2026 International Conference on Computing, Networking and Communications (ICNC): Next generation of mobile networks (6G and beyond)

SA-OOSC: A Multimodal LLM-Distilled Semantic Communication Framework for Enhanced Coding Efficiency with Scenario Understanding

Feifan Zhang*, Yuyang Du*, Yifan Xiang, Xiaoyan Liu, Soung Chang Liew

***Abstract*—This paper introduces SA-OOSC, a multimodal large language models (MLLM)-distilled semantic communication framework that achieves efficient semantic coding with scenario-aware importance allocations. This approach addresses a critical limitation of existing object-oriented semantic communication (OOSC) systems – assigning static importance values to specific classes of objects regardless of their contextual relevance. Our framework utilizes MLLMs to identify the scenario-augmented (SA) semantic importance for objects within the image. Through knowledge distillation with the MLLM-annotated data, our vectorization/de-vectorization networks and JSCC encoder/decoder learn to dynamically allocate coding resources based on contextual significance, i.e., distinguishing between high-importance objects and low-importance according to the SA scenario information of the task. The framework features three core innovations: a MLLM-guided knowledge distillation pipeline, an importance-weighted variable-length JSCC framework, and novel loss function designs that facilitate the knowledge distillation within the JSCC framework. Experimental validation demonstrates our framework's superior coding efficiency over conventional semantic communication systems, with open-sourced MLLM-annotated and human-verified datasets established as new benchmarks for future research in semantic communications.**

***Index Terms*—Semantic communication, multimodal large language models, knowledge distillation**

I. Introduction

With the rapid development of visual applications such as augmented reality (AR) and virtual reality (VR), as well as the growing connectivity demands of the Internet of Things (IoT), the amount of data that needs to be transmitted within a network has increased dramatically [1]. Although the evolution of wireless communication technologies from 1G to 5G has brought significant improvements in packet transmission rates, bit-based communication technologies are approaching the Shannon capacity at the physical layer [2].

Semantic communication has emerged as a highly promising paradigm for next-generation wireless communication to overcome this backdrop [3]. Unlike traditional communication, semantic communication focuses on transmitting the meaning of information rather than its precise representation, thus offering the potential to go beyond the capacity limitations in conventional systems [4].

Although semantic communication systems can significantly reduce bandwidth usage, their coding efficiency and efficacy can potentially be further improved. The authors in [5] proposed a pioneering joint source-channel coding (JSCC) framework that used deep neural networks (DNN) to map an image's pixel representation into a complex-valued vector to be transmitted over the noisy channel. This method relied on a fixed rate encoding method. A subsequent advancement, referred to as NTSCC [6], introduced a variable-length JSCC scheme that divided an image into patches and estimated the information entropy of each patch with an entropy model. The entropy information was utilized to guide the variable-length JSCC process – allocating more coding resources to regions of larger entropy value with higher texture complexity [7]. While this approach marked a significant step forward, it primarily focused on the texture and structural complexity of image regions, ignoring task-relevant semantic importance that is critical for many real-world applications. To address this limitation, task-oriented semantic communication methods were developed [8], [9]. In [8], the authors proposed a reinforcement learning-based adaptive semantic coding framework that extracted semantic importance using semantic segmentation algorithms and allocated coding bits according to semantic importance – a traffic light, for example, is considered important and is given more coding bits for autonomous driving tasks. However, these methods rely only on object segmentation to partition objects and assign predefined semantic importance values that are deterministic and independent of the specific context. Such static approaches fail to account for the dynamic nature of real-world scenarios, where the importance of the same class of object can vary depending on the practical context, leading to inefficiencies in coding resource allocation.

Let us continue with the autonomous driving setting. In a vehicular network, multiple vehicles share real-time road images to build a comprehensive environmental model that supports the vehicle's decision-making [10]. As illustrated in Fig. 1, traditional task-oriented semantic communication methods – referred as Object-Oriented Semantic Communication (OOSC) in this paper – typically treat all objects of certain categories, such as vehicles and pedestrians, as equally important, allocating uniform coding resources to each without considering their specific locations or actual importance within

*F. Zhang and Y. Du contribute equally to this work.

