

Semantic Communication With Memory

Huiqiang Xie¹, Graduate Student Member, IEEE, Zhijin Qin², Senior Member, IEEE,
and Geoffrey Ye Li³, Fellow, IEEE

Abstract—While semantic communication succeeds in efficiently transmitting due to the strong capability to extract the essential semantic information, it is still far from the intelligent or human-like communications. In this paper, we introduce an essential component, memory, into semantic communications to mimic human communications. Particularly, we investigate a deep learning (DL) based semantic communication system with memory, named Mem-DeepSC, by considering the scenario question answer task. We exploit the universal Transformer based transceiver to extract the semantic information and introduce the memory module to process the context information. Moreover, we derive the relationship between the length of semantic signal and the channel noise to validate the possibility of dynamic transmission. Specially, we propose two dynamic transmission methods to enhance the transmission reliability as well as to reduce the communication overheads by masking some unessential elements, which are recognized through training the model with mutual information. Numerical results show that the proposed Mem-DeepSC is superior to benchmarks in terms of answer accuracy and transmission efficiency, i.e., number of transmitted symbols.

Index Terms—Semantic communications, memory task, dynamic transmission, deep learning.

I. INTRODUCTION

THE seamlessly connected world fosters unique services, like virtual reality (VR), mobile immersive eXtended reality (XR), or autonomous driving, and brings new challenges to communication systems, such as the scarcity of resources, the congestion of network traffic, and the scalable connectivity for edge intelligence [1]. To materialize the vision, semantic communication [2] is a communication paradigm by directly delivering the meanings of information, and extracting and transmitting only important information relevant to the task at the receiver. In the past couple of years, semantic communication is attracting extensive attention from both academia [2] and industry [3], [4]. The latest works take advantage of deep learning (DL) to design end-to-end

semantic communication systems for various types of source reconstruction and specific tasks execution, which has shown a great potential to increase the reliability in performing intelligent tasks, reducing the network traffic, and thus alleviating spectrum shortage.

The existing works in semantic communication can be categorized by the types of source data. For image-based semantic communication systems, Bourtsoulatzé et al. [5] have investigated the deep image transmission semantic communication systems, in which the semantic and channel coding are optimized jointly. Kurka and Gunduz [6] extended Bourtsoulatzé's work with the channel feedback to improve the quality of image reconstruction. Jankowski et al. [7] have developed digital and analog deep joint source-channel coding (JSCC) to perform the person/car re-identification task directly, which improves the image retrieval accuracy effectively. Lee et al. [8] have considered the image classification as the communication task, where JSCC is based on the convolutional neural network (CNN). Moreover, Hu et al. [9] have designed a robust semantic communication system against semantic noise by employing adversarial training, which reduces the probability of misleading in classification. In order to reduce the communication overheads, Yang et al. [10] have developed bandwidth-limited semantic communication system by removing the redundancy of semantic features while keeping similar classification accuracy. Shao et al. [11] have proposed a dynamic semantic communication system to adaptively adjust the number of the active semantic features under different signal-to-noise ratios (SNRs) with a graceful classification accuracy degradation. Huang et al. [12] 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.

Apart from image based semantic communication systems, the video based semantic communication systems also attracted much attention. Tung and Gunduz [13] 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. [14] 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. [15] 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. [16]

Manuscript received 1 September 2022; revised 29 March 2023; accepted 3 May 2023. Date of publication 21 June 2023; date of current version 7 August 2023. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62293484. (Corresponding author: Zhijin Qin.)

Huiqiang Xie is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: h.xie@qmul.ac.uk).

Zhijin Qin is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: qinzhijin@tsinghua.edu.cn).

Geoffrey Ye Li is with the School of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: geoffrey.li@imperial.ac.uk).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSAC.2023.3288236>.

Digital Object Identifier 10.1109/JSAC.2023.3288236

also considered the video conference transmission. Different from [15], 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.

Meanwhile, there exist works on the speech-based, text-based, and multimodal semantic communication systems. Weng et al. [17] have developed the speech recognition-oriented semantic communication, named DeepSC-SR, in which the transmitter sends the speech signal and the receiver restores the text directly. Han et al. [18] have designed an more energy-efficient speech-to-text system by introducing the redundancy removal module to lower the transmitted data size. With the depth exploration in semantic communications, Xie and Qin [19] have developed a powerful joint semantic-channel coding, named DeepSC, to encode text information into various lengths over complex channels. Moreover, Xie et al. [20] also have proposed an environment-friendly semantic communication system, named L-DeepSC, for capacity-limited devices. Zhou et al. [21] have proposed a Universal Transformer based semantic communication system for text transmission with lower complexity. Peng et al. [22] have designed robust semantic communication systems to prevent the semantic delivery from the source noise, e.g., typos and syntax errors. By considering the multi-user scenario for multi-modal data transmission, Xie et al. [23] have proposed a unified Transformer based semantic communication framework to support the image and text transmission and to enable the receiver to perform various multimodal tasks. These existing works have revealed the potential of semantic communication in various intelligent tasks and moved towards the ideal intelligent communication. However, the proposed semantic communication systems only consider the current time slot inputs and ignore the previous time slots inputs. For the next step of task-oriented semantic communication, we can find some inspiration from human communication.

In the human communication [24], people can perform the both memoryless tasks and memory tasks. Memoryless tasks are only relevant to the inputs received in the current time slot, e.g., receiving the image and recognizing its category. Memory tasks [25] are relevant to inputs received in both the current and past time slots, e.g., the response in the conversation relying not only on the currently listened sentences but also on the previous context. While the developed semantic communication systems only consider the inputs in the current time slot and neglected those in the previous time slots. With such a design, semantic communication is incapable of serving the memory tasks, such as scenario question answer, scenario visual question task, and scenario conversations.

Note that one of the key modules in human communications is *memory*, which can store the context semantic information and enable people to perform tasks requiring memory. We are inspired to introduce the memory module in semantic communications so as to execute tasks with and without memory. By dosing so, machine to machine commendations and human to machine communications will become more intelligent and human-like, which could fully exploit the advancements of semantic communications. It is

of great interest to design a semantic communication system that utilizes memory information to facilitate the semantic information transmission and task execution at the receiver. To design a semantic communication system with memory, we are facing 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?

In this paper, we investigate a semantic communication for memory tasks by taking the scenario question answer task as an example. Particularly, we develop a DL enabled semantic communication system with memory (Mem-DeepSC) to address the aforementioned challenges. The main contributions of this paper are summarized as follows:

- Based on the universal Transformer [26], 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 dynamic transmission environment, the relationship between the length of semantic signal and the 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*.

