Knowledge distillation based semantic communications for multiple users

Article

Accepted Version

Knowledge Distillation Based Semantic Communications For Multiple Users

Chenguang Liu, Yuxin Zhou, Yunfei Chen, *Senior Member, IEEE*,
Shuang-Hua Yang, *Senior Member, IEEE*

***Abstract*—Deep learning (DL) has shown great potential in revolutionizing the traditional communications system. Many applications in communications have adopted DL techniques due to their powerful representation ability. However, the learning-based methods can be dependent on the training dataset and perform worse on unseen interference due to limited model generalizability and complexity. In this paper, we consider the semantic communication (SemCom) system with multiple users, where there is a limited number of training samples and unexpected interference. To improve the model generalization ability and reduce the model size, we propose a knowledge distillation (KD) based system where Transformer based encoder-decoder is implemented as the semantic encoder-decoder and fully connected neural networks are implemented as the channel encoder-decoder. Specifically, four types of knowledge transfer and model compression are analyzed. Important system and model parameters are considered, including the level of noise and interference, the number of interfering users and the size of the encoder and decoder. Numerical results demonstrate that KD significantly improves the robustness and the generalization ability when applied to unexpected interference, and it reduces the performance loss when compressing the model size.**

***Index Terms*—Deep learning, knowledge distillation, model compression, multi-user interference, semantic communication, text transmission.**

I. Introduction

ACCORDING to Shannon and Weaver [1], communications can be categorized into three levels: transmission of symbols, transmission of semantics behind symbols, and effectiveness of semantics transmission. The first level aims to accurately transmit the symbols from the transmitter to the receiver by minimizing the bit error rate (BER) or the symbol error rate (SER). The second level semantic communication (SemCom) focuses on precisely conveying the meaning behind the bits. The third level concentrates on the effectiveness of the tasks that the communication intends to achieve over semantics transmission.

However, the limited spectrum resources constrain the capacity of traditional data communications at the first level, following the Shannon limit. To address this, SemCom extracts the meaning behind data and transmits only the essential semantic information, prioritizing semantic-level fidelity over bit-level accuracy. This is useful for applications requiring extensive data exchange but with limited bandwidth, where task effectiveness precedes exact information recovery. Potential applications include human-machine symbiosis, intelligent transportation, and extended reality (XR) [2]–[4]. For instance, the XR performance relies on processing the essential user data (*e.g.*, head movement, gestures and text input). Fast data transmission via low-latency networks to XR servers is vital for data processing and the corresponding tactile feedback. By filtering out the non-essential data with semantic understanding, SemCom allows end devices to transmit only pertinent data required for the operation at the XR server, thereby reducing bandwidth requirement and computational costs on the XR server. To enable such functions, it is crucial to investigate effective techniques for extracting semantic information.

Recently, deep learning (DL) has been widely applied to address problems in natural language processing and computer vision due to their powerful pattern recognition and representation capacity. Inspired by this, several works have been conducted to explore the DL-enabled SemCom systems for text [5]–[9], image [10]–[13] and speech transmission [9]. Different channel conditions are considered, including additive white Gaussian noise, Rician fading, and Rayleigh fading. However, it has yet to be studied whether a SemCom system, well-suited for end-to-end (E2E) communications [7], can effectively function in the presence of multi-user (MU) interference. MU interference, such as co-channel interference, is usually caused by multiple radios transmitting on the same frequency simultaneously due to the overly crowded spectrum [14]. Although interference can be avoided or mitigated through effective resource management [15], [16], these come at the expense of system complexity and resource utilization efficiency. When there is a high user density or uncontrolled interference sources, eliminating MU interference may not be practical. Allowing interference to co-exist with SemCom systems without significantly degrading the system performance could be a simple but effective way to overcome this challenge. Therefore, it is necessary to evaluate the SemCom system

This work is supported in part by EC H2020 DAWN4IoE-Data Aware Wireless Network for Internet-of-Everything under Grant 778305, the King Abdullah University of Science and Technology Research Funding (KRF) under Award ORA-2021-CRG10-4696, the National Natural Science Foundation of China (Grant No. 92067109, 61873119, 62211530106), and Shenzhen Science and Technology Program (Grant No. ZDSYS20210623092007023, GJHZ20210705141808024). (*Corresponding author: Shuang-Hua Yang.)*

Chenguang Liu is with the School of Engineering, University of Warwick, Coventry, UK, CV4 7AL. e-mail: Chenguang.Liu@warwick.ac.uk

Yuxin Zhou is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. e-mail: zhouyx2020@mail.sustech.edu.cn

Yunfei Chen is with the Department of Engineering, University of Durham, Durham, UK, DH1 3LE. e-mail: Yunfei.Chen@durham.ac.uk

Shuang-Hua Yang is with Shenzhen Key laboratory of Safety and Security for Next Generation of Industrial Internet, Southern University of Science and Technology, Shenzhen, China, and also with Department of Computer Science, University of Reading, UK. e-mail: yangsh@sustech.edu.cn

quantitatively and whether the learning-based techniques can achieve a tolerable performance in the presence of MU interference.

Moreover, the work in [17] has emphasized that researchers should focus on not only applying the existing DL techniques to the improvement of the current communications system but also considering the requirements and constraints of the communications network, such as low model complexity and low power consumption for low-power chips, to enable learning and data-driven distributed mobile devices. The semantic information varies with the transmission task, such as a highly convoluted image feature map or compressed text embeddings [9]. Therefore, we need a powerful model to mine the deeper information hidden in the raw training data and understand the relationship of the transmitted words. This typically requires a model designed with a complex and deep structure and an extensive amount of training data to cultivate the generalization ability. However, when it comes to deployment, a lighter model is preferred due to the constraints of computation complexity and time. To achieve this, knowledge distillation (KD) can be used to keep the light size and the generalizability, which was first proposed in [18] to compress the knowledge from an intelligent ensemble of models into a single light model. KD is widely utilized to reduce the size and improve the generalization for language understanding tasks [19], [20]. Nevertheless, KD has not yet been applied to SemCom.

A. Related work

To explore how learning-based algorithms can transform the communications system for lower complexity and better performance, representative works were conducted in [17], [21]–[23]. The challenges and opportunities of machine learning in communications were reviewed and discussed in [17] for the physical layer. The future research directions powered by the data-driven and learning approaches were pointed out. The recent advances in applying DL in the physical layer were demonstrated in [21], [22] to provide potential research directions for intelligent learning-based communications. The work in [24] proposed a deep neural network for multi-input multi-output (MIMO) detection in different channel conditions, which has near-optimal performance with perfect channel state information (CSI). To address the channel distortion, a fully connected neural network was proposed for channel estimation and signal detection in the OFDM system, which has comparable system performance with the minimum mean-square error estimator [25]. The work in [26] demonstrated that the learning-based detector could perform without knowledge of CSI by proposing a sliding bidirectional recurrent neural network to detect the signals. Moreover, the effect of co-channel interference and radar interference on learning-based detectors were analyzed in [27] and [28], respectively.

Unlike the aforementioned works that only optimize and deploy the learning-based receiver, several works have been conducted to jointly optimize the transmitter and receiver. For example, an E2E learning-based communications system was proposed in [21] using autoencoders to replace the traditional transmitter and receiver, to significantly reduce the complexity of design and implementation compared with traditional block-wise communications systems. Recently, research has shifted from transmission of symbols to transmission of semantic meaning inspired by the significant advancements of DL in natural language processing. The scalability and capacity of DL enables semantic understanding on deeper information, such as word meaning and word relations in text transmission, to improve the system performance. A comprehensive overview was conducted in [23] on how the communications system can benefit from semantic and goal-oriented communications in terms of effectiveness and sustainability. This overview strengthens the idea that recovering the meaning behind the bits or completing the task that the transmission intends to achieve is key to the recovery of the transmitted information at the receiver. Understanding the meaning or the goals behind the bits requires that the coding and decoding schemes can identify the internal relationship of the transmitted information.

There have been several studies on DL-enabled SemCom including text transmission [5]–[8], image transmission [10]–[13], speech transmission [9] and task-oriented transmission [29]. The work in [5] proposed a joint source-channel coding (JSCC) communication framework using recurrent neural network for text transmission, which had lower word error rates than conventional coding schemes. By mapping the words in a semantic space, words with similar meaning can have close distance. Then, a DL-enabled SemCom system, DeepSC, was proposed in [6], [7] to use Transformer as the semantic encoder and decoder, to outperform traditional coding schemes, especially in low SNR regime. Additionally, a lite SemCom system, L-DeepSC, was proposed in [8] for distributed IoT devices, which used parameter pruning and quantization to reduce the model size so that it can work with the limited bandwidth and transmission conditions. Moreover, the work in [9] proposed an attention-based residual network as the joint transceiver for speech signals, which showed better robustness and performance than the traditional benchmarks with regard to the speech signal metrics. A JSCC was proposed for wireless image compression and transmission using two convolutional neural networks [10]. The work in [11] proposed a practical JSCC scheme based on autoencoder taking channel output feedback into account to improve the image reconstruction quality. The work in [12] proposed an iterative source-channel decoder to explicitly consider residual bit error of each iteration for image transmission. The work in [13] proposed coarse-to-fine image semantic coding model for multimedia SemCom system using generative adversarial networks. Apart from joint source-channel coding schemes for texts, images and speech, a task-oriented MU SemCom, MU-DeepSC, was designed to deal with multi-modal data [29].

B. Motivation and contribution

Although all the previous works have demonstrated novelty and satisfactory performance by adopting DL-based semantic systems for robustness and effectiveness, there has not been any works on applying KD to the SemCom system with MU interference. MU interference of the communications system

is inevitable in practice due to spectrum sharing. Yet, it is either not studied or ignored in the previous works. Moreover, revolutionizing the traditional communications system with a DL-enabled source-channel coding scheme still has to overcome many practical problems, including model generalizability and complexity. Specifically, the learning-based model can be overly dependent on the training dataset samples and consequently perform worse on unseen data. Several questions need to be addressed in DL-based SemCom systems:

1) How well can the model generalize on unseen interference?
2) How well can the model perform by training with limited dataset?
3) How light can the model be with negligible performance loss?