F. Zhang, Y. Du, Y. Xiang, X. Liu and S. C. Liew are with the Department of Information Engineering, The Chinese University of Hong Kong, HKSAR, China. S. C. Liew is the corresponding author (e-mail:soung@ie.cuhk.edu.hk).

The work was partially supported by the Shen Zhen-Hong Kong-Macao technical program (Type C) under Grant No. SGDX20230821094359004. The investigation was conducted in the JC STEM Lab of Advanced Wireless Networks for Mission-Critical Automation and Intelligence funded by The Hong Kong Jockey Club Charities Trust.

the scene. This approach does not align with the cognitive and decision-making patterns of human drivers. In practice, a human driver dynamically adjusts attention based on the specific context of the scene. For example, a driver may pay greater attention to a nearby vehicle in the same driving lane than a parked vehicle at a distance. Similarly, the importance of a pedestrian crossing the road should be higher than the one walking along the road.

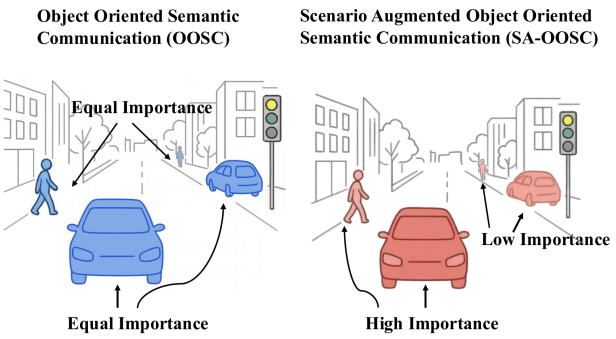

Fig. 1. Illustration of semantic importance identification in OOSC and SA-OOSC under the autonomous driving scenario. In OOSC, task-relevant objects, such as vehicles and pedestrians, are uniformly assigned equal coding importance, regardless of their contextual relevance. In SA-OOSC, an object's semantic importance is decided by the scenario understanding, prioritizing objects such as a nearby vehicle or an imminent pedestrian based on their contextual significance, to achieve more effective and adaptive semantic communication.

To overcome this limitation, we propose the Scenario Augmented OOSC (SA-OOSC) approach as depicted in Fig. 1. This method incorporates scenario understanding to analyze the semantic importance of each object according to its contextual relevance to the ongoing task. To imitate a human understanding of the task scenario, we leverage the image analysis capabilities of multimodal large language models (MLLMs) to automatically label the contextual significance of objects within the scene. In the driving scenario within Fig. 1, for example, the vehicle approaching head-on near the road centerline at close, or the nearby pedestrian crossing the road, should be treated with more attention; while the vehicle parked far away and the pedestrian walking along the sidewalk are less significant in terms of the scenario understanding.

With the massive amount of MLLM-labeled data with scenario understanding, we train the vectorization/inverse vectorization network and JSCC encoder/decoder within our semantic communication framework for knowledge distillation purposes. Our framework, referred to as the SA-OOSC framework, learns from the MLLM-labeled data in the distillation process and gains the ability to allocate more coding resources to high-importance areas (e.g., the nearby pedestrian crossing the road) and assign fewer resources to low-importance areas (e.g., the pedestrian walking along the sidewalk). As we will later show through experiments, SA-OOSC achieves more efficient and effective resource utilization than previous semantic communication systems owing to this scenario-aware importance allocation mechanism.

The rest of this paper uses autonomous driving as the validation scenario. Our major contributions are as follows: First, this paper highlights an important problem overlooked in current semantic communication systems – dynamically determining semantic importance of different objects according to the communication scenario. To address this issue, we put forth SA-OOSC, a framework that leverages MLLM to simulate a human's decision-making process in allocating coding resources according to an object's semantic importance under a specific context.

Second, to leverage the ability of the MLLM, we propose a comprehensive image processing pipeline with variable-length JSCC in the implementation of SA-OOSC. This framework includes three major technical innovations: 1) knowledge distillation with MLLM-annotated data, which equips the framework with scenario understanding capabilities so that it can automatically generate the scenario-augmented semantic importance labels for objects therein; 2) assigning the corresponding importance weights to each image patch and generating their latent representations based on the labeling results; and 3) designing a novel scenario-augmented semantic loss function to optimize the variable-length JSCC scheme, thereby enabling more efficient coding resource allocation than previous methods.