The rest of this paper is organized as follows. The system model is introduced in Section II. The semantic communication system with memory module is proposed in Section III. Section IV details the proposed dynamic transmission methods. Numerical results are presented in Section V to show the performance of the proposed frameworks. Finally, Section VI concludes this paper.

Notation: Bold-font variables denote matrices or vectors. $\mathbb{C}^{n \times m}$ and $\mathbb{R}^{n \times m}$ represent complex and real matrices of size $n \times m$, respectively. $x \sim \mathcal{CN}(\mu, \sigma^2)$ means variable x follows a circularly-symmetric complex Gaussian distribution with mean μ and covariance σ^2 . $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and Hermitian, respectively. \mathbf{I}_M is the $M \times M$ is the identity matrix, $\mathbf{1}_{M \times 1}$ is the all-one vector with length M , $\mathbf{1}_i$ is the one-hot vector with one in the i -th position. \odot and \oslash are the element-wise multiplication and division, respectively. $\mathbf{x}(k)$ represents the signal at the k -th time slot. $x[k]$ represents the k -th element in the vector. $\mathbf{X}[k]$ represents the k -th row in the matrix.

II. SYSTEM MODEL

As shown in Fig. 1, we consider a single-input single-output (SISO) communication system, which is with one antenna at the transmitter and one at the receiver. We focus on the text scenario question answer, therefore, the transmission includes two phases: i) memory shaping to transmit the context, e.g., multiple sentences, images, or speeches, to the receiver via

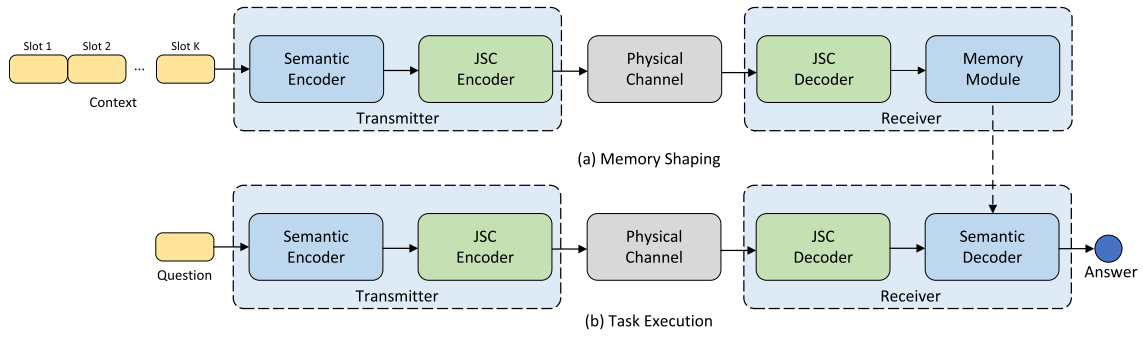


Fig. 1. The proposed framework for semantic communication systems with memory module.

multiple time slots; ii) task execution to transmit the question relevant to the context so as to obtain the answer at the receiver. In this paper, we consider multiple sentences as the context.

The transceiver has three modules, a semantic codec to extract the semantic features of the source and perform the task, a joint source-channel (JSC) codec to compress and recover the semantic features, and the memory module to store the received context in multiple time slots and aid semantic decoder in performing the task.

A. Memory Shaping

We assume the k -th context, $\mathbf{s}(k)$, is transmitted during the k -th time slot and denote \mathbf{s}^c and \mathbf{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. Subsequently, with the semantic encoder and channel encoder, the k -th context sentence over the k -th time slot can be encoded as

$$\mathbf{x}^c(k) = C(S(\mathbf{s}^c(k); \boldsymbol{\alpha}); \boldsymbol{\beta}), \quad (1)$$

where $\mathbf{x}^c(k) \in \mathbb{C}^{L \times 1}$ is the transmitted signals after the power normalization, $S(\cdot; \boldsymbol{\alpha})$ and $C(\cdot; \boldsymbol{\beta})$ 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

$$\mathbf{y}^c(k) = \mathbf{h}(k) \odot \mathbf{x}^c(k) + \mathbf{n}(k), \quad (2)$$

where $\mathbf{h}(k)$ is the channel coefficients and \mathbf{n} is the additive white Gaussian noise (AWGN), in which $\mathbf{n}(k) \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_L)$. For the Rayleigh fading channel, the channel coefficient follows $\mathbf{h}(k) \sim \mathcal{CN}(0, \mathbf{I}_L)$; for the Rician fading channel, it follows $\mathbf{h}(k) \sim \mathcal{CN}(\mu_h \mathbf{I}_{L \times 1}, \sigma_h^2 \mathbf{I}_L)$ 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 $\mathbb{E}(\|\mathbf{h}(k) \odot \mathbf{x}^c(k)\|^2) / \mathbb{E}(\|\mathbf{n}(k)\|^2)$.

With the estimated channel state information (CSI), $\hat{\mathbf{h}}(k)$, the transmitted signals, $\hat{\mathbf{x}}(k)$, can be detected by

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{h}}^H(k) \odot \mathbf{y}^c(k) \odot \left(\hat{\mathbf{h}}(k) \odot \hat{\mathbf{h}}^H(k) \right). \quad (3)$$

After signal detection, the semantic features can be recovered by

$$\hat{\mathbf{z}}^c(k) = C^{-1}(\hat{\mathbf{x}}^c(k); \boldsymbol{\gamma}), \quad (4)$$

where $\hat{\mathbf{z}}^c(k) \in \mathbb{R}^{N \times 1}$ and $C^{-1}(\cdot; \boldsymbol{\gamma})$ is denoted as the channel decoder with parameter $\boldsymbol{\gamma}$. Then, the recovered semantic features will be inputted into the memory module.

Inspired by the short-term memory, we model the memory module as a queue with length T , which only concerns the context. The memory queue at the k -th time slot is represented by

$$\mathbf{Q}(k) = [\hat{\mathbf{z}}^c(k-T+1); \hat{\mathbf{z}}^c(k-T+2); \dots; \hat{\mathbf{z}}^c(k)]. \quad (5)$$

where $\mathbf{Q}(k) \in \mathbb{R}^{N \times T}$, T is the length of memory. For $k < 0$, $\hat{\mathbf{z}}^c(k)$ is the vector with all zero elements. The memory queue is updated with the incoming received latest semantic features and pop the oldest features out of the queue. For example, the memory queue at the $(k+1)$ -th time slot is given by

$$\mathbf{Q}(k+1) = [\hat{\mathbf{z}}^c(k-T+2); \hat{\mathbf{z}}^c(k-T+3); \dots; \hat{\mathbf{z}}^c(k+1)]. \quad (6)$$

For scenario communications, it is important to recognize the sequence of the memory queue. In other words, we need to know the order of sentences happened earlier or later. Therefore, we need to add the temporal information before inputting to the models, which is given by

$$\mathbf{M}(k) = \mathbf{Q}(k) + \mathbf{T}, \quad (7)$$

where $\mathbf{T} = [\mathbf{t}_1; \mathbf{t}_2; \dots; \mathbf{t}_T] \in \mathbb{R}^{N \times T}$ is the temporal information matrix, any two elements of which satisfy $\langle \mathbf{t}_m, \mathbf{t}_n \rangle = g(m-n)$ for some function $g(\cdot)$. We choose the positional coding employed in Transformer [27] here.