In this paper, our work focuses on the SemCom system using KD to improve the model generalization capacity and lower the model complexity. The main contributions of this paper can be summarized as below:

1) To the best of the authors' knowledge, this is the first work that applies KD to the DL-enabled SemCom system with MU interference. The random occurrences and delays are considered for the interference. The performance of this system is evaluated for different signal-to-noise ratios (SNRs), signal-to-interference ratios (SIRs), and numbers of interfering users.
2) We propose four types of KD approaches by training the distilled student models for a limited range of SNR regimes in the absence of interference samples and then applying them to a wider range of SNR regimes with unseen MU interference. Numerical results show that distilled models outperform the non-distilled baselines and the conventional communications system with and without interference. Furthermore, it is proved that KD largely improves the generalizability and robustness of the model, which address Question 1) and 2) mentioned before.
3) By adopting model compression in KD after training, we apply model compression to the semantic encoder-decoder and the channel encoder-decoder to reduce performance loss. We also demonstrate the complexity and performance analysis in terms of the size and number of parameters, which address Question 3). Additionally, an ablation study is conducted to analyze the effect of various losses on the distilled student models.

The rest of the paper is organized as follows. In Section II, we will introduce the system model of the SemCom system with MU interference and describe the main challenges. Section III will discuss the KD-based SemCom, model compression and training process. Simulation settings and numerical results will be demonstrated in Section IV. Finally, Section V will conclude the work.

Notation: To represent the parameters and outputs from the interfering user, we use the superscript $\mathcal{I}$ to represent the interfering user. Also, we use the superscript $\mathcal{T}$ and $\mathcal{S}$ to represent the parameters and outputs from the teacher model and the student model in the distillation training process. $\mathbb{E}$ denotes the mean of each element in the vector.

II. System Model and Problem Formulation

In this section, we will firstly describe the SemCom system with MU interference. Then, we will point out the challenges that DL-based SemCom might encounter, including generalizing on unseen data, limited training data and model complexity.

A. SemCom system

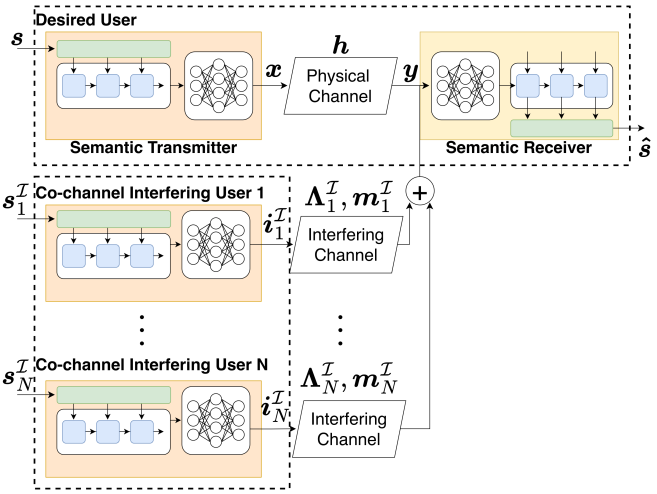


Fig. 1: SemCom with MU interference due to co-channel competition.

As shown in Fig. 1, we consider a SemCom system with multiple users, where all users are equipped with a semantic transmitter and compete for the same channel to incur co-channel interference. In this MU system, each user could be the desired user or cause interference to other users. When they transmit signals simultaneously, they can interfere with each other. Therefore, we model this SemCom system to have one desired user in the presence of co-channel interference from N interfering users. Note that the interference in this paper refers to the signals transmitted from interfering users to the desired user. Each interference has random occurrences and delays in the transmission. Moreover, each desired user has a transmitter-receiver pair with one semantic encoder and channel encoder at the transmitter, one channel decoder and a semantic decoder at the receiver. The semantic encoder and decoder are responsible for compressing and extracting the information from the source at the semantic level. The channel encoder and decoder are designed to counteract the channel effect and recover the encoded semantic information.

We focus on a text transmission task using the SemCom system in the presence of MU interference and noise. The text input is expressed as $\boldsymbol{s} = [w_1, w_2, ..., w_n]$, where w_i denotes the i-th word in the sentence $\boldsymbol{s}$. Then, each word is successively encoded by the semantic encoder and channel encoder to formulate the transmitted symbols $\boldsymbol{x}$ as

$$\boldsymbol{p} = \mathcal{SE}(\boldsymbol{s}, \boldsymbol{\alpha}), \tag{1}$$

$$\boldsymbol{x} = \mathcal{CE}(\boldsymbol{p}, \boldsymbol{\beta}), \tag{2}$$

where $\boldsymbol{s}$ is the text sentence as input, $\boldsymbol{p}$ denotes the semantic encoded information, $\mathcal{SE}(\cdot)$ and $\mathcal{CE}(\cdot)$ is the semantic encoder and the channel encoder with parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively, and $\boldsymbol{x}$ is the transmitted symbols as output.

Consider a SemCom system with one desired user and multiple interfering users equipped with semantic transmitters, the interference signals transmitted by the k-th interfering user can be expressed by

$$\boldsymbol{i}_k^{\mathcal{I}} = \mathcal{CE}_k^{\mathcal{I}}(\mathcal{SE}_k^{\mathcal{I}}(\boldsymbol{s}_k^{\mathcal{I}}, \boldsymbol{\alpha}_k^{\mathcal{I}}), \boldsymbol{\beta}_k^{\mathcal{I}}), \tag{3}$$

where $\boldsymbol{i}_k^{\mathcal{I}}$ is the interfering symbols from the k-th interfering user. Then, the signals transmitted by the interfering users arrive at the receiver of the desired user as,

$$\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x} + \sum_{k=1}^{N} \boldsymbol{\Lambda}_k^{\mathcal{I}} * \boldsymbol{m}_k^{\mathcal{I}} * \boldsymbol{i}_k^{\mathcal{I}} + \boldsymbol{n}, \tag{4}$$

$$\boldsymbol{\Lambda}_k^{\mathcal{I}} = [\lambda_{k,1}, \lambda_{k,2}, \cdots, \lambda_{k,n}], \tag{5}$$

$$\lambda_{k,j} = \begin{cases} 1 & \text{if the interference symbol occurs} \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where $\boldsymbol{h}$ and $\boldsymbol{m}_k^{\mathcal{I}}$ are the physical channel and the k-th interfering channel following Gaussian distributions, respectively, $\boldsymbol{\Lambda}_k$ denotes the random occurrence of the k-th interference, $\lambda_{k,n}$ is the binary interference occurrence indicator for j-th symbol in the k-th interference, $\boldsymbol{n}$ denotes the additive white Gaussian noise with mean zero and variance σ^2 and the operation $*$ denotes the element-wise multiplication. To decode the received signals for the desired user, the decoding process can be expressed by,

$$\hat{\boldsymbol{p}} = \mathcal{CD}(\boldsymbol{y}, \boldsymbol{\gamma}), \tag{7}$$

$$\hat{\boldsymbol{t}} = \mathcal{SD}(\hat{\boldsymbol{p}}, \boldsymbol{\delta}), \tag{8}$$

where $\mathcal{CD}(\cdot)$ and $\mathcal{SD}(\cdot)$ are the channel decoder and semantic decoder with parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$, respectively; $\boldsymbol{p}$ is the decoded channel information, which is also the input to the semantic decoder, $\hat{\boldsymbol{t}}$ denotes the decoded semantic information. Finally, a dense layer with softmax activation function is applied as the prediction layer to estimate the predicted sentence $\hat{\boldsymbol{s}}$ from the semantic decoded information $\hat{\boldsymbol{t}}$, which can be expressed by

$$\begin{aligned} \hat{\boldsymbol{s}} &= \mathcal{F}_{pred}(\hat{\boldsymbol{t}}, \boldsymbol{w}_{pred}, \boldsymbol{b}_{pred}) \\ &= Softmax(\boldsymbol{w}_{pred}\hat{\boldsymbol{t}} + \boldsymbol{b}_{pred}), \end{aligned} \tag{9}$$

where $\boldsymbol{w}_{pred}$ and $\boldsymbol{b}_{pred}$ are the prediction layer parameters, $\hat{\boldsymbol{s}}$ denotes the predicted sentence.

The goal of the SemCom system is to recover the text sentence $\hat{\boldsymbol{s}}$ in the presence of interference and noise. In order to explore the generalization ability of the SemCom system, we assume the receiver has the perfect CSI of the desired-user channel gain $\boldsymbol{h}$ but no knowledge of the interfering channels. Then, the perfect CSI $\boldsymbol{h}$ is adopted by zero-forcing detector at the receiver to obtain the recovered signals $\hat{\boldsymbol{x}}$ from the received signal $\boldsymbol{y}$. The reason for this assumption is to focus on evaluating the proposed methods' performance in the presence of unseen interference rather than the effect of channel estimation errors or other practical limitations. The extension to the case with both interference and channel estimation error is not studied here due to space limit. Besides, we consider Rayleigh fading, random interference occurrences and transmission delay for the MU interference.

B. Problem description

The cross-entropy loss is utilized to measure the difference between the ground truth hard labels and the predicted text sentence, which can be expressed by

$$\begin{aligned} \mathcal{L}_{hard} &= \mathcal{L}_{CE}(\boldsymbol{s}, \hat{\boldsymbol{s}}) \\ &= -\frac{1}{n}\sum_{m=1}^{n} P(\boldsymbol{s}[m]) \log P(\hat{\boldsymbol{s}}[m]), \end{aligned} \tag{10}$$

where $\boldsymbol{s} = [w_1, w_2, ..., w_n]$, $P(\boldsymbol{s}[m])$ is the probability for the real m-th word w_m in the text sentence $\boldsymbol{s}$, and $P(\hat{\boldsymbol{s}}[m])$ is the probability for the predicted m-th word $\hat{w}_m$ in the text sentence $\hat{\boldsymbol{s}}$. Semantic transmitter and receiver are jointly optimized by adjust their parameters sets to minimize the loss considering physical channel attenuation, interference and noise. However, this training method also brings several challenges.