Finally, as the first study to utilize MLLMs for scenario-augmented JSCC, we have open-sourced the scenario-driven semantic communication dataset annotated by GPT-4V. This dataset is positioned as a vital benchmark for training/testing future scenario-augmented semantic communication systems. Further, to validate the accuracy of GPT-4V in assigning semantic importance according to the driving scenario, we invited three experienced drivers to manually annotate a subset of the original dataset as a cross check. The valuable human-annotated data is also made publicly available and can be used to evaluate the alignment of the proposed semantic importance assignment method with human preferences.[1]

II. MLLM for Knowledge Distillation

We consider an image S to be transmitted over a noisy channel, where $S \in \mathbb{R}^{h \times w \times 3}$ represents an RGB image with height h, width w, and three color channels. In our system, the MLLM-based identification module is designed to label the scenario-augmented semantic importance (henceforth referred to as the SA semantic importance) of different regions within this image.

This module starts with an object detection process to identify and localize all objects within the input image, with identification results annotated with bounding boxes (see Fig. 2 for illustrations). In the subsequent step, the annotated image, as well as a detailed textual description of the downstream task, are given to the MLLM integrated in this module. The duty of the MLLM is to assign different levels of semantic importance to each object detected. In the autonomous driving scenario depicted in Fig. 2, for example, the vehicle in

[1] https://github.com/xyfyyds/Semantic-Communication-Cityscapes

the same lane ahead of the ego-vehicle is assigned "high importance", the oncoming vehicle in the opposite lane at a distance is assigned "medium importance", while the vehicle parked at the roadside is assigned "low importance". Without loss of generality, we assign the importance labels of the above three types of objects (i.e., high, medium, and low importance) as 3, 2, and 1, respectively.

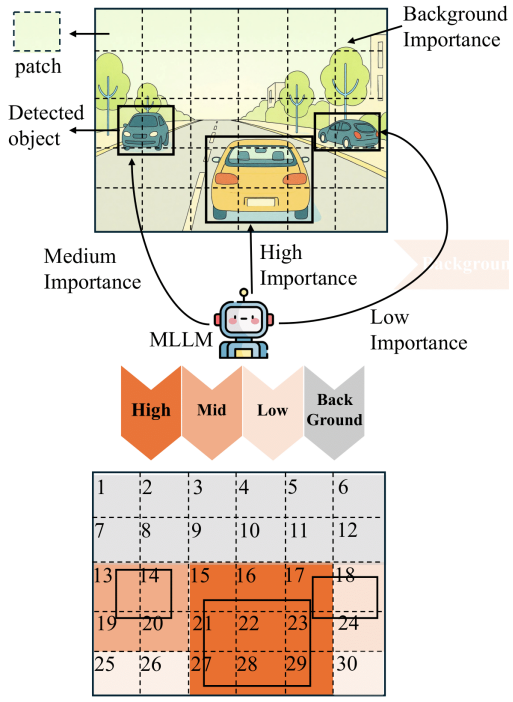

Fig. 2. Illustrations of the object-based (top) and the patch-based (bottom) SA semantic importance.

The above assignments of SA semantic importance are object-based. Yet, image processing frameworks in semantic communication systems, including our scheme to be presented in section III, are patch-based. As a typical pre-processing step before vectorization and encoding, the original image S will be tokenized into L non-overlapping patches, indexed in a raster-scan order (left-to-right, top-to-bottom, see patch indexes in Fig. 2 for example) to form the patch sequence $\mathbf{P} = [P_1, \ldots, P_L]$. To align with later treatments in our image semantic communication framework (see discussion in section III), we must assign each image patch with an associated importance label, forming the importance sequence $\mathbf{I} = [I_1, \ldots, I_L]$ that one-on-one maps to the patch sequence $\mathbf{P}$. Therefore, we need a transformation between object-based importance (i.e., what we have obtained from the object-based importance identification, see the image on the top of Fig. 2) and patch-based importance (i.e., what we need in later image processing, see the image at the bottom of Fig. 2). We now explain how we assign the importance label of an image patch according to the object(s) therein:

Single-object patch: For an image patch containing one object only, we assign the object's importance label to the patch. Patch 13 is an example – the patch contains part of the medium-importance vehicle detected and is therefore identified as a medium-importance patch.