When performing the memory task, the memory shaping phase will introduce the overheads in terms of time availability due to occupying multiple time slots. For example, considering the task offloading scenario, compared to the memoryless tasks that only use several time slots, the memory shaping will occupy multiple time slots and may congest the transmission pipe. This makes the time consumption a little high at the initial memory shaping phase. However, when the memory module is shaped, such consumption can be alleviated by reusing or partly updating the shaped memory module, because the memory module serves multiple questions. Therefore, the frequency of carrying memory shaping phase depends on the

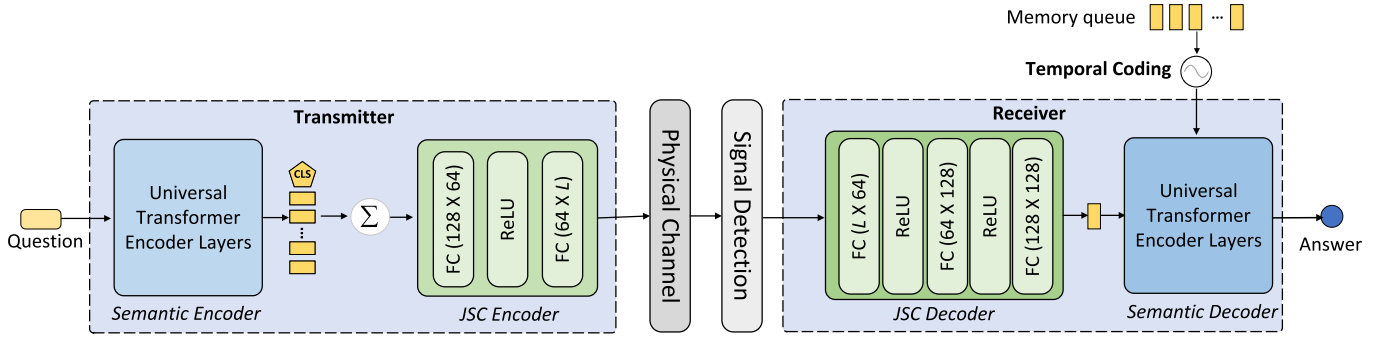


Fig. 2. The proposed network structure of Mem-DeepSC system.

demands of users. When performing the memory task, the user will decide whether the current context stored in the memory module can answer the question. If not, the memory needs to be updated.

B. Task Execution

In the task execution phase, the transmitter sends the question sentence, s^q , to the receiver to perform the task. Specially, s^q is encoded into x^q by (1), transmitted over the air, and decoded into \hat{z}^q by (4). In the scenario question answer task, the question is not only relevant 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

$$\hat{a} = S^{-1}([\hat{z}^q, \mathbf{M}(k)]; \varphi), \quad (8)$$

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

III. 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.

A. Model Description

The proposed Mem-DeepSC is shown in Fig. 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:

- The state-of-arts, i.e., BERT [28] and GPT-3 [29], choose Transformer to deal with textual information, which shows the powerful extraction capability of text semantic information.
- The universal Transformer can be trained and tested much faster than the architectures based on recurrent layers due to the parallel computation [26].
- Compared with the classic Transformer, the universal Transformer shares the parameters, which can reduce the model size.

After the semantic encoder, 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:

- The JSC codec aims to compress the semantic features and transmit it effectively over the air. 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. This can enhance the system's robustness to channel noise.

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, *temporal coding* is employed to add temporal information to the memory queue, in which we adopt the positional coding [27] as the temporal coding.

B. Training Details

As shown in Algorithm 1, the training of Mem-DeepSC includes three steps, which is similar to the training algorithm proposed in [23]. The first step is to train the semantic codec. In order to improve the accuracy of answers, we choose the cross-entropy (CE) as the loss function instead of the answer accuracy. The cross-entropy is given by

$$\mathcal{L}_{\text{CE}} = -p(a) \log(p(\hat{a})), \quad (9)$$

where $p(a)$ is the real probability of answer and $p(\hat{a})$ is the predicted probability. After convergence, the model learns to extract the semantic features and predict the answers. The following proposition proven 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.

After converged, the model is capable of extracting semantic features and predicting the answer accurately. Subsequently, 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 mean-squared error (MSE) loss function,

$$\mathcal{L}_{\text{MSE}} = \|\mathbf{z}^c(k) - \hat{\mathbf{z}}^c(k)\|^2, \quad (10)$$

Algorithm 1 Mem-DeepSC Training Algorithm

```

1 Function Train Semantic Codec():
   Input:  $\{(s^c(1), s^c(2), \dots, s^c(K)), s^q, a\}$  from
   dataset.
2 for  $k = 1 \rightarrow K$  do
3    $S(s^c(k); \alpha) \rightarrow \mathbf{W}^c(k)$ ,
4   Take the summation operation,
    $z^c(k) = \sum_j \mathbf{W}^c(k)[j]$ ,
5    $S(s^q; \alpha) \rightarrow \mathbf{W}^q$ , and  $z^q = \sum_j \mathbf{W}^q[j]$ ,
6   Shape the memory queue  $\mathbf{Q}(K)$  by (5),
7   Take the temporal coding for  $\mathbf{M}(K)$  by (7),
8    $S^{-1}([z^q, \mathbf{M}(K)]; \varphi) \rightarrow \hat{a}$ ,
9   Compute CE loss with  $a$  and  $\hat{a}$ .
10  Train  $\alpha, \varphi \rightarrow$  Gradient descent with CE loss.
   Return:  $S(\cdot; \alpha)$  and  $S^{-1}(\cdot; \varphi)$ .
11 Function Train JSC Codec():
   Input: Semantic features  $z^c(k)$ .
12 Transmitter:
13    $C(z^c(k); \beta) \rightarrow x^c(k)$ ,
14   Power Normalization,
15   Transmit  $x^c(k)$  over the air.
16 Receiver:
17   Receive  $y^c(k)$ ,
18   Signal detection by (3) to get  $\hat{x}^c(k)$ ,
19    $C^{-1}(\hat{x}^c(k); \gamma) \rightarrow \hat{z}^c(k)$ ,
20   Compute MSE loss with  $z^c(k)$  and  $\hat{z}^c(k)$ .
21   Train  $\beta, \gamma \rightarrow$  Gradient descent with MSE loss.
   Return:  $C(\cdot; \beta)$  and  $C^{-1}(\cdot; \gamma)$ .
22 Function Train Whole Network():
   Input:  $\{(s^c(1), s^c(2), \dots, s^c(K)), s^q, a\}$  from
   dataset.
23   Repeat line 2-5, 12-19, and 6-8 to get  $\hat{a}$ ,
24   Compute CE loss with  $\hat{a}$  and  $a$ .
25   Train  $\alpha, \beta, \gamma, \varphi \rightarrow$  Gradient descent with CE loss.
   Return:  $S(\cdot; \alpha)$ ,  $S^{-1}(\cdot; \varphi)$ ,  $C(\cdot; \beta)$ , and
    $C^{-1}(\cdot; \gamma)$ .