The first challenge is the training of the semantic transmitter and receiver for generalization ability. The cross-entropy loss function only takes the final output of the SemCom system into account, so that the semantic encoder, channel encoder, channel decoder and semantic decoder are jointly trained as one black box. Although the SemCom system technically has a semantic transmitter and a semantic receiver, it is difficult to interpret the intermediate output as it is part of the convergence. Consequently, the model can easily overfit on the training dataset or under-trained. This is inevitable for data-driven and learning-based systems, where the quality of the trained model cannot be guaranteed unless we conduct extensive experiments on all unseen data to validate its generalization ability. In practice, this may lead to excessive resources for training and testing. Therefore, it is important to design a training method so that the model can have considerable performance and generalization ability with unseen data.

The second challenge is the limitation of the training data. The data-driven and learning-based communications system highly relies on the training dataset to maintain the model performance. An extensive amount of datasets containing sufficient patterns lays the foundation for training a powerful learning-based communications system. However, it can be challenging in a practical communications system to obtain the ideal dataset which meets such requirements. Therefore, it is important to train the model properly with fewer dataset.

The third challenge is the model complexity. A complex model normally possesses a high convergence ability to accurately learn and approximate the relationship between inputs and outputs for a given dataset and generalizes well to unseen data. Nevertheless, the computational complexity can be too high for devices with limited computation resources. Therefore, it can be very challenging to reduce the model complexity while preserving its convergence. Next, we will address these challenges.

III. KD-based SemCom system

In this section, we will introduce the Transformer-based semantic transceiver. Then, we will introduce a KD-based

SemCom system to address the challenges mentioned above. Different model compression algorithms are used to address Question 3), and KD is adopted to solve Questions 1) and 2). Finally, the training procedure will be demonstrated.

A. Transformer based semantic transceiver

Inspired by the bidirectional encoder representations from Transformers (BERT) [30], we adopt the Transformer structure as the semantic encoder and decoder to compress and extract the semantic information. The attention scheme of the Transformer can correlate the contextual information for each word in the sentence. The attention can be computed by

$$\boldsymbol{Q} = \boldsymbol{W}^{Q}\boldsymbol{X}^{Q}, \quad \boldsymbol{K} = \boldsymbol{W}^{K}\boldsymbol{X}^{K}, \quad \boldsymbol{V} = \boldsymbol{W}^{V}\boldsymbol{X}^{V}, \tag{11}$$

$$\mathcal{F}_A(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V}, \tag{12}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V} \in \mathbb{R}^{B\times L\times D_{model}}$ denote the representations of the query, the key and the value with the input $\boldsymbol{X}^Q$, $\boldsymbol{X}^K$ and $\boldsymbol{X}^V \in \mathbb{R}^{B\times L\times D_{model}}$ and the parameter $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$, respectively, B is the batch size, L is the sequence length and D_{model} is the embedding dimension, $\mathcal{F}_A(\cdot)$ denotes the attention function and d_k is the scaling factor. To obtain the information from different representation subspaces at different positions, the multi-head attention is used to calculate the attention in parallel and then concatenate the independent attention. The multi-head attention with N heads can be computed by

$$\boldsymbol{Q}_i = \boldsymbol{Q}\boldsymbol{W}_i^{Q}, \quad \boldsymbol{K}_i = \boldsymbol{K}\boldsymbol{W}_i^{K}, \quad \boldsymbol{V}_i = \boldsymbol{V}\boldsymbol{W}_i^{V}, \tag{13}$$

$$\mathcal{F}_{MA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = [\mathcal{F}_{\mathcal{A}}(\boldsymbol{Q}_1, \boldsymbol{K}_1, \boldsymbol{V}_1)|| \cdots ||\mathcal{F}_{\mathcal{A}}(\boldsymbol{Q}_N, \boldsymbol{K}_N, \boldsymbol{V}_N)]\boldsymbol{W}^0, \tag{14}$$

where $\mathcal{F}_{MA}(\cdot)$ is the multi-head attention function with parameter $\boldsymbol{W}^0$, $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$ and $\boldsymbol{V}_i \in \mathbb{R}^{B\times L\times D_{head}}$ are the query, the key and the value of the i-th head with parameters $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^V \in \mathbb{R}^{D_{model}\times D_{head}}$, $D_{head} = D_{model}/N$ is the embedding dimension of each head and $||$ denotes the concatenation operation. Then, the feed forward layer is applied to the output of the multi-head attention layer, which is expressed as,

$$\mathcal{F}_{FF}(\boldsymbol{X}) = \boldsymbol{W}_{\mathcal{FF}}\boldsymbol{X} + \boldsymbol{b}_{\mathcal{FF}}, \tag{15}$$

where $\boldsymbol{W}_{\mathcal{FF}}$ and $\boldsymbol{b}_{\mathcal{FF}}$ are the parameters of the feed forward layer. Layer normalization is applied to each output from the multi-head attention layer and the feed forward layer to rescale and shift the outputs, which can be described by

$$\mathcal{F}_{LN}(\boldsymbol{X}) = \frac{\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}]}{\sqrt{\sigma_{\boldsymbol{X}}^2 + \epsilon}}\theta + \mu, \tag{16}$$

where $\boldsymbol{X} \in \mathbb{R}^{D_{model}}$ is the input of the layer normalization, θ and μ are the trainable parameters, $\sigma_{\boldsymbol{X}}$ is the variance of the input $\boldsymbol{X}$, ϵ is an arbitrarily small number. Also, we apply skip connection using addition by adding the output from the preceding layer to the layer ahead.

For the Transformer based semantic transmitter, the text sequence $\boldsymbol{s}$ is preprocessed by a text tokenizer to split the text into words by punctuation and whitespaces, and then map each word with the corresponding scalar number according to the word representation dictionary. Then, the text sequence $\boldsymbol{s}$ is embedded as $\boldsymbol{t} \in \mathbb{R}^{B\times L\times D_{model}}$, which is the input to the Transformer layer. Each layer of the Transformer based semantic encoder contains a multi-head self-attention layer and a feedforward layer processed by residual connection and layer normalization. The Transformer layer of the semantic encoder can be expressed by,

$$\boldsymbol{z}_{self} = \mathcal{F}_{LN}(\mathcal{F}_{MA}(\boldsymbol{W}^{Q}\boldsymbol{t}, \boldsymbol{W}^{K}\boldsymbol{t}, \boldsymbol{W}^{V}\boldsymbol{t}) + \boldsymbol{t}), \tag{17}$$

$$\boldsymbol{p} = \mathcal{F}_{LN}(\mathcal{F}_{FF}(\boldsymbol{z}_{self}) + \boldsymbol{z}_{self}), \tag{18}$$

where $\boldsymbol{z}_{self}$ is the output of the multi-head self-attention processed by layer normalization and residual connection with embeddings $\boldsymbol{t}$ as input, $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ are the weights parameters, $\boldsymbol{p}$ denotes the semantic encoded information in (1) and also the output of the Transformer layer with a size determined by sentence length and output units of the semantic encoder.

For the channel encoder and decoder, fully connected dense layers are used to encode and recover the information from the corrupted channel condition, which can be expressed as,

$$\mathcal{F}_{FC}(\boldsymbol{X}) = \rho(\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}), \tag{19}$$

$$\boldsymbol{x} = \mathcal{F}_{FC}(\boldsymbol{p}, \boldsymbol{\beta}), \tag{20}$$

$$\hat{\boldsymbol{p}} = \mathcal{F}_{FC}(\hat{\boldsymbol{x}}, \boldsymbol{\gamma})), \tag{21}$$

where $\mathcal{F}_{FC}(\cdot)$ denote the fully connected layer with input $\boldsymbol{X}$ and parameters $\boldsymbol{W}$ and $\boldsymbol{b}$, ρ is the activation function, $\boldsymbol{x}$ denotes the channel encoded information in (2) with semantic encoded information $\boldsymbol{p}$ as input and $\boldsymbol{\beta}$ as trainable parameters, $\hat{\boldsymbol{p}}$ denotes the channel decoded information with recovered symbols $\hat{\boldsymbol{x}}$ at receiver as input and $\boldsymbol{\gamma}$ as trainable parameters.

Different from the semantic encoder, where the attention is calculated instantaneous for the entire sequence, the semantic decoder estimates the sequence by iteratively predicting each word sequentially using the previous estimate as input. Therefore, an additional self-attention for the predicted words of each iteration is required for the semantic decoder. The Transformer layer of the semantic decoder can be expressed as

$$\boldsymbol{z}' = \mathcal{F}_{LN}(\mathcal{F}_{MA}(\boldsymbol{W}'^{Q}\hat{\boldsymbol{t}}', \boldsymbol{W}'^{K}\hat{\boldsymbol{t}}', \boldsymbol{W}'^{V}\hat{\boldsymbol{t}}') + \hat{\boldsymbol{t}}'), \tag{22}$$

$$\boldsymbol{z}_{cross} = \mathcal{F}_{LN}(\mathcal{F}_{MA}(\boldsymbol{W}^{Q}\boldsymbol{z}', \boldsymbol{W}^{K}\hat{\boldsymbol{p}}, \boldsymbol{W}^{V}\hat{\boldsymbol{p}}) + \boldsymbol{z}'), \tag{23}$$

$$\hat{\boldsymbol{t}} = \mathcal{F}_{LN}(\mathcal{F}_{FF}(\boldsymbol{z}_{cross}) + \boldsymbol{z}_{cross}), \tag{24}$$

where $\boldsymbol{z}'$ denotes the multi-head self-attention with the predicted embeddings $\hat{\boldsymbol{t}}'$ as input, $\hat{\boldsymbol{t}}' \subseteq \hat{\boldsymbol{t}}$ denotes the results of each iteration to estimate the sequence, $\boldsymbol{W}'^{Q}$, $\boldsymbol{W}'^{K}$ and $\boldsymbol{W}'^{V}$ denote the parameters for the self-attention of the previous predictions, $\boldsymbol{z}_{cross}$ is the multi-head cross-attention with attention of the previous predictions $\boldsymbol{z}'$ and channel decoded information $\hat{\boldsymbol{p}}$ as input, $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ denote the parameters for the multi-head cross-attention, $\hat{\boldsymbol{p}}$ is the output of the channel decoder in (7), $\hat{\boldsymbol{t}}$ is the output of the semantic decoder in (8).