Multiple-object patch: For an image patch containing multiple objects, we let the patch's importance label be the importance label of the most important object therein. One of the most typical examples is Patch 17, which contains both the low-importance vehicle parked at the roadside and the vehicle driving ahead of the ego-vehicle. The importance of Patch 17 is labeled according to the high-importance driving vehicle.

Background patch: For an image patch that does not contain any task-relevant object detected, such as Patch 1 to Patch 12 with far-away trees or the sky only, we assign the lowest importance to that patch. Such patches typically play non-significant roles in the downstream task. We treat them as the image's "background" and assign zero, the lowest semantic importance level, to these patches.

For technical details about the MLLM-empowered semantic importance assignment process, including the deployment of the MLLM based on the GPT-4V model and the prompt engineering we have conducted, we refer readers to Appendix A for our full-version technical report [11] for details. In Section II, we only present the object-based importance and the patch-based importance in Fig. 2 for illustration purposes. It is worth noting that, in practical implementations, the number of patches L is typically much larger than 30 (often in the hundreds or even thousands) to facilitate more efficient encoding. Here, we use 30 patches solely to convey the main concept.

In the following, we use the well-known Cityscapes dataset [12] as a proof-of-concept for the scenario-augmented semantic communication system. Applying the MLLM-based SA importance labeling scheme to the dataset, we obtain high-quality distillable data that incorporates the MLLM's scenario understanding knowledge. We can use this data to train the entire networks within our semantic communication framework to equip the framework with the ability to analyze the SA semantic importance within an image. This process is known as *knowledge distillation*, wherein the scenario understanding ability of a large-scale model can be transformed into smaller-scale ones with a relatively low cost [13].

III. Image Transmission within the SA-OOSC Framework

This section provides a comprehensive overview of the proposed system, including a high-level description of the framework and an outline of the interaction between key modules. We present the SA-OOSC framework in an order corresponding to how an image is transmitted and recovered: first the transmitter design, then the wireless channel model, and finally the receiver design. This section focuses on the image transmission process within SA-OOSC. For model training details, such as the training procedures of each module and the loss function design therein, we refer readers to Appendixes B to H of our full-version technical report [11] for details.

Transmitter Design: With reference to Fig. 3, for an image to be transmitted, we first obtain the patch sequence $\mathbf{P}$ after

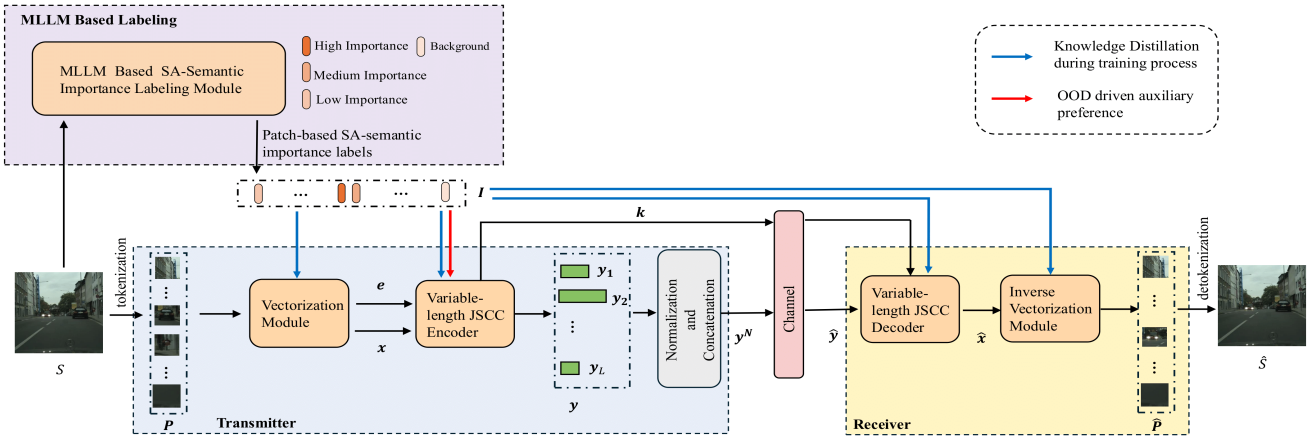

Fig. 3. The block diagram of the SA-OOSC system. SA importance labeling is involved in both the transmission of a tested image (red arrow) and the training of vectorization/de-vectorization/JSCC encoding (blue arrows). Section III only focuses on the signal processing pipeline for image transmissions. Model training details (i.e., related to blue arrows) are available in Appendixes B to H of our technical report [11].