```

where $z^c(k)$ and $\hat{z}^c(k)$ 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 module.

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.

IV. ADAPTIVE RATE TRANSMISSION

In this section, we firstly derive the relationship between the length of semantic signal and channel noise, which inspires us

to transmit different length signals according to SNRs. We then 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.

A. The Relationship Between the Length of Semantic Signal and Channel Noise

Adaptive modulation has been developed for conventional communications [30], 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 over AWGN. 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 [11], [13] 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 the relationship between the length of semantic signal and the channel noise 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 $x^c(k)$ into

$$x^c(k) = r^c(k) + \mathbf{n}_{\text{model}}, \quad (11)$$

where $r^c(k)$ is the semantic information selected from the latent semantic codewords, $\mathbf{n}_{\text{model}} \sim \mathcal{CN}(0, \sigma_m^2 \mathbf{I})$ is the model noise. Inspired by that the generative models, e.g., the diffusion model and deep hierarchical VAE, model the latent features with Gaussian distribution. Besides, we generally initialize the model weights with Gaussian distribution and apply the batch normalization/layer normalization to normalize the outputs following $\mathcal{N}(0, 1)$ [31]. 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 $x^c(k)$ is L . By applying the packing sphere theory [32], $x^c(k)$ can be mapped to the L -dimension sphere space as shown in Fig. 3(a). In the Fig. 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 $\sqrt{L(\mu_{\text{max}}^2 + \sigma_m^2)}$, where μ_{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

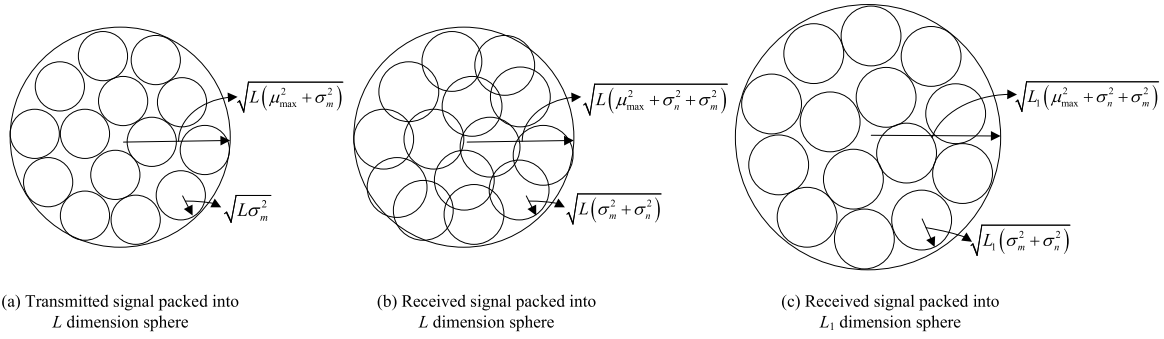


Fig. 3. An example of the relationship between the length of semantic signal and channel noise over AWGN: (a) the transmitted signal of length L packed into L dimension sphere; (b) the received signal of length L over AWGN channels packed into L dimension sphere; (c) the received signal of length L_1 over AWGN channels packed into L_1 dimension sphere.

sphere over the model noise is

$$N = \frac{\left(\sqrt{L(\mu_{\max}^2 + \sigma_m^2)}\right)^L}{\left(\sqrt{L(\sigma_m^2)}\right)^L} = \left(1 + \frac{\mu_{\max}^2}{\sigma_m^2}\right)^{\frac{L}{2}}. \quad (12)$$

After transmitting $\mathbf{x}^c(k)$ over the AWGN channels, the received signals can be represented by submitting (11) into (2),

$$\mathbf{y}^c(k) = \mathbf{r}^c(k) + \mathbf{n}_{\text{model}} + \mathbf{n}_{\text{channel}}, \quad (13)$$

where \mathbf{n} in (2) is re-denoted to $\mathbf{n}_{\text{channel}}$. The $\mathbf{y}^c(k)$ can also be mapped to the L -dimension sphere space shown in Fig. 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.

In order to eliminate the overlapping, one way is to increase the length of $\mathbf{x}^c(k)$ from L to L_1 ($L_1 > L$) 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

$$N = \frac{\left(\sqrt{L_1(\mu_{\max}^2 + \sigma_m^2 + \sigma_n^2)}\right)^{L_1}}{\left(\sqrt{L_1(\sigma_m^2 + \sigma_n^2)}\right)^{L_1}} = \left(1 + \frac{\mu_{\max}^2}{\sigma_m^2 + \sigma_n^2}\right)^{\frac{L_1}{2}}. \quad (14)$$

The semantic codewords only describe the semantic information of the source and are irrelevant to the channel noise, which means that the numbers of semantic codewords in (12) and (14) 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

$$L_1 = L \times \frac{\log\left(1 + \frac{\mu_{\max}^2}{\sigma_m^2}\right)}{\log\left(1 + \frac{\mu_{\max}^2}{\sigma_m^2 + \sigma_n^2}\right)}. \quad (15)$$

With proposition 2, the masked ratio to different SNRs can be computed theoretically. With (15), the asymptotic analysis can be derived into four cases listed below.

- **Case 1:** When $\sigma_n^2 \rightarrow 0$, then $L_1 \rightarrow L$. 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 $\sigma_n^2 \rightarrow \infty$, then $L_1 \rightarrow \infty$. The number of transmitted symbols will lead to infinity. In this case, the semantic communication system experiences an outage.
- **Case 3:** When $\sigma_m^2 \rightarrow 0$, then $L \rightarrow 0$. L_1 only depends on the channel noise and can be computed by

$$L_1 = \frac{2 \log(N)}{\log\left(1 + \frac{\mu_{\max}^2}{\sigma_n^2}\right)}. \quad (16)$$

In this situation, L_1 is computed by the traditional channel capacity and the number of semantic codewords.