In training, we use the masked sequence embedding $\hat{\boldsymbol{t}}_{masked}$ as input to predict the masked word instead of using the output of the previous predictions $\hat{\boldsymbol{t}}'$ to predict the next word. The attention mechanism can learn the contextual information around the masked word during training. This could speed

up the training and address the problems that the model cannot make reliable predictions when under-trained. During the deployment and testing, we always use the predictions of the previous iterations to estimate the next. Note that the semantic encoder-decoder and the channel encoder-decoder can have multiple layers, which are iterated by using the output of the current layer as the input for the next.

B. KD-based system model

As shown in Fig. 2, we propose a tailored KD algorithm for the SemCom system. In KD, the teacher is required to provide soft targets as knowledge to train the student model. The soft targets are the probability distribution obtained by applying the softmax function to the output of the model. By adding the temperature parameter T in the softmax, it could control the level of uncertainty in the output probabilities. By raising the temperature, the soft targets become more diffuse with less emphasis on the most probable class, which can help prevent the student model from overfitting to the training data and encourage it to learn more generalizable features [18]. The softmax function with temperature T can be described by

$$Q(z_i;T) = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}, \tag{25}$$

where z_i is the i-th output of the model which can be the final prediction or the intermediate feature representation, and T denotes the temperature parameter. To transfer knowledge from teacher to student, instead of only considering the final output logits, we consider the intermediate outputs in the SemCom system including the encoded semantic information $\boldsymbol{p}^{\mathcal{T}}$, $\boldsymbol{p}^{\mathcal{S}}$, the encoded channel information $\boldsymbol{x}^{\mathcal{T}}$, $\boldsymbol{x}^{\mathcal{S}}$, the decoded channel information $\hat{\boldsymbol{p}}^{\mathcal{T}}$, $\hat{\boldsymbol{p}}^{\mathcal{S}}$, the decoded semantic information $\hat{\boldsymbol{t}}^{\mathcal{T}}$, $\hat{\boldsymbol{t}}^{\mathcal{S}}$ and the final predictions $\hat{\boldsymbol{s}}^{\mathcal{T}}$, $\hat{\boldsymbol{s}}^{\mathcal{S}}$. In this way, teachers' intermediate outputs can be used as additional supervisory data to guide student training and as a comparative benchmark to implicitly explain students' intermediate outputs in the SemCom system. Taking semantic encoded information as an example, tests show that a closer distribution to the teacher model could enhance the capability of the student model.

The total distillation loss is computed by

$$\begin{aligned}\mathcal{L}_{total\_distill} &= \sum_{(O^{\mathcal{S}},O^{\mathcal{T}})\in\mathbb{O}} \mathcal{L}_{distill}(O^{\mathcal{S}},O^{\mathcal{T}}) \\ &= \sum_{(O^{\mathcal{S}},O^{\mathcal{T}})\in\mathbb{O}} \mathbb{E}\Big\{\eta_{O^{\mathcal{S}},O^{\mathcal{T}}} T^2 \mathcal{D}_{KL}[Q(O^{\mathcal{S}};T)||Q(O^{\mathcal{T}};T)]\Big\},\end{aligned} \tag{26}$$

$$\mathbb{O} \subseteq \Big\{(\boldsymbol{p}^{\mathcal{S}},\boldsymbol{p}^{\mathcal{T}}),(\boldsymbol{x}^{\mathcal{S}},\boldsymbol{x}^{\mathcal{T}}),(\hat{\boldsymbol{p}}^{\mathcal{S}},\hat{\boldsymbol{p}}^{\mathcal{T}}),(\hat{\boldsymbol{t}}^{\mathcal{S}},\hat{\boldsymbol{t}}^{\mathcal{T}}),(\hat{\boldsymbol{s}}^{\mathcal{S}},\hat{\boldsymbol{s}}^{\mathcal{T}})\Big\}, \tag{27}$$

where $\mathcal{L}_{distill}(\cdot)$ is the corresponding distillation loss of each output from the teacher and the student; $O^{\mathcal{T}}$ denotes the output from the teacher serving as the reference to the output from the student $O^{\mathcal{S}}$; $O^{\mathcal{T}}$ and $O^{\mathcal{S}}$ are the subset of the distillation information set $\mathbb{O}$; $\eta_{O^{\mathcal{S}},O^{\mathcal{T}}}$ is the weight parameter for each distillation loss; $\mathcal{D}_{KL}(\cdot)$ denotes the Kullback-Leibler (KL) divergence [31]. KL divergence can compare two probability distributions with different scales, which provides a way to measure how much the student distribution deviates from the teacher distribution. Minimizing this difference enables the student to learn from the teacher and reproduce the teacher's probability distribution. Additionally, KL divergence allows the temperature parameter T to be adjusted, which improves the flexibility of the training. To obtain the overall training loss for the student model, we combine the cross entropy loss with the hard labels and the sum of the distillation loss, which can be represented as,

$$\begin{aligned}\mathcal{L}_{overall} &= (1 - \sum_{(O^{\mathcal{T}},O^{\mathcal{S}})\in\mathbb{O}} \eta_{O^{\mathcal{T}},O^{\mathcal{S}}}) \mathcal{L}_{CE}(\boldsymbol{s}^{\mathcal{S}},\hat{\boldsymbol{s}}^{\mathcal{S}}) \\ &+ \sum_{(O^{\mathcal{T}},O^{\mathcal{S}})\in\mathbb{O}} \mathbb{E}\Big\{\eta_{O^{\mathcal{T}},O^{\mathcal{S}}} T^2 \mathcal{D}_{KL}(Q(O^{\mathcal{S}};T)||Q(O^{\mathcal{T}};T))\Big\}.\end{aligned} \tag{28}$$

Then, the gradient of the overall loss is calculated and back-propagated to update the parameters. Therefore, the error between the teacher's output probability distributions and student output distributions can be minimized. In other words, the optimization for the student can be guided by the teacher model, which is equivalent to matching the corresponding outputs of each part from the teacher model to the student model [18].

C. Model compression

The proposed SemCom system adopts the Transformer structure as the semantic encoder-decoder and dense layers as the channel encoder-decoder. Unlike the conventional learning-based system that is treated as a black box, the training of the proposed KD-based SemCom system can be divided into several small black boxes based on the distilled knowledge. Each black box can converge the corresponding outputs from the pre-trained teacher model. This process can potentially have more control on the optimization process of the student model. Therefore, we can conduct the model compression by reducing the size of each component in the student model while converging the outputs, as shown in Fig. 2. However, simply reducing the number of Transformers in the semantic encoder-decoder could weaken the robustness of the model. To alleviate this nagative effect, the student learns from the teacher by mimicking the teacher's outputs to improve generalization ability since the teacher model is over-parameterized and pre-trained with extensive data. This is achieved by minimizing $\mathcal{L}_{overall}$ in equation (28). Similarly, the number of dense layers in the channel encoder-decoder can also be reduced for model compression. Although this might affect the information recovery performance, the knowledge from the teacher is transferred to compensate for the impact on the performance.

Also, inspired by the work in [8] using network quantization for the SemCom system, we propose to use post training dynamic quantization to further compress our model after reducing the number of layers and parameters via KD. Dynamic quantization converts the float representation of the weights to the reduced integer representation, which essentially saves

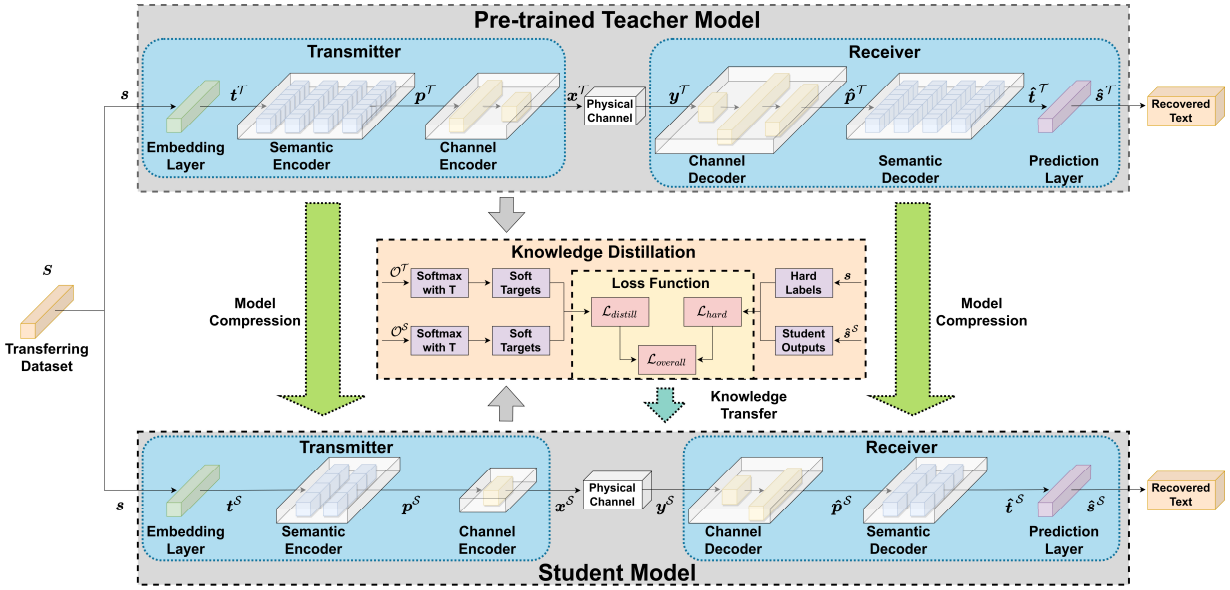


Fig. 2: The structure of the KD-based SemCom system.

the model size and computational complexity. The weights quantization can be expressed as,

$$\boldsymbol{X}_q = round(\varphi \boldsymbol{X}_{float} - \omega), \tag{29}$$

where $\boldsymbol{X}_q$ is the quantized output, $\boldsymbol{X}_{float}$ is the float input, φ is the scale parameter and ω is the zero point. Note that, overly decreasing the size of the mode could cause the model to diverge and lose generalization ability. The trade-off between performance and the size of the model will be discussed to answer Question 3).