image tokenization, and we next process sequence $\mathbf{P}$ with a revised hyperprior variational (HV) vectorization network. The HV method, originally proposed in [7], can effectively model the spatial dependencies in the patch sequence $\mathbf{P}$ for improved image compression efficiency. We denote the latent representation sequence generated by the HV vectorization as $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$, where $\mathbf{x}_i$ is a c-dimensional real-valued embedding vector and c is the dimension hyperparameter of the embedding. In addition to the latent representation sequence $\mathbf{x}$, the HV vectorization also has a built-in entropy model that estimates the discrete entropy vector $\mathbf{e} = [e_1, e_2, \ldots, e_L]$ for given image patches according to the distribution parameter obtained from latent representations $\mathbf{x}$.

This paper revises the conventional HV training framework in [7] by employing the MLLM as a teacher model and distilling its knowledge of SA semantic importance into the HV vectorization network serving as the student model. Specifically, the SA-semantic importance labels $\mathbf{I}$ from the MLLM are used as additional supervisory labels to guide the learning process of the student network. This revised training process allows the network to effectively learn the MLLM's scenario-augmented knowledge, making entropy $\mathbf{e}$ characterize not only the structural/texture complexity as designed in [7], but also the scenario-relevant semantic importance of each patch, as emphasized in this paper. Henceforth, we refer to this revised entropy as *SA entropy*.

Training details of the revised HV vectorization networks, as well as the re-designed loss function that facilitates the MLLM knowledge distillation, are available in Appendix D of our technical report [11]. Here in Section III, we explain only the superiority of SA entropy over the conventional entropy from [7] with the following example: in autonomous driving scenarios, a patch depicting the background may contain rich texture details such as distant buildings of different shapes, making the entropy of this patch relatively high when the conventional scenario-unaware HV network is considered. The revised HV networks distilled by the MLLM, however, understand that the shape of a background building, though rich in structural details, is irrelevant to the current driving scenario. Therefore, the SA entropy should be assigned to this background patch to facilitate later entropy-based patch compression in the JSCC encoding network.

With the above discussion, we write the HV vectorization process as: $\{\mathbf{x}, \mathbf{e}\} = F_e(\mathbf{P})$, where $\mathbf{P}$ is the input patch sequence, $\mathbf{e}$ is the SA entropy vector, and $\mathbf{x}$ is the latent representation sequence vectorized from the patch sequence.

Subsequently, $\mathbf{x}$ and $\mathbf{e}$ are input into a variable-length JSCC encoder, which adaptively encodes $\mathbf{x}_i$ into a variable-length vector $\mathbf{y}_i$ with simultaneous consideration of both source coding and channel coding in conventional wireless communication systems. Ideally, as in previous investigations (e.g., [6], [14]), the length of vector $\mathbf{y}_i$ reflects the coding resource allocated to image patch i, and therefore it should be proportional to e_i, the SA entropy of the patch.

However, the above ideal setup overlooks potential out-of-distribution (OOD) issues that may arise in real-world applications. OOD refers to situations where the model encounters samples or scenarios that fall outside the distribution of its training data. Such data may appear rarely or not at all during training but could occur occasionally in practical environments, bringing potential risk to the system if anti-OOD approaches are not considered. For example, autonomous vehicles may encounter novel traffic signs, construction machinery, or unusual obstacles arising from unexpected incidents. If the system fails to assign appropriate semantic importance to these OOD objects, its performance in real-world testing scenarios can be severely compromised. Recent studies have demonstrated that, owing to large-scale training datasets and diverse data sources, language models possess strong generalization capabilities in image analysis and are better equipped to handle the OOD problem [15]. Thus, in addition to SA-entropy, our system further incorporates the SA-semantic importance labels assigned by the MLLM as an auxiliary preference to more accurately determine the length of the signal vector $\mathbf{y_i}$. This

strategy is expected to combat the potential negative impact of OOD data on overall system performance.