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

The key differences between the relationship and traditional channel capacity can be summarized in the following,

- The relationship indicates how much semantic information can be transmitted error-free while the traditional channel capacity indicates how many bits can be transmitted error-free.
- The length of semantic vectors 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.
- When channel noise disappears, the length of semantic vectors has the lower bound, L . The traditional channel capacity does not have such a lower bound.

With the relationship, 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. As shown in Fig. 4, we propose two mask methods subsequently, importance mask method and consecutive mask method.

B. Importance Mask

As shown in Fig. 4(a), the importance mask method introduces the importance-aware model to identify the importance order among the elements of $\mathbf{x}^c(k)$, which can be expressed as

$$\mathbf{p}^c(k) = F(\mathbf{x}^c(k); \boldsymbol{\theta}), \quad (17)$$

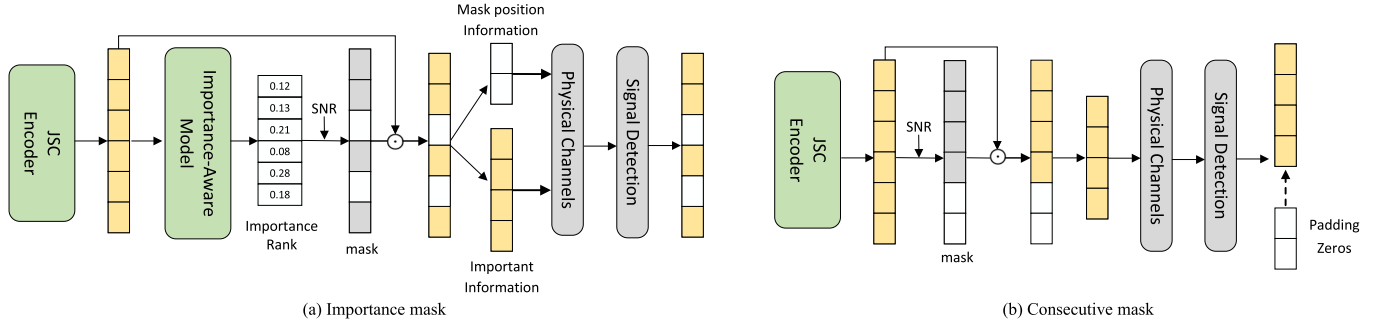


Fig. 4. The proposed two dynamic transmission methods: (a) Importance mask method; (b) Consecutive mask method.

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

By setting the threshold, κ , the mask, $\mathbf{m}^c(k)$, can be computed with the $\mathbf{p}^c(k)$ by

$$\mathbf{m}^c(k)[i] = \begin{cases} 1, & \mathbf{p}^c(k)[i] > \kappa, \\ 0, & \mathbf{p}^c(k)[i] \leq \kappa. \end{cases} \quad (18)$$

where κ can be determined by the L_1 -th element in the sorted importance rank with the descend order, and L_1 is computed by (15).

Then, the masked transmitted signal can be generated by

$$\tilde{\mathbf{x}}^c(k) = \mathbf{x}^c(k) \odot \mathbf{m}^c(k). \quad (19)$$

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

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

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 $\tilde{\mathbf{x}}^c(k)$ and the goal a is employed as the loss function,

$$\mathcal{L}_{\text{MI}} = -I(\tilde{\mathbf{x}}^c(k); a). \quad (20)$$

However, minimizing (20) with gradients descending algorithm is hard since \mathcal{L}_{MI} 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 proven in Appendices B and C, respectively.

Algorithm 2 Importance Mask Training Algorithm

```

1 Function Train Importance Model():
2   Input:  $\{\mathbf{x}^c(k), \mathbf{x}^q, a\}$ . Freeze Mem-DeepSC.
3   Transmitter:
4      $F(\mathbf{x}^c(k); \theta) \rightarrow \mathbf{p}^c(k)$ ,
5     Compute the mask,  $\mathbf{m}^c(k)$ , by (18)
6     Compute the mask signal,  $\tilde{\mathbf{x}}^c(k)$ , by (19),
7     Transmit  $\tilde{\mathbf{x}}^c(k)$  and  $\mathbf{x}^q$  over the air,
8   Receiver:
9     Receive signal and perform signal detection,
10     $C^{-1}(\tilde{\mathbf{x}}^c(k); \gamma) \rightarrow \hat{\mathbf{z}}^c(k)$ , and  $C^{-1}(\hat{\mathbf{x}}^q; \gamma) \rightarrow \hat{\mathbf{z}}^q$ 
11    Update the memory queue,  $\mathbf{Q}(k)$ , with  $\hat{\mathbf{z}}^c(k)$ ,
12    Take the temporal coding for  $\mathbf{M}(k)$  by (7),
13     $S^{-1}([\hat{\mathbf{z}}^q, \mathbf{M}(k)]; \varphi) \rightarrow \hat{a}$ ,
14    Compute CE loss with  $a$  and  $\hat{a}$ .
15    Train  $\theta \rightarrow$  Gradient descent with CE loss.
16  Return:  $F(\cdot; \theta)$ .

```

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

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

With Propositions 3 and 4, the mutual information loss function in (20) can be changed to the cross-entropy loss function in (9).

2) *Training Details*: As shown in Algorithm 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.

C. Consecutive Mask

As shown in Fig. 4(b), the consecutive mask method masks the last consecutive elements in the $\mathbf{x}^c(k)$ 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 $x^c(k)$. 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 $x^c(k)$ 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 $x^c(k)$ following the order of descend importance.

The training of the consecutive mask method only includes one step, which is similar to the Train Whole Network in Algorithm 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.

V. 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.

A. Implementation Details

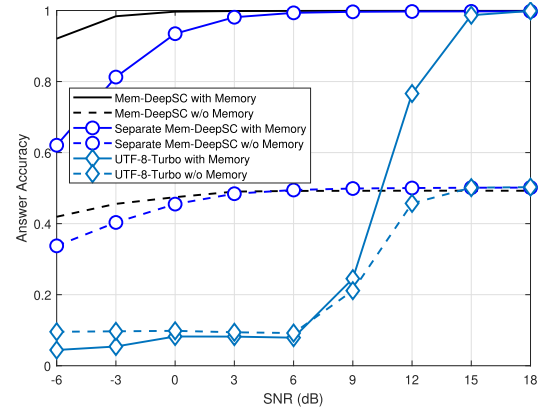
1) *The Dataset*: We choose the *bAbI-10k* dataset [33], including 20 different types of scenario tasks. Each example is composed of a set of facts, a question, the answer, and the supporting facts that lead to the answer. We split the 10k examples into 8k examples for training, 1k examples for validation, and 1k examples for testing.