D. Training

To train the KD-based SemCom system, the combined training process is demonstrated in Algorithm 4. It can be divided into three phases: training the teacher model, training the student model and applying post-training quantization. Algorithm 1 demonstrates the feed forward process to generate the outputs for teacher and student models. Since the proposed SemCom system adopts a DL-based E2E transceiver design, the feed-forward process is implemented as the complete process of encoding the information at the transmitter and recovering it at the receiver. In order to generate inputs for the semantic encoder, the dataset is partitioned into batches for parallel processing during training, and each word is then transformed into word embeddings for use as inputs. The semantic encoder consists of multiple Transformer layers with the multi-head self-attention in (17) and (18), while the semantic decoder additionally computes the multi-head cross-attention to account for previous predictions, denoted in (22), (23) and (24). For channel encoder and decoder, it adopts multiple fully connected dense layers in (19). As the teacher model is pre-trained, we assume that the teacher model has sufficient computational resources and large amount of training data. Also, we simulate different SNRs on the transmitted signals. With these samples in the training dataset, the teacher can have robust performance in the scenarios across different SNRs, with or without interference. The training algorithm for the teacher model is demonstrated in Algorithm 2. Firstly, the feed forward process is applied to generate the teacher's outputs, which introduces the interference signals from the interfering semantic transmitters. Afterwards, the cost is computed by calculating cross-entropy loss and propagating back to compute gradients. Then, the stochastic gradient descent is used to update the parameters in the semantic transceiver.

After obtaining the trained teacher model, we train the

Algorithm 1 Data generation

Input: : Dataset $\boldsymbol{S}$, number of interfering users N, SemCom model $\mathcal{SE}(\cdot)$, $\mathcal{CE}(\cdot)$, $\mathcal{CD}(\cdot)$, $\mathcal{SD}(\cdot)$ and $\mathcal{P}_{pred}(\cdot)$ with parameters $\Theta = \{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}, \boldsymbol{w}_{pred},\boldsymbol{b}_{pred}\}$

$\boldsymbol{s} \leftarrow BatchDataset(\boldsymbol{S})$.
$\boldsymbol{t} \leftarrow Embedding(\boldsymbol{s})$.
Compute the output of semantic encoder by (17) and (18), $\boldsymbol{p} \leftarrow \mathcal{SE}(\boldsymbol{t}, \boldsymbol{\alpha})$.
Compute the output of channel encoder by (19) and (20), $\boldsymbol{x} \leftarrow \mathcal{CE}(\boldsymbol{p}, \boldsymbol{\beta})$.
if Train the teacher model **then**
  Transmit $\boldsymbol{x}$ over the physical channel with MU interference $\boldsymbol{i}_1^I, ..., \boldsymbol{i}_N^I$ in (3).
else if Train the student model **then**
  Transmit $\boldsymbol{x}$ over the physical channel with no MU interference.
end if
Receive $\boldsymbol{y}$.
Compute the output of channel decoder by (19) and (21), $\hat{\boldsymbol{p}} \leftarrow \mathcal{CD}(\boldsymbol{y}, \boldsymbol{\gamma})$.
Compute the output of semantic decoder by (22), (23) and (24), $\hat{\boldsymbol{t}} \leftarrow \mathcal{SD}(\hat{\boldsymbol{p}}, \boldsymbol{\delta})$.
Compute the predicted results by (9), $\hat{\boldsymbol{s}} \leftarrow \mathcal{P}_{pred}(\hat{\boldsymbol{t}}, \boldsymbol{w}_{pred}, \boldsymbol{b}_{pred})$.

Output: $\boldsymbol{s}$, $\boldsymbol{p}$, $\boldsymbol{x}$, $\hat{\boldsymbol{p}}$, $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{s}}$.

Algorithm 2 Training algorithm of the teacher model

Input: Training dataset $\boldsymbol{S}^{\mathcal{T}}$, MU interfering model $\mathcal{SE}^{I}(\cdot)$, $\mathcal{CE}^{I}(\cdot)$, number of epochs E, Teacher model $\mathcal{SE}^{\mathcal{T}}(\cdot)$, $\mathcal{CE}^{\mathcal{T}}(\cdot)$, $\mathcal{CD}^{\mathcal{T}}(\cdot)$, $\mathcal{SD}^{\mathcal{T}}(\cdot)$ and $\mathcal{P}^{\mathcal{T}}_{pred}(\cdot)$ with parameters $\Theta^{\mathcal{T}} = \{\boldsymbol{\alpha}^{\mathcal{T}}, \boldsymbol{\beta}^{\mathcal{T}}, \boldsymbol{\gamma}^{\mathcal{T}}, \boldsymbol{\delta}^{\mathcal{T}}, \boldsymbol{w}^{\mathcal{T}}_{pred}, \boldsymbol{b}^{\mathcal{T}}_{pred}\}$.

1: Initialize parameters $\Theta^{\mathcal{T}}$.
2: **for** $e = 1$ to E **do**
3: Perform forward propagation to compute output $\hat{\boldsymbol{s}}^{\mathcal{T}}$.
4: Compute cost $J(\Theta^{\mathcal{T}})$ using loss function $\mathcal{L}_{hard}$ in (10).
5: Perform backward propagation to compute gradients $\frac{\partial J}{\partial \Theta^{\mathcal{T}}}$.
6: Update parameters $\Theta^{\mathcal{T}}$ using stochastic gradient descent.
7: **end for**

Output: Trained $\mathcal{SE}^{\mathcal{T}}(\cdot)$, $\mathcal{CE}^{\mathcal{T}}(\cdot)$, $\mathcal{CD}^{\mathcal{T}}(\cdot)$, $\mathcal{SD}^{\mathcal{T}}(\cdot)$ and $\mathcal{P}^{\mathcal{T}}_{pred}(\cdot)$ with parameters $\Theta^{\mathcal{T}}$.

Algorithm 3 Training algorithm of the student model

Input: Training dataset $\boldsymbol{S}^{\mathcal{S}}$, number of epochs E, pretrained teacher model $\mathcal{SE}^{\mathcal{T}}(\cdot)$, $\mathcal{CE}^{\mathcal{T}}(\cdot)$, $\mathcal{CD}^{\mathcal{T}}(\cdot)$, $\mathcal{SD}^{\mathcal{T}}(\cdot)$ and $\mathcal{P}^{\mathcal{T}}_{pred}(\cdot)$, student model $\mathcal{SE}^{\mathcal{S}}(\cdot)$, $\mathcal{CE}^{\mathcal{S}}(\cdot)$, $\mathcal{CD}^{\mathcal{S}}(\cdot)$, $\mathcal{SD}^{\mathcal{S}}(\cdot)$ and $\mathcal{P}^{\mathcal{S}}_{pred}(\cdot)$ with parameters $\Theta^{\mathcal{S}} = \{\boldsymbol{\alpha}^{\mathcal{S}}, \boldsymbol{\beta}^{\mathcal{S}}, \boldsymbol{\gamma}^{\mathcal{S}}, \boldsymbol{\delta}^{\mathcal{S}}, \boldsymbol{w}^{\mathcal{S}}_{pred}, \boldsymbol{b}^{\mathcal{S}}_{pred}\}$.

1: Initialize parameters $\Theta^{\mathcal{S}}$, load pretrained teacher $\mathcal{SE}^{\mathcal{T}}(\cdot)$, $\mathcal{CE}^{\mathcal{T}}(\cdot)$, $\mathcal{CD}^{\mathcal{T}}(\cdot)$, $\mathcal{SD}^{\mathcal{T}}(\cdot)$ and $\mathcal{P}^{\mathcal{T}}_{pred}(\cdot)$
2: **for** $e = 1$ to E **do**
3: Compute the outputs of pretrained teacher $\boldsymbol{p}^{\mathcal{T}}$, $\boldsymbol{x}^{\mathcal{T}}$, $\hat{\boldsymbol{p}}^{\mathcal{T}}$, $\hat{\boldsymbol{t}}^{\mathcal{T}}$ and $\hat{\boldsymbol{s}}^{\mathcal{T}}$.
4: Perform forward propagation to compute the student outputs $\boldsymbol{p}^{\mathcal{S}}$, $\boldsymbol{x}^{\mathcal{S}}$, $\hat{\boldsymbol{p}}^{\mathcal{S}}$, $\hat{\boldsymbol{t}}^{\mathcal{S}}$ and $\hat{\boldsymbol{s}}^{\mathcal{S}}$.
5: Compute cost $J(\Theta^{\mathcal{S}})$ using loss function $\mathcal{L}_{overall}$ in (28).
6: Perform backward propagation to compute gradients $\frac{\partial J}{\partial \Theta^{\mathcal{S}}}$.
7: Update parameters $\Theta^{\mathcal{S}}$ using stochastic gradient descent.
8: **end for**

Output: Trained $\mathcal{SE}^{\mathcal{S}}(\cdot)$, $\mathcal{CE}^{\mathcal{S}}(\cdot)$, $\mathcal{CD}^{\mathcal{S}}(\cdot)$, $\mathcal{SD}^{\mathcal{S}}(\cdot)$ and $\mathcal{P}^{\mathcal{S}}_{pred}(\cdot)$ with parameters $\Theta^{\mathcal{S}}$.

student model in the second phase, which is demonstrated in Algorithm 3. We use the transfer training dataset $\boldsymbol{S}^{\mathcal{S}}$, which contains limited samples to simulate the scenario when there are not enough training data. Unlike the training dataset for the teacher model, we do not add MU interference in this training data and simulate a limited range of SNRs. Considering these constraints, we apply the feed forward propagation in Algorithm 1 to obtain the outputs of the students and the teacher from semantic encoder, channel encoder, channel decoder, semantic decoder, and prediction layer, respectively. Then, we apply the KD algorithm by computing the overall loss $\mathcal{L}_{overall}$, which considers the distilled loss $\mathcal{L}_{total\_distill}$ between the student and the teacher and the cross-entropy loss between the predicted sentence and the ground truth. Then,

Algorithm 4 Combined training algorithm for the proposed KD-based SemCom

Input: Training dataset $\boldsymbol{S}^{\mathcal{T}}$, transfer dataset $\boldsymbol{S}^{\mathcal{S}}$.