Therefore, for each patch i, we jointly utilize its SA-entropy e_i and SA-semantic importance label I_i to determine k_i, the length of vector $\mathbf{y}_i$. Specifically, we let

$$k_i = Q_0 \left[C_1(e_i) + C_2(I_i) \right], \tag{1}$$

where $C_1(e_i)$ is the conventional vector length term that is proportional to the SA entropy (see (14) in our technical report [11] for implementation details). And $C_2(I_i)$ is the auxiliary vector length term developed to overcome the OOD problem with the MLLM (see (15) in [11] for details); and Q_0 denotes a simple scalar quantizer whose duty is to map the sum $C_1(e_i)+C_2(I_i)$ into an integer within a pre-defined set of vector lengths supported by the communication system.[2]

After obtaining $\mathbf{k} = [k_1, k_2, \ldots, k_L]$, we have, in essence, completed the dynamic encoding resource allocation with the help of the SA semantic importance. The rest of the task resembles the conventional JSCC methodology, i.e., coding image patch i with the constraint of k_i under the JSCC setup. Importantly, the MLLM's scenario understanding is also involved in the training process by supervising the evaluation of the image distortion loss. In other words, the training of the JSCC encoder also follows the knowledge distillation setup discussed above. For details about the image distortion evaluation, we refer interested readers to Appendix B of our technical report [11], while information about the loss function design is in Appendix G of [11].

With the above discussion, we formally write the JSCC encoding process as $\mathbf{y} = G_e(\mathbf{e}, \mathbf{x}, \mathbf{I})$. After the JSCC encoder, the complex-valued output $\mathbf{y}_i \in \mathbb{C}^{k_i}$. To satisfy the power constraint for signal transmission, vector $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L]$ needs to undergo a power normalization process. The normalized channel input signal $\mathbf{y}^N$ satisfies the following constraint:

$$\sum_{i=1}^{L} \|\mathbf{y}_i\|^2 \Big/ \sum_{i=1}^{L} k_i = 1, \tag{2}$$

where $\|\mathbf{y}_i\|^2$ denotes the squared norm of the complex-valued vector y_i, and k_i is the encoding length for the corresponding patch. The normalized vector $\mathbf{y}^N = [\mathbf{y}_1^N, \mathbf{y}_2^N, \ldots, \mathbf{y}_L^N]$ is then transmitted over the noisy channel. Meanwhile, we note that the variable-length JSCC decoding process requires a priori knowledge about the vector length of each image patch. Therefore, we followed the setup in [6] and transmitted $\mathbf{k} = [k_1, k_2, \ldots, k_L]$ through the channel.[3]

[2]We refer readers to [6] for theoretical justifications about the necessity of the scalar quantizer, and refer readers to (16) in Appendix F of our technical report [11] for implementation details of the scalar quantizer in this paper.

[3]We note that the a priori information of k_i is vital for the JSCC decoding of patch i, just like the role of packet headers in ethernet network. To ensure reliable image reconstructions in the receiver, we followed the setup in [6] and transmitted vector $\mathbf{k}$ with the communication scheme of low modulation order and high redundancy coding.

Wireless Channel: When transmitted over the channel, the power-normalized signal $\mathbf{y}^N$ is passed through an additive white Gaussian noise (AWGN) channel, which is modeled as:

$$\hat{\mathbf{y}} = H(\mathbf{y}^N) = \mathbf{y}^N + \mathbf{n}, \tag{3}$$

where $\mathbf{n} \sim \mathcal{CN}(0, \sigma_n^2)$ represents Gaussian noise with noise power σ_n^2, and $\hat{\mathbf{y}}$ denotes the received signal at the decoder.

Receiver Design: At the receiver side, the decoder aims to reconstruct the original image from the noise-corrupted received signal $\hat{\mathbf{y}}$, following a decoding process that mirrors the encoding procedure. Specifically, the received signal $\hat{\mathbf{y}}$ first passes through a variable-length JSCC decoder. Subsequently, the JSCC decoding output $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_L]$ is fed into the inverse HV vectorization module to generate the reconstructed patch sequence $\hat{\mathbf{P}} = [\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_L]$, which are then concatenated to form the reconstructed image $\hat{S}$.