2) *Training 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 I.

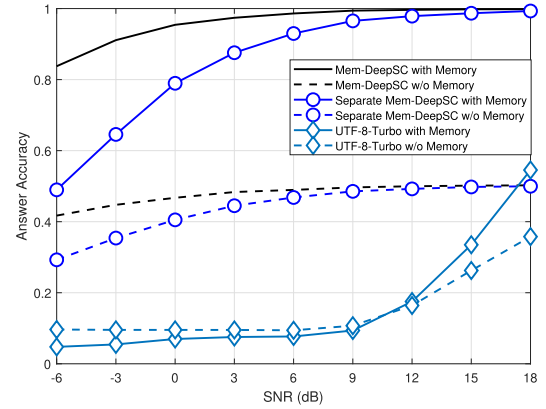
3) *Benchmarks and Performance Metrics*: We adopt the typical source and channel coding method as the benchmark of the proposed Mem-DeepSC, and the random mask method as the counterpart of the proposed two mask methods.

- **Separate Mem-DeepSC**: The semantic codec and channel codec are trained with (9) and (10), respectively.
- **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;
 - Turbo coding for text channel coding, popular channel coding for a small size file;
 - 16-quadrature amplitude modulation as the modulation.
- **Random Mask**: Mask the elements in the transmitted signal randomly.

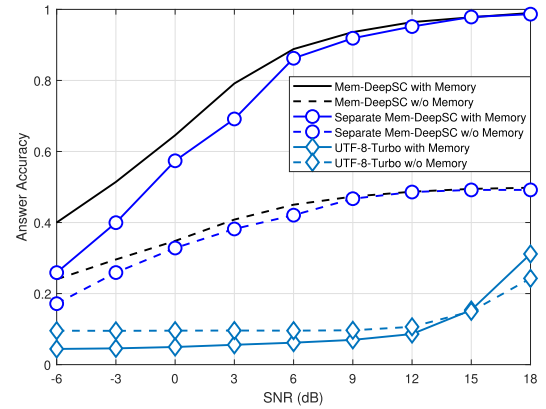
In the simulation, the coding rate is 1/3 and the block length is 256 for the Turbo codes. The coherent time is set as the transmission time for each context in the simulation. We set



(a) AWGN channels.



(b) Rician channels with perfect CSI.



(c) Rician channels with imperfect CSI.

Fig. 5. Answer accuracy comparison between Mem-DeepSC and UTF-8-Turbo with 16-QAM over different channels.

$r = 2$ for the Rician channels and $\mathbf{h} = \mathbf{1}$ for the AWGN channels. In order to compute the required length of semantic vectors, we train multiple Mem-DeepSC with different sizes to find the values of μ_{\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.

B. Memory Semantic Communication Systems

Fig. 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

TABLE I
THE TRAINING SETTINGS

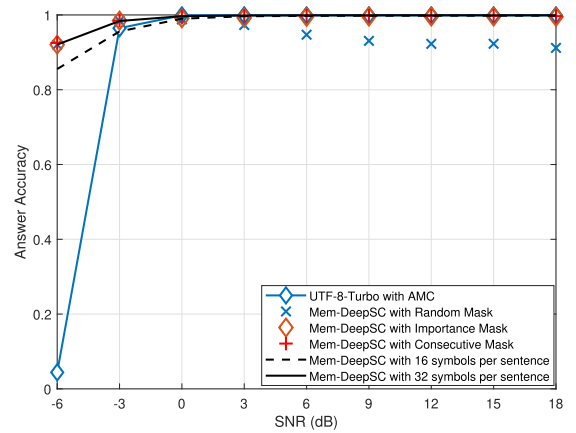
	Batch Size	Learning Rate	Epoch
Train Semantic Codec	200	5×10^{-4}	250
Train Channel Codec	100	1×10^{-4}	50
Train Whole Network	200	5×10^{-4}	30
Train Importance Mask	200	5×10^{-4}	10
Train Consecutive Mask	200	1×10^{-4}	50

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.

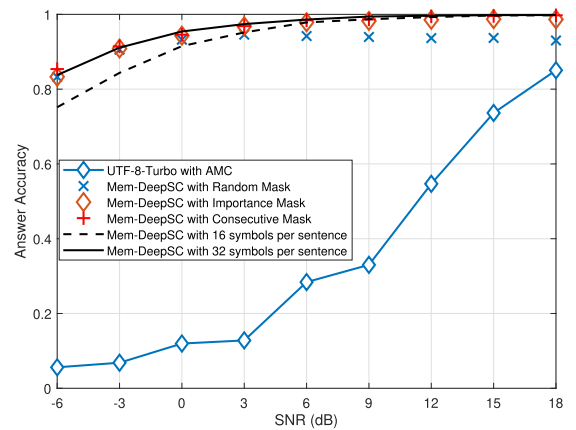
C. The Proposed Mask Methods

Table II 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 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 (15) is validated by the following simulation in Fig. 6.

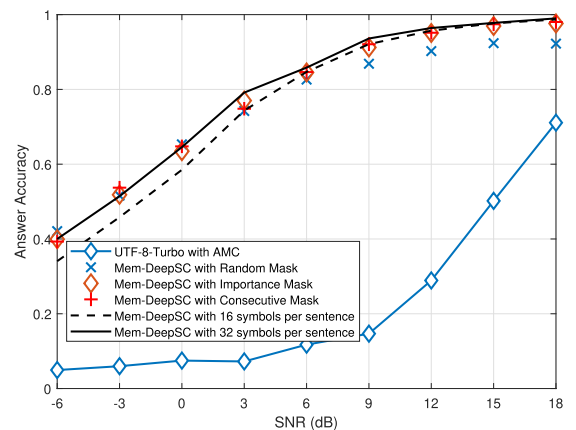
Fig. 6 verifies the effectiveness of the proposed mask strategy. For Mem-DeepSC with no mask, we provided two cases with 16 symbols and 32 symbols per sentence, respectively. Utilizing adaptive modulation and channel coding (AMC) on UTF-8-Turbo can yield comparable answer accuracy to that of Mem-DeepSC over AWGN channels. However, this comes at the expense of a reduced transmission rate. Then comprising the no mask cases with different number 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 the transmission



(a) AWGN channels.



(b) Rician channels with perfect CSI.



(c) Rician channels with imperfect CSI.

Fig. 6. Answer accuracy comparison between Mem-DeepSC with different number of transmitted symbols over different channels.

efficiency. The proposed importance mask and consecutive mask keep the similar answer accuracy as the Mem-DeepSC with 32 symbols per sentence in all SNR regimes over the AWGN and the Rician channels.