1: Train the teacher model $\leftarrow \boldsymbol{S}^{\mathcal{T}}$.
2: Train the student model $\leftarrow \boldsymbol{S}^{\mathcal{S}}$, pretrained teacher model $\mathcal{SE}^{\mathcal{T}}(\cdot)$, $\mathcal{CE}^{\mathcal{T}}(\cdot)$, $\mathcal{CD}^{\mathcal{T}}(\cdot)$, $\mathcal{SD}^{\mathcal{T}}(\cdot)$ and $\mathcal{P}^{\mathcal{T}}_{pred}(\cdot)$.
3: Quantize the student model by (29).

Output: : Trained and quantized $\mathcal{SE}^{\mathcal{S}}(\cdot)$, $\mathcal{CE}^{\mathcal{S}}(\cdot)$, $\mathcal{CD}^{\mathcal{S}}(\cdot)$, $\mathcal{SD}^{\mathcal{S}}(\cdot)$ and $\mathcal{P}^{\mathcal{S}}_{pred}(\cdot)$.

the student model's parameters are optimized by stochastic gradient descent after backpropagation. In the third phase, we apply post-training quantization to the weights and activation function so that the model size can be further compressed.

The overall loss function using cross-entropy and KL divergence could have multiple local minima due to their asymmetric inputs. Besides, the semantic encoder-decoder and channel encoder-decoder consists of millions of parameters with several hidden layers and non-linear activation function, which also causes multiple local minima. Therefore, the optimization problem using KD is considered non-convex. To alleviate the non-convexity and approximate the global minimum, we adopt Adam optimizer [32], layer normalization and dropout layer in the network to avoid local minima and improve convergence.

IV. Numerical results and discussion

In this section, we examine the performance of the proposed KD-based SemCom system in Rayleigh fading channels with MU interference. We assume perfect CSI for the desired user and various SNR regime is simulated in the experiment.

A. Simulation setting

There are several types of knowledge from the teacher model, and increasing distilled knowledge for the student model can increase the difficulty of the training by introducing more hyperparameters. Therefore, we propose four types of student models to evaluate the performance for different distilled knowledge and model compression. These four models will be trained using the limited dataset and tested in the SemCom system in the presence of co-channel interference to analyze the generalization ability on unseen data, which could answer Questions 1) and 2).

The parameter settings for the teacher and the student models are shown in Table I. We adopt the same structure as the L-DeepSC [8] for Teacher and Student 1, which has 4 layers of Transformer with 8 attention heads and 128 units as the semantic encoder and decoder. Additionally, 2 dense layers with 256 and 16 units are used for the channel encoder, and 3 dense layers with 128, 512, and 128 units are used for the channel decoder. Student 2 adopts 2 Transformer layers for the semantic encoder and decoder, while Student 3 and 4 further reduce the model size by using 1 dense layer for channel encoder and 2 dense layers for channel decoder. Finally, a dense layer is adopted as the prediction layer to

TABLE I: The setting of the teacher and student models for the SemCom system

	Teacher	Student 1	Student 2	Student 3	Student 4
Semantic encoder	4 × Transformer layers 8 heads 128 units		2× Transformer layers 8 heads 128 units		
Channel encoder	2 × Dense layers 256, 16 units			1 × Dense layers 16 units	
Channel decoder	3 × Dense layers 128, 512, 128 units			2 × Dense layers 128, 128 units	
Semantic decoder	4 × Transformer layers 8 heads 128 units		2× Transformer layers 8 heads 128 units		
Prediction layer	1 × Dense layers dictionary units				
Distilled knowledge	-	$(\boldsymbol{x}^{\mathcal{S}}, \boldsymbol{x}^{\mathcal{T}})$,$(\hat{\boldsymbol{t}}^{\mathcal{S}}, \hat{\boldsymbol{t}}^{\mathcal{T}})$ and $(\hat{\boldsymbol{s}}^{\mathcal{S}}, \hat{\boldsymbol{s}}^{\mathcal{T}})$		$(\boldsymbol{p}^{\mathcal{S}}, \boldsymbol{p}^{\mathcal{T}})$, $(\boldsymbol{x}^{\mathcal{S}}, \boldsymbol{x}^{\mathcal{T}})$, $(\hat{\boldsymbol{p}}^{\mathcal{S}}, \hat{\boldsymbol{p}}^{\mathcal{T}})$, $(\hat{\boldsymbol{t}}^{\mathcal{S}}, \hat{\boldsymbol{t}}^{\mathcal{T}})$ and $(\hat{\boldsymbol{s}}^{\mathcal{S}}, \hat{\boldsymbol{s}}^{\mathcal{T}})$	
Quantization	-	-	-	-	✓

output a vector with a dictionary size to represent the predicted word. All of the student models conduct KD from the teacher model. Students 1 and 2 consider the knowledge from the outputs of channel encoder, semantic encoder, and prediction layer, while Students 3 and 4 additionally consider the outputs from the semantic encoder and channel decoder. Moreover, post-training dynamic quantization is applied to Student 4. All learning-based SemCom systems require 8 symbols to represent one word since the output units of the channel encoder are set as 16, and it is converted to a two-dimensional vector as complex for transmission.

The performance of all baseline models is evaluated using the same number of transmitted symbols to guarantee a fair comparison. The settings of the baselines are described below:

1) Baselines of learning-based SemCom systems: All baselines are trained without KD to benchmark the contribution of distilled knowledge.

- DeepSC [7]: We apply the same structure of DeepSC, which has 3 layers of Transformer with 8 attention heads as semantic encoder and decoder and 2 dense layers as channel encoder and decoder.
- Baseline 1: We adopt the same structure as L-DeepSC, Teacher and Student 1.
- Baseline 2: We adopt the same model structure as Student 2.
- Baseline 3: We adopt the same model structure as Student 3.

2) Conventional communications systems: We adopt 8 symbols to represent one word for different source coding methods by choosing appropriate code rate for low-density parity-check code (LDPC) and 16-QAM for modulation, which are the same as the learning-based SemCom system.

- Huffman coding and LDPC: Huffman coding requires about 20 bits to represent a word, and we adopt 16-QAM for modulation and 5/8 as the code rate for LDPC.
- 5-Bit coding and LDPC: We adopt 16-QAM and 7/8 for the code rate of LDPC, since 5-bit uses about 28 bits to represent a word. Also, 5-bit coding is used to benchmark the performance without compression.

The source of the training dataset is the English dataset from the Europarl [33], which contains over 2 million sentences and 53 million words. The sentences are randomly split into the training set and the testing set. As the teacher model is pre-trained, we conduct different experiments to randomly add extra interference and noise to formulate the training datasets for the teacher model to make it robust. To simulate the interference, we use randomly selected sources. Moreover, the random occurrences and delays of interference are simulated by applying a 90% of occurrence rate and a maximum three-word (24 symbols) delay to the interfering signals. Different from the extensive training datasets for the teacher, the transferring dataset for the student model and the baselines is formulated with an interference-free channel and a limited regime of noise, which does not contain any interference samples. In this case, the student models and the baselines are trained with limited transferring datasets compared with the teacher model. Therefore, we can test the student and baseline models with unseen interference to evaluate the robustness and generalization ability under limited training data. The difference between the student models and the baselines is that the student models are conducted KD, whereas the baselines are not.

To measure the performance of the models mentioned above, we adopt a bilingual evaluation understudy (BLEU) score to measure the difference between two sentences [34]. However, it is difficult for BLEU to distinguish synonyms or polysemy. Thus, we also use sentence similarity [7], which adopts BERT [30] to map the sentences into semantic vector space and compare their semantic vectors. The simulation is performed on a computer with Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz and NVIDIA GeForce GTX 1080 Ti.

B. Performance without interference

In this section, we compare the performance without MU interference. The teacher model is trained using a dataset with SNR randomly changing from 10 dB to 15 dB, while others are trained by the transfer training dataset with a limited SNR ranging from 15 dB to 18 dB. Fig. 3 evaluates the BLEU performances for the teacher, the students, and the baselines. The BLEU accuracy of the teacher model ranges from about 52% to over 90% when the SNR increases from 0 dB to 18 dB. It outperforms other models because using the sufficient

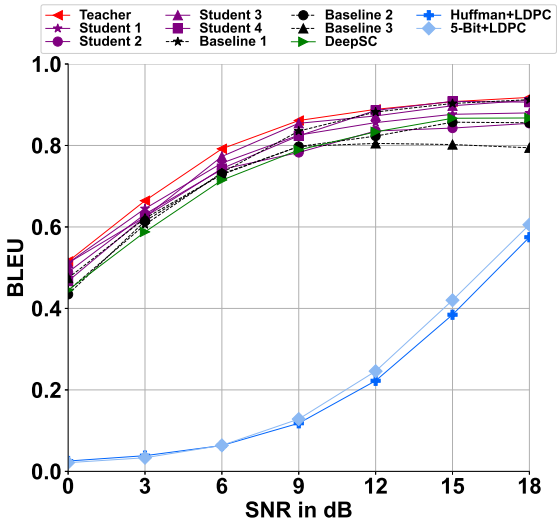


Fig. 3: BLEU score of the SemCom with no interference.

training set with a wider SNR range improves its robustness over different SNRs. With the transferred knowledge from the teacher model, all student models can perform better than the baselines without KD when the SNR is less than 9 dB. The reason is that the knowledge for the generalization in the low SNR regime is transferred from the teacher to the students, while the baselines can only learn through the hard label information. This also illustrates that, when the transfer dataset is limited, KD from the teacher can improve the robustness of the model.