We note that the JSCC decoder is jointly trained with the JSCC encoder, and the inverse HV vectorization network is also jointly trained with the HV vectorization network. That is, knowledge distillation is also applied in the training of receiver-side networks. For technical details about the joint network training, we refer readers to Appendix H of [11].

IV. Model Training

As illustrated in Fig. 3, this paper redesigns several key components within the conventional variable-length JSCC framework [6], [14]. Due to space constraints, this section provides only an outline of our contributions in model training. We refer interested readers to appendices of our full-version technical report [11] for details.

First, we present a revised evaluation scheme for image reconstruction distortion in Appendix B of [11]. This evaluation method emphasizes the reconstruction of scenario-relevant information through a SA weighted mechanism, serving as a critical loss term in the network training process. Subsequently, Appendix C of [11] presents the architectural improvements made to the conventional HV vectorization and inverse vectorization networks, while Appendix D therein details their loss function design. And this is followed by the network implementation details in Appendix E. Meanwhile, variable-length JSCC encoder and decoder are trained, with their implementation details and loss functions introduced in Appendices F and G of [11], respectively. The JSCC encoder addresses the OOD problem identified in [15] through an auxiliary loss term based on the MLLM's importance identification. Finally, Appendix H of [11] presents the overall system training framework and hyperparameter configuration.

V. Experiments

This paper has comprehensively evaluated the two major technical contributions of this paper: 1) the SA-semantic importance identification empowered by MLLM, and 2) the SA-OOSC framework. Due to page limit, we cannot present all experimental details here. Discussion about the MLLM-empowered SA semantic importance identification are given in Appendix I of [11], while the rest of this section details

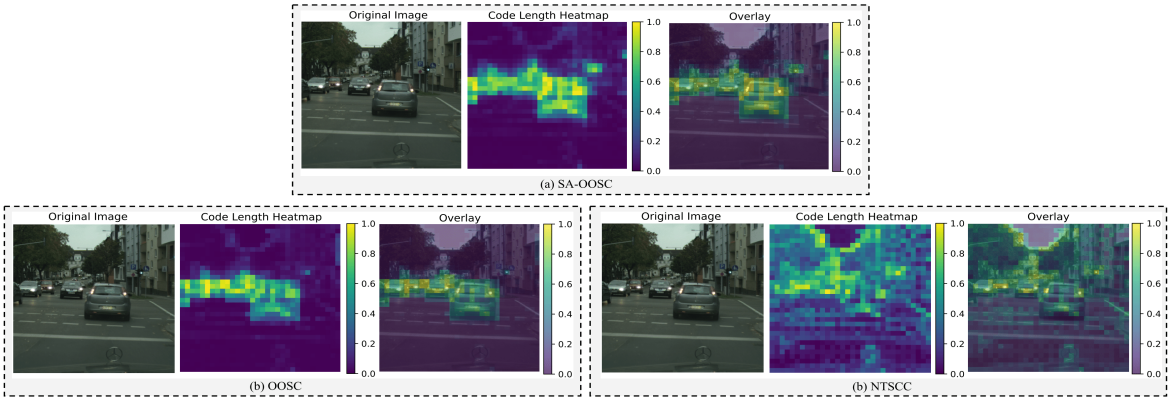

Fig. 4. Comparison of code rate distributions under different schemes. For each scheme, the left column shows the original image, the middle column presents the code length heatmap, and the right column displays the overlay of the image and its code length distribution. The heatmap reflects the allocation of channel resources across image patches, with brighter colors indicating a larger number of channel symbols assigned. The overlay intuitively visualizes the spatial regions prioritized by each encoder. To ensure a fair comparison, all schemes are configured to have similar CBR (0.0228, 0.0230 and 0.0241 for SA-OOSC, OOSC, and NTSCC, respectively).

experiment and associated results for testing the SA-OOSC framework. Experimental setups, including dataset applied, baseline methods, and evaluation metrics, are given as follows.[4]

Dataset: We utilized the Cityscapes dataset [12] to evaluate the SA-OOSC framework, using the autonomous driving scenarios as a proof-of-concept. The dataset consists of 2975 training images and 1525 testing images, each with a resolution of 2048×1024 pixels.