However, the random mask experiences significant answer accuracy loss in the high SNR regimes as the number of masked elements increases because some important elements are masked randomly to reduce the answer accuracy. This validates the effectiveness of both proposed mask methods.

TABLE II
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)

VI. CONCLUSION

In this paper, we have proposed a memory-aided semantic communication system, named Mem-DeepSC. The scenario question answer task is taken as the example. 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 the channel noise over AWGN 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.

APPENDIX A

PROOF OF PROPOSITION 1

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

$$Acc = \frac{1}{|B|} \sum_B \langle \mathbf{1}_i, \mathbf{1}_j \rangle, \quad (21)$$

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

$$\mathbf{1}_j = \text{onehot}(\arg \max(\mathbf{l})), \quad (22)$$

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

Since softmax function is the soft function of $\text{onehot}(\arg \max(\cdot))$, the $\mathbf{1}_j$ can be approximated by

$$\mathbf{1}_j \approx \mathbf{p} = \text{softmax}(\mathbf{l}) \quad (23)$$

where \mathbf{p} is the predicted probabilities.

Submitting the (23) to (21), the answer accuracy can be approximated as

$$Acc \approx \frac{1}{|B|} \sum_B \langle \mathbf{1}_i, \mathbf{p} \rangle = \frac{1}{|B|} \sum_B p(a)p(\hat{a}). \quad (24)$$

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

Based on (24), the loss function of answer accuracy can be designed as

$$\mathcal{L}_{Acc} = -\mathbb{E} [p(a)p(\hat{a})]. \quad (25)$$

The derivation of \mathcal{L}_{Acc} for the parameters φ is

$$\nabla_{\varphi} \mathcal{L}_{Acc} = p(\hat{a}) (1 - p(\hat{a})) \nabla_{\varphi} \mathbf{l}. \quad (26)$$

From (26), there exist two optimization directions when $\nabla_{\varphi} \mathcal{L}_{Acc} \rightarrow 0$, i.e., $p(\hat{a}) \rightarrow 0$ and $p(\hat{a}) \rightarrow 1$. However, $p(\hat{a}) \rightarrow 0$ causes worse prediction results and should avoid. In order to make the optimization stable, the \mathcal{L}_{Acc} should be refined. One refined loss function is the cross-entropy loss function given by

$$\mathcal{L}_{CE} = -\mathbb{E} [p(a) \log(p(\hat{a}))]. \quad (27)$$

The derivation of \mathcal{L}_{CE} for the parameters φ is

$$\nabla_{\varphi} \mathcal{L}_{Acc} = (1 - p(\hat{a})) \nabla_{\varphi} \mathbf{l}. \quad (28)$$

Compared (26) and (28), the derivation of \mathcal{L}_{CE} only has one correct optimization direction $p(\hat{a}) \rightarrow 1$, which is more stable during training. Therefore, the proposition 1 is derived.

APPENDIX B

PROOF OF PROPOSITION 2

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

$$I(\tilde{\mathbf{x}}^c(k); a) = H(a) - H(a|\tilde{\mathbf{x}}^c(k)). \quad (29)$$

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

The cross-entropy between the real label and the predicted label given $\tilde{\mathbf{x}}^c(k)$ is

$$H(a; \hat{a}|\tilde{\mathbf{x}}^c(k)) = H(a|\tilde{\mathbf{x}}^c(k)) + D_{KL}(a||\hat{a}|\tilde{\mathbf{x}}^c(k)), \quad (30)$$

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

$$H(a; \hat{a}|\tilde{\mathbf{x}}^c(k)) \geq H(a|\tilde{\mathbf{x}}^c(k)), \quad (31)$$

Submitting (31) into (29), the lower bound of $I(\tilde{\mathbf{x}}^c(k); a)$ can be obtained

$$I(\tilde{\mathbf{x}}^c(k); a) \geq H(a) - H(a; \hat{a}|\tilde{\mathbf{x}}^c(k)). \quad (32)$$

From (32), since $H(a)$ is constant, maximizing the $I(\tilde{\mathbf{x}}^c(k); a)$ can be approximated to minimizing the $H(a; \hat{a}|\tilde{\mathbf{x}}^c(k))$. The lower bound will be closer to $I(\tilde{\mathbf{x}}^c(k); a)$ when the model is trained. Therefore, the proposition 3 is derived.

APPENDIX C

PROOF OF PROPOSITION 3

For the regression task, the mutual information, $I(\tilde{\mathbf{x}}^c(k); a)$, can be expressed as (29).

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

$$\mathbb{E} \left[\left(X - \hat{X}(Y) \right)^2 \right] \geq \frac{1}{2\pi e} e^{2H(X|Y)}. \quad (33)$$

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

$$H(a|\tilde{\mathbf{x}}^c(k)) < \mathbb{E}[\ln |a - \hat{a}(\tilde{\mathbf{x}}^c(k))|] \quad (34)$$

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

Submitting (34) into (29), the lower bound of $I(\tilde{\mathbf{x}}^c(k); a)$ can be obtained

$$I(\tilde{\mathbf{x}}^c(k); a) > H(a) - \mathbb{E}[\ln |a - \hat{a}|]. \quad (35)$$

From (35), since $H(a)$ is constant, maximizing the $I(\tilde{\mathbf{x}}^c(k); a)$ can be approximated to minimizing the $\mathbb{E}[\ln |a - \hat{a}|]$. However, directly minimizing the $\mathbb{E}[\ln |a - \hat{a}|]$ may cause the gradient explosion.

Given the derivation of $\ln |a - \hat{a}|$ for the parameters φ ,

$$\nabla_{\varphi} \ln |a - \hat{a}| = \frac{1}{|a - \hat{a}|} \nabla_{\varphi} \hat{a}. \quad (36)$$

From (36), when $\hat{a} \rightarrow a$, $\nabla_{\varphi} \ln |a - \hat{a}| \rightarrow \infty$. In order to alleviate the gradient explosion, the approximation of $\ln |a - \hat{a}|$ is derived by applying the Taylor series expansion

$$\ln (|a - \hat{a}| - 1 + 1) \approx |a - \hat{a}| - 1. \quad (37)$$

The derivation of (37) for the parameters φ is

$$\nabla_{\varphi} |a - \hat{a}| = \nabla_{\varphi} \hat{a}. \quad (38)$$

Compared (38) and (36), the item, $\frac{1}{|a - \hat{a}|}$, is removed, therefore, the gradient explosion is eliminated. Then, the lower bound of $I(\tilde{\mathbf{x}}^c(k); a)$ can be expressed as

$$I(\tilde{\mathbf{x}}^c(k); a) > H(a) - \mathbb{E}[|a - \hat{a}|]. \quad (39)$$

From (39), maximizing the $I(\tilde{\mathbf{x}}^c(k); a)$ can be approximated to minimizing the $\mathbb{E}[|a - \hat{a}|]$. The lower bound will be closer to $I(\tilde{\mathbf{x}}^c(k); a)$ when the model is trained. Therefore, the proposition 4 is derived.