For conventional communications systems, the BLEU can reach about 60% when SNR is 18 dB. The non-compression coding scheme, 5-Bit with LDPC, performs slightly better than Huffman and LDPC when SNR increases over 9 dB. However, there is still a significant performance gap compared with learning-based SemCom systems. Additionally, a slight performance decline for Baseline 3 can be observed when SNR increases over 12 dB due to overly simplifying the model without the distilled knowledge from the teacher. One of the advantages of KD is that the student model does not need to have access to a large amount of training data while the teacher can be trained offline anywhere. This allows for data isolation and privacy. Also, compared with conducting several experiments to train the student model for different values of SNRs and SIRs, directly distilling the knowledge from the teacher could save the experiment time.

C. Performance with MU interference

To further investigate the effect of the proposed KD-based SemCom system, we evaluate the BLEU, sentence similarity and BER with one co-channel interference when SIR increases from 0 to 18 dB in Fig. 4. This is equivalent to increasing the number of interfering users while keeping the SIR for each user. The BER of the SemCom system is computed by converting the recovered words into bits using ASCII. To evaluate the KD-based models with unseen data, the students and the baselines are trained with the transferring dataset without no MU interference samples, while they are tested in the presence of MU interference to show robustness.

For the BLEU score, although the performance gap between the baselines and the student models narrows when SIR grows, the student models with distilled knowledge perform better than the baselines when SIR is less than 12 dB. For the conventional communications system, 5-bit and LDPC range from 30% to over 50% when SIR increases to 18 dB, which performs better than Huffman and LDPC. Nevertheless, the students with KD still greatly outperform the conventional communications systems regardless of the SIRs. The performance of sentence similarity in Fig. 4 demonstrates the same trend as the BLEU scores. The sentence similarity of students ranges from about 60% to over 80% when SIR increases to 18 dB, which is better than the baselines and conventional communications systems. This shows that the models with KD have better word accuracy in BLEU scores and recover the sentences that are easier to understand. Moreover, the learning-based communications systems outperform the conventional communications system for all SIRs, indicating that people can better understand the text with the SemCom system. However, the conventional baselines do perform better than all the SemCom systems in terms of BER. This is attributed to the transformer-based encoder-decoder used by SemCom, which recovers the intended message through contextual reasoning on a word-by-word basis, instead of adhering strictly to the precise wording. SemCom sacrifices bit errors for the recovery of the meaning of words to save spectrum. In this regard, the semantic systems may have limited applications in services requiring precise bits, such as voice control. Despite this, the students with KD still outperform the learning-based baselines in BER, demonstrating the enhanced generalization to unseen interference by KD.

Fig. 5 demonstrates the performances in the presence of one interfering user when SIR is 0 dB and 10 dB to simulate the strong and weak interference. When the physical channel is severely interfered, the BLEU accuracy of the baselines and DeepSC can barely reach 20% because there is neither KD in the training nor interference samples in the training data. However, it still outperforms the conventional communications system with Huffman and 5-bit coding schemes when the SNR is less than 12 dB. The overall BLEU score of the teacher ranges from about 30% to 50% as the SNR increases from 0 to 18 dB due to its powerful generalization ability. On the other hand, the BLEU accuracy of the student models with distilled knowledge ranges from 25% to 30% when SNR increases from 0 to 18 dB, which outperforms the baselines and DeepSC by over 10%. This shows that distilled knowledge can improve the model performance of generalizing on unseen data. Moreover, Student 2 and Student 1 have better performances than other students because overly compressing the size of the model can result in performance degradation.

When the SIR is 10 dB, the BLEU score of the teacher ranges from about 47% to over 80% when the SNR is from 0 to 18 dB. Again, the student models perform better than the baselines and DeepSC from 0 to 18 dB. Also, Student 2 and Student 1 perform slightly better than Student 3 and Student 4. Student 2 only compresses the semantic encoder

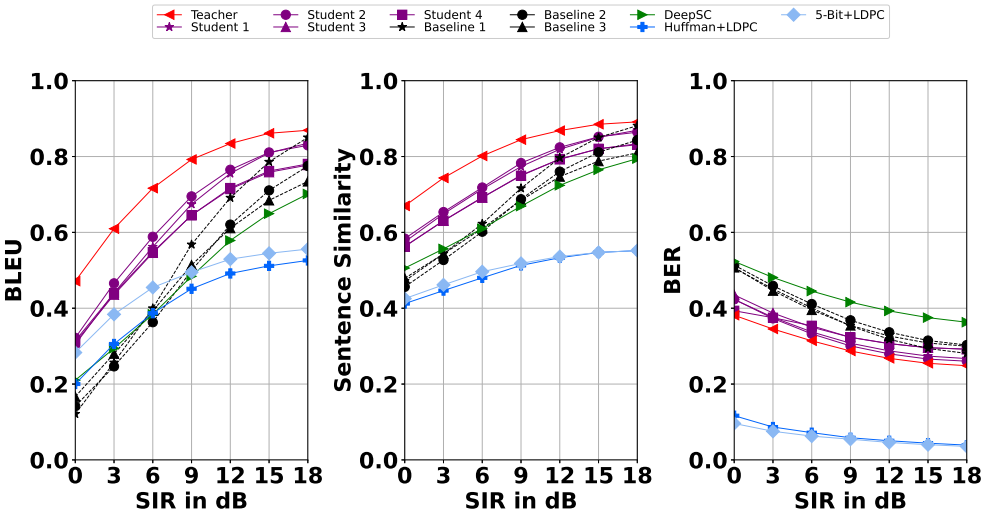


Fig. 4: BLEU score and sentence similarity of the SemCom with one MU interference when the SNR is 18 dB.

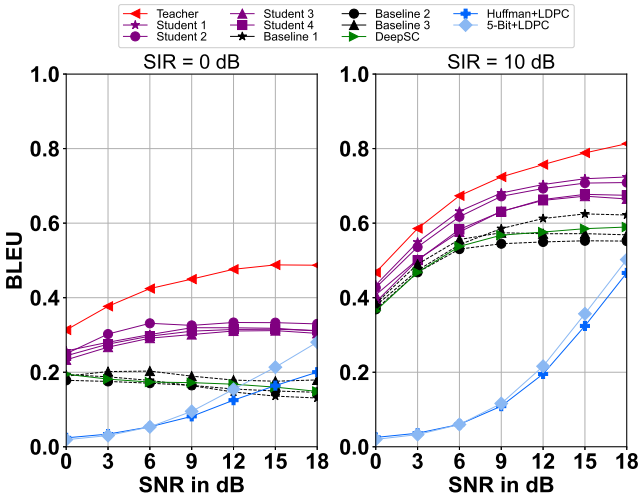


Fig. 5: BLEU score of the SemCom with one MU interference.

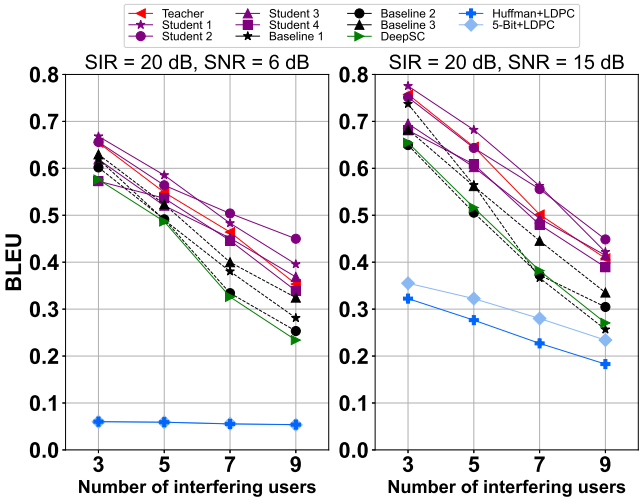


Fig. 6: BLEU score of the SemCom with multiple interference.

and decoder, while Student 3 and Student 4 conduct model compression on both semantic encoder-decoder and channel encoder-decoder. Additionally, Student 3 and Student 4 have similar performances, which illustrates that the post-training dynamic quantization can barely have any effect on the BLEU score performance.

To test the performance when there are more interfering users, Fig. 6 demonstrates the BLEU performance with multiple interference when SIR is 20 dB. When SNR is 6 dB, Student 1 and Student 2 outperform the teacher, as the number of interference increases from 3 to 9. This demonstrates that distilled knowledge can provide the generalization ability for Students 1 and 2, and direct the student models in optimization to outperform their teacher. Moreover, this compresses the model and improves the performance simultaneously. Additionally, all student models can generally outperform the baselines and DeepSC, except that Student 4 has a relatively poor accuracy when the number of interference is 3. For conventional communications systems, the BLEU of the Huffman and 5-bit coding with LDPC remains less than 10%. When the SNR is 15 dB, the BLEU score of the teacher ranges from about 77% to 41% as the number of interfering users increases from 3 to 9. The 5-Bit and LDPC scheme performs slightly better than Huffman and LDPC with an accuracy ranging from 35% to 25%. However, Student 1 and Student 2 still demonstrate better performance than the teacher, the baselines, and the conventional communications. Comparing Student 1 and Baseline 1, KD without model compression improves the accuracy by over 20%, as the number of interfering users increases. Furthermore, Student 1 has a slightly better performance than Student 2, which shows that KD can reduce the performance loss while compressing the model.

D. Ablation experiment

In Table II, we investigate the effect of various components of the loss function on the distilled student models. We

TABLE II: Variations on BLEU score relative to the model trained with all losses.

