Proc. Advances Neural Info. Process. Systems (NIPS)*, vol. 28, Montreal, Quebec, Canada, Dec. 2015.

[121] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, Sep. 2018.

[122] H. He, C. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, Feb. 2020.

[123] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Aug. 2018.

[124] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[125] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. Int'l. Conf. Acoustics Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 2326–2330.

[126] H. Ye, L. Liang, G. Y. Li, and B. Juang, "Deep learning based end-to-end wireless communication systems with conditional GAN as unknown channel," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3133–3143, Feb. 2020.

[127] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proc. Int'l. Conf. Machine Learning (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 531–540.

[128] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. Annual Meeting Assoc. Comput. Linguistics (ACL)*, Philadelphia, PA, USA, Jul. 2002, pp. 311–318.

[129] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int'l. Conf. Learning Representations (ICLR)*, Scottsdale, Arizona, USA, May 2013.

[130] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. North American Chapter of the Assoc. for Comput. Linguistics: Human Language Tech., (NAACL-HLT)*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[131] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT*

summit, vol. 5. Citeseer, Sep. 2005, pp. 79–86.

[132] C. Heegard and S. B. Wicker, *Turbo coding.* Springer Science & Business Media, 2013, vol. 476.

[133] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. the Society for Industrial and Applied Math.*, vol. 8, no. 2, pp. 300–304, Jan. 1960.

[134] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int'l Conf. Learn. Representations (ICLR)*, Banff, AB, Canada, Apr. 2014.

[135] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4004–4012.

[136] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[137] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. Int'l Conf. Comput. Vision Workshops (ICCV)*, Sydney, Australia, Dec. 2013, pp. 554–561.

[138] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1096–1104.

[139] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proc. Conf. Comput. Vision Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1988–1997.

[140] J. R. Smith, "Image retrieval evaluation," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998, pp. 112–113.

[141] H. V. Poor and S. Verdú, "Probability of error in MMSE multiuser detection," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 858–871, May 1997.

[142] J. Zuo, Q. Sun, and F. Zhao, "Computational complexities and relative performance of ldpc codes and turbo codes," in *Proc. Int'l. Conf. Software Engineering*

and Service Science, May, 2013, pp. 251–254.

[143] M. Tomasello, *Origins of human communication.* MIT press, 2010.

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

[145] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. PMLR Int'l. Conf. on Machine Learning (ICML)*, Lille, France, Jul. 2015, pp. 448–456.

[146] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.

[147] M. Boudiaf, J. Rony, I. M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed, "A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, UK, August 23–28, 2020, pp. 548–564.

[148] J. Weston, A. Bordes, S. Chopra, and T. Mikolov, "Towards ai-complete question answering: A set of prerequisite toy tasks," in *Proc. Int'l. Conf. Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

[149] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of Things: A review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, p. 1069, Jul. 2016.

[150] "IEEE standard for floating-point arithmetic," *IEEE Std 754-2008*, pp. 1–70, 2008.

[151] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication phy using neural network," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1364–1373, Mar. 2019.

[152] K. Thakkar, A. Goyal, and B. Bhattacharyya, "Deep learning and channel estimation," in *Proc. Int'l Conf. on Adv. Comput. and Commun. Systems (ICACCS)*, Coimbatore, India, Mar. 2020, pp. 745–751.

[153] E. Balevi, A. Doshi, and J. G. Andrews, "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, Jan. 2020.

[154] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Feb. 2017.

[155] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, “Attention-guided cnn for image denoising,” *Neural Netw.*, vol. 124, pp. 117–129, Apr. 2020.

[156] R. Dorrance, F. Ren, and D. Marković, “A scalable sparse matrix-vector multiplication kernel for energy-efficient sparse-blas on fpgas,” in *Proc. ACM/SIGDA Int'l sym. Field-programmable gate arrays*, Feb. 2014, pp. 161–170.

[157] L. Zhuo and V. K. Prasanna, “Sparse matrix-vector multiplication on fpgas,” in *Proc. ACM/SIGDA Int'l sym. Field-programmable gate arrays*, Feb. 2005, pp. 63–74.

[158] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” *arXiv preprint arXiv:1308.3432*, Aug. 2013.