REFERENCES

- [1] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.
- [2] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Ye Li, "Semantic communications: Principles and challenges," 2021, *arXiv:2201.01389*.
- [3] W. Tong and G. Y. Li, "Nine challenges in artificial intelligence and wireless communications for 6G," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 140–145, Aug. 2022.
- [4] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6G AI-native air interface," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 76–81, May 2021.
- [5] E. Boursoulatz, D. B. Kurka, and D. Gunduz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.
- [6] D. B. Kurka and D. Gunduz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, May 2020.
- [7] M. Jankowski, D. Gunduz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 89–100, Jan. 2021.
- [8] C. Lee, J. Lin, P. Chen, and Y. Chang, "Deep learning-constructed joint transmission-recognition for Internet of Things," *IEEE Access*, vol. 7, pp. 76547–76561, 2019.
- [9] Q. Hu, G. Zhang, Z. Qin, Y. Cai, G. Yu, and G. Y. Li, "Robust semantic communications with masked VQ-VAE enabled codebook," *IEEE Trans. Wireless Commun.*, early access, Apr. 12, 2023, doi: 10.1109/TWC.2023.3265201.
- [10] Y. Yang et al., "Semantic communications with artificial intelligence tasks: Reducing bandwidth requirements and improving artificial intelligence task performance," *IEEE Ind. Electron. Mag.*, early access, May 26, 2022, doi: 10.1109/MIE.2022.3174331.
- [11] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 197–211, Jan. 2022.
- [12] D. Huang, F. Gao, X. Tao, Q. Du, and J. Lu, "Toward semantic communications: Deep learning-based image semantic coding," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 55–71, Jan. 2023.
- [13] T. Tung and D. Gunduz, "DeepWiVe: Deep-learning-aided wireless video transmission," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2570–2583, Sep. 2022.
- [14] S. Wang et al., "Wireless deep video semantic transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 214–229, Jan. 2023.
- [15] P. Jiang, C. Wen, S. Jin, and G. Y. Li, "Wireless semantic communications for video conferencing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 230–244, Jan. 2023.
- [16] P. Tandon et al., "Txt2Vid: Ultra-low bitrate compression of talking-head videos via text," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 107–118, Jan. 2023.
- [17] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, "Deep learning enabled semantic communications with speech recognition and synthesis," *IEEE Trans. Wireless Commun.*, early access, Feb. 6, 2023, doi: 10.1109/TWC.2023.3240969.
- [18] T. Han, Q. Yang, Z. Shi, S. He, and Z. Zhang, "Semantic-preserved communication system for highly efficient speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 245–259, Jan. 2023.
- [19] 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, Jan. 2021.
- [20] H. Xie, Z. Qin, G. Y. Li, and B. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.
- [21] Q. Zhou, R. Li, Z. Zhao, C. Peng, and H. Zhang, "Semantic communication with adaptive universal transformer," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 453–457, Mar. 2022.
- [22] X. Peng et al., "A robust deep learning enabled semantic communication system for text," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 2704–2709.
- [23] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-oriented multi-user semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022.
- [24] M. Tomasello, *Origins of Human Communication*. Cambridge, MA, USA: MIT Press, 2010.

- [25] S. Sukhbaatar, J. Weston, and R. Fergus, “End-to-end memory networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 28, Montreal, QC, Canada, Dec. 2015, pp. 1–9.
- [26] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, “Universal transformers,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–23.
- [27] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.
- [28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. ACL Conf. NAACL HLT*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.
- [29] T. Brown et al., “Language models are few-shot learners,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, Dec. 2020, pp. 1877–1901.
- [30] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [31] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 448–456.
- [32] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [33] J. Weston, A. Bordès, S. Chopra, and T. Mikolov, “Towards AI-complete question answering: A set of prerequisite toy tasks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–14.



Huiqiang Xie (Graduate Student Member, IEEE) received the B.S. degree from Northwestern Polytechnical University and the M.S. degree from Chongqing University. He is currently pursuing the Ph.D. degree with the Queen Mary University of London, U.K. His research interests include semantic communication, massive MIMO, and machine learning. He received the IEEE ICC 2023 Student Travel Grant.



Zhijin Qin (Senior Member, IEEE) was with the Imperial College London, Lancaster University, and the Queen Mary University of London, from 2016 to 2022. She is currently an Associate Professor with Tsinghua University. Her research interests include semantic communications and sparse signal processing. She was a recipient of the 2017 IEEE GLOBECOM Best Paper Award, the 2018 IEEE Signal Processing Society Young Author Best Paper Award, the 2021 IEEE Communications Society Signal Processing for Communications Committee Early Achievement Award, and the 2022 IEEE Communications Society Fred W. Ellersick Prize. She served as a Guest Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Special Issue on Semantic Communications and an Area Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Series. She also served as the Symposium Co-Chair for IEEE GLOBECOM 2020 and 2021. She is serving as an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON COGNITIVE NETWORKING, and IEEE COMMUNICATIONS LETTERS.



Geoffrey Ye Li (Fellow, IEEE) is currently a Chair Professor with the Imperial College London, U.K. Before joining Imperial College London in 2020, he was a Professor with the Georgia Institute of Technology, USA, for 20 years, and a Principal Technical Staff Member with the AT&T Laboratories—Research, NJ, USA, for five years. His general research interests include statistical signal processing and machine learning for wireless communications. In the related areas, he has published over 600 journals and conference papers in addition to over 40 granted patents and several books. His publications have been cited over 58,000 times with an H-index over 110 and he has been recognized as a Highly Cited Researcher, by Thomson Reuters, almost every year. He was awarded a fellow of IET for his contributions to signal processing for wireless communications. He won several prestigious awards from IEEE Signal Processing Society, IEEE Vehicular Technology Society, and IEEE Communications Society, including IEEE ComSoc Edwin Howard Armstrong Achievement Award in 2019. He also received the 2015 Distinguished ECE Faculty Achievement Award from Georgia Tech. He has been involved in editorial activities for over 20 technical journals, including the Founding Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Special Series on ML in Communications and Networking. He has organized and chaired many international conferences, including the Technical Program Vice-Chair of the IEEE ICC’03 and the General Co-Chair of the IEEE GlobalSIP’14, the IEEE VTC’19 Fall, the IEEE SPAWC’20, and IEEE VTC’22 Fall.