	Without interference		With interference	
	SNR=6	SNR=15	SNR=6	SNR=15
Without $\mathcal{L}_{distill}(\hat{\boldsymbol{s}}^{\mathcal{T}}, \hat{\boldsymbol{s}}^{\mathcal{S}})$	-5.16%	-2.25%	-5.11%	-1.07%
Without $\mathcal{L}_{distill}(\boldsymbol{x}^{\mathcal{T}}, \boldsymbol{x}^{\mathcal{S}})$ and $\mathcal{L}_{distill}(\hat{\boldsymbol{t}}^{\mathcal{T}}, \hat{\boldsymbol{t}}^{\mathcal{S}})$	-6.08%	-4.62%	-5.73%	-1.45%
Without $\mathcal{L}_{distill}(\boldsymbol{p}^{\mathcal{T}}, \boldsymbol{p}^{\mathcal{S}})$ and $\mathcal{L}_{distill}(\hat{\boldsymbol{p}}^{\mathcal{T}}, \hat{\boldsymbol{p}}^{\mathcal{S}})$	-5.08%	-6.35%	-0.89%	-1.18%
Without $\mathcal{L}_{hard}(\boldsymbol{s}^{\mathcal{S}}, \hat{\boldsymbol{s}}^{\mathcal{S}})$	-18.63%	-12.59%	-11.71%	-6.44%

categorize the distilled losses by considering the symmetric structure of the SemCom system. The SemCom systems with no interference and with one interference when SIR is 10 dB are considered.

When there is no interference, the distilled knowledge from the outputs of the channel encoder and the semantic decoder has more impact on the performance than other distilled knowledge when there is strong noise in the system. Conversely, the knowledge from the intermediate outputs of the encoded semantic information $\boldsymbol{p}^{\mathcal{T}}$ and $\boldsymbol{p}^{\mathcal{T}}$ tends to have more influence on the performance when there is less noise. The cross-entropy loss $\mathcal{L}_{hard}(\boldsymbol{s}^{\mathcal{S}}, \hat{\boldsymbol{s}}^{\mathcal{S}})$ is the most important loss function regardless of the presence of the interference because it has the hard label information, which is the direct way to improve the performance. When there is unseen interference, the effect of the hard labels information and the intermediate outputs of the semantic encoded information diminishes, while $\mathcal{L}_{distill}(\hat{\boldsymbol{s}}^{\mathcal{T}}, \hat{\boldsymbol{s}}^{\mathcal{S}})$, $\mathcal{L}_{distill}(\boldsymbol{x}^{\mathcal{T}}, \boldsymbol{x}^{\mathcal{S}})$ and $\mathcal{L}_{distill}(\hat{\boldsymbol{t}}^{\mathcal{T}}, \hat{\boldsymbol{t}}^{\mathcal{S}})$ can have similar effects on the BLEU score to the condition with no interference. The student model improves the robustness of generalizing on the unseen data by learning from the teacher model through these distilled knowledge. Moreover, these results can be used as the guidance to determine the proportion of different loss functions.

E. Complexity Analysis

TABLE III: The complexity analysis with number of parameters, model size, training time and inference time.

	Parameters	Size (MB)	Training time (ms/batch)	Inference time (ms/sentence)
Teacher	2022672	12.46	108.86	24.10
Student 1	2022672	12.46	199.75	22.90
Student 2	1096976	6.98	169.87	14.87
Student 3	946704	6.06	166.86	14.50
Student 4	5376	0.05	166.86	14.18
DeepSC	1462928	9.18	95.76	19.79
LDPC	-	-	-	42.74

In Table III, we conduct the complexity analysis for the student models in terms of the number of parameters, the size of the models, training time per batch and average inference time per sentence. With the model compression for semantic encoder-decoder components, the size of the semantic encoder and decoder is reduced from 12.46 MB to 6.98 MB. Moreover, about 50% of the parameters are reduced for Student 2 compared with Student 1. Student 3 is compressed on the channel encoder and decoder, which has 0.92 MB, slimmer than Student 2. Furthermore, Student 4 uses post-training dynamic quantization based on Student 3, which further reduces the size from 6.06 MB to 0.05 MB.

Furthermore, Student 1 costs about 199 ms/batch for training, which is the longest since it has the same size as the teacher but considers the extra distilled knowledge from the teacher. Students 2, 3, and 4 benefit from a reduced model size, resulting in approximately 30 ms/batch reduction in training time than Student 1. Despite this, they still require longer training time than the non-distilled models. This shows a potential drawback of KD for introducing extra computational overhead during model training. In terms of the average inference time, it is related to the size of the model for the learning-based models. Students 2, 3, and 4 can have an inference time of less than 15 ms/sentence, whereas Teacher and Student 1 take more than 20 ms/sentence. This significantly outperforms the traditional communications system using LDPC. Therefore, KD-assisted model compression could reduce the sentence processing time during inference, improving the real-time latency. However, this comes at the expense of increasing training costs, which may not be suitable for applications, such as online learning, where the model is continuously trained with new incoming data.

V. Conclusion

In this paper, we have proposed a KD-based SemCom system with MU interference. Specifically, four distilled student models have been designed and trained with the constraints of limited training samples. Performances have been compared and analyzed for different SNRs, SIRs, and the number of interference. Numerical results have shown distilled models perform better than the non-distilled baselines and the conventional communications system with Huffman codes and LDPC as the source and channel coding scheme when generalizing on unseen interference. KD can greatly improve the generalization and robustness of the student models. Moreover, the complexity analysis has been conducted to illustrate that KD can reduce inference time by compressing the model while compromising on training cost. Furthermore, an ablation study has compared the importance of various distilled loss functions on the distilled student models. Finally, simulation results have also shown that the post-training dynamic quantization has a very limited effect on the system performance.

References

[1] C. E. Shannon and W. Weaver, "The mathematical theory of communication," *The University of Illinois Press*, 1949.

[2] W. Yang, H. Du, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Shen, and C. Miao, "Semantic communications for future internet: Fundamentals, applications, and challenges," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 1, pp. 213–250, 2023.

[3] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is semantic communication? a view on conveying meaning in the era of machine intelligence," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, 2021.

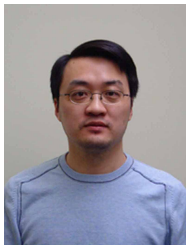
[4] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2022.

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

[6] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning based semantic communications: An initial investigation," in *IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.
[7] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.
[8] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, 2021.
[9] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, 2021.
[10] E. Bourtsoulatze, D. Burth 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, 2019.
[11] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, 2020.
[12] S. Wang, J. Dai, S. Yao, K. Niu, and P. Zhang, "A novel deep learning architecture for wireless image transmission," in *IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.
[13] D. Huang, X. Tao, F. Gao, and J. Lu, "Deep learning-based image semantic coding for semantic communications," in *IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.
[14] S. Catreux, P. F. Driessen, and L. J. Greenstein, "Attainable throughput of an interference-limited multiple-input multiple-output (mimo) cellular system," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1307–1311, 2001.
[15] A. Omri and M. O. Hasna, "Novel cooperative communication schemes with interference management for multi-user wireless networks," in *IEEE Int. Conf. Commun.*, 2013, pp. 5651–5656.
[16] S. Huang, J. Cai, H. Chen, and F. Zhao, "Low-complexity priority-aware interference-avoidance scheduling for multi-user coexisting wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 1, pp. 112–126, 2018.
[17] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2184–2199, Oct 2019.
[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531
[19] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.
[20] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Nov. 2020, pp. 4163–4174.
[21] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.
[22] Z. Qin, H. Ye, G. Y. Li, and B. F. Juang, "Deep learning in physical layer communications," *IEEE Wirel. Commun.*, vol. 26, no. 2, pp. 93–99, April 2019.
[23] E. Calvanese Strinati and S. Barbarossa, "6g networks: Beyond shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107930, 2021.
[24] N. Samuel, T. Diskin, and A. Wiesel, "Deep mimo detection," in *IEEE Workshop Signal Process. Adv. Wirel. Commun.*, 2017, pp. 1–5.
[25] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 1, pp. 114–117, 2018.
[26] 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, 2018.
[27] C. Liu, Y. Chen, and S.-H. Yang, "Signal detection with co-channel interference using deep learning," *Phys. Commun.*, vol. 47, p. 101343, 2021.
[28] C. Liu, Y. Chen, and S.-H. Yang, "Deep learning based detection for communications systems with radar interference," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6245–6254, 2022.
[29] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented multi-user semantic communications for vqa," *IEEE Wirel. Commun. Lett.*, vol. 11, no. 3, pp. 553–557, 2022.
[30] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186.
[31] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
[32] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, [Online]. Available: https://arxiv.org/abs/1412.6980.
[33] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. of Machine Translation Summit X: Papers*, 2005, pp. 79–86.
[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

Chenguang Liu Chenguang Liu received his B.E. degree in software engineering from Dalian University of Technology, Dalian, P.R.China, in 2016 and M.S. degree in advanced computer science from The University of Manchester, U.K., in 2017. He is currently studying as a Ph.D. student at the University of Warwick, U.K. His research interests include deep learning, wireless communications and cooperative perception.

Yuxin Zhou Yuxin Zhou received his B.E. degree in robotics engineering from Harbin Institute of Technology, Weihai, P.R.China, in 2020. He is currently studying as a M.E. student at the Southern University of Technology and Science, Shenzhen, P.R.China. His research interests include deep learning and computer vision.



Yunfei Chen Yunfei Chen (S'02-M'06-SM'10) received his B.E.and M.E. degrees in electronics engineering from Shanghai Jiaotong University, Shanghai, P.R.China,in 1998 and 2001, respectively. He received his Ph.D. degree from the University of Alberta in 2006. He is currently working as a Professor at the University of Durham, U.K. His research interests include wireless communications, cognitive radios, wireless relaying and energy harvesting.

Shuang-Hua Yang Shuang-Hua Yang (Senior Member, IEEE) received the B.S. degree in instrument and automation and the M.S. degree in process control from the China University of Petroleum, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in intelligent systems from Zhejiang University, Hangzhou, China, in 1991. He is currently the director of the Shenzhen Key Laboratory of Safety and Security for Next Generation of Industrial Internet, Southern University of Science and Technology, and a Professor and the Head of the Department of Computer Science, University of Reading, U.K. His research interests include Internet of Things, industrial internet, Cyber-Physical System safety and security, Process Systems Engineering. Prof. Yang is a fellow of IET and InstMC, U.K. He is an Associate Editor of IET Cyber-Physical Systems: Theory and Applications.