Baseline Methods Benchmarked: We compared our method with the representative baselines as follows: 1) the fixed-rate image transmission scheme (Deep JSCC) proposed in [5], 2) the adaptive-rate transmission scheme (NTSCC) developed in [6], and 3) the OOSC scheme introduced in [8].

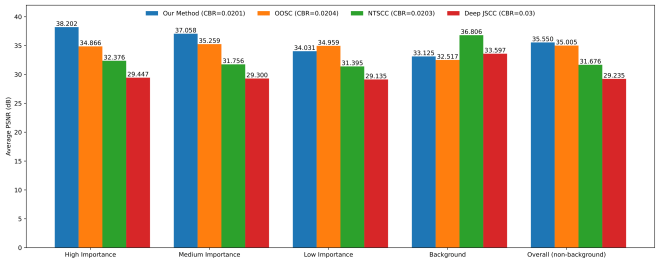

Fig. 5. Image reconstruction evaluation for batches with different semantic importance.

Evaluation Metrics: Following [5], Channel Bandwidth Ratio (CBR) was considered as the main metric for evaluating the compression efficiency of image transmission. The original bandwidth was defined as $n = 3hw$, where h and w denoted the height and width of the image, respectively. For an image with L patches, the encoded image could be represented by L vectors of varying lengths, with the i-th vector having length k_i. The total bandwidth cost was $m = \sum_{i=1}^{L} k_i$, and the system's CBR was defined as $R = m/n$ to represent the average number of available channel symbols per source dimension. PSNR was used to assess the quality of reconstructed image patches.

We start with a case study at the transmitter side. Fig. 4 compares our method with OOSC and NTSCC. The fixed-rate Deep JSCC approach is not included here in this case study, as it does not support semantic-driven adaptive rate allocation. In this experiment, an image was randomly selected from the dataset to illustrate the normalized code rate distributions under different encoding strategies.

Visualizations in Fig. 4 demonstrate that our method could more precisely allocate channel resources to patches with high SA semantic importance, effectively reducing bandwidth assigned to less relevant background areas. For example, in this urban driving scene, the vehicle immediately in front of the ego car should be assigned the highest semantic importance, while other vehicles farther away, although still relevant, should have lower priority. SA-OOSC adaptively captured this semantic hierarchy and reflected such priority in channel resource allocation. The OOSC scheme, on the other hand, allocated resources rather coarsely based on predefined object categories (e.g., vehicles, pedestrians), lacking the ability to distinguish importance among similar objects in different spatial contexts. Hence, it failed to assign sufficient resources to the most critical vehicle (i.e., the closest car), while expending unnecessary bandwidth on less relevant vehicles in other lanes. The NTSCC scheme, while capable of adaptive rate allocation, did not explicitly incorporate task-relevant semantic importance, distributing resources primarily according to image texture and detail – substantial resources were wasted on irrelevant regions such as background buildings.[5]

[4]For the hyperparameter setup of our model tested in the experimental section, we refer readers to Appendix H of [11]. Regarding the implementation of baseline methods, we refer readers to our baseline implementation open-sourced along with the data benchmark.

[5]Besides the transmitter-side case study presented, we also have a case study that took the noisy channel and image reconstruction into account. We refer readers to Appendix J of [11] for details of this receiver-side case study.

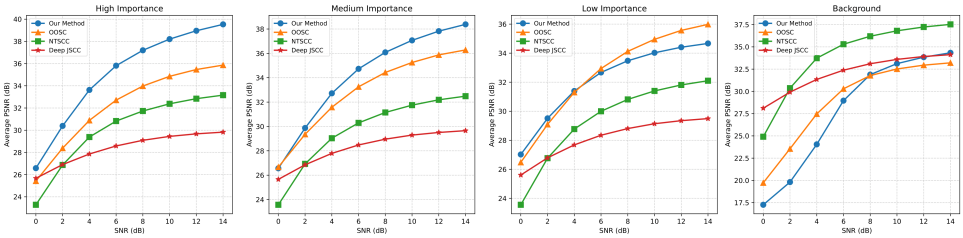

Fig. 6. Image reconstruction across semantic importance levels under varying SNR conditions.

Experiments in Fig. 5 further evaluated the performance of the proposed SA-OOSC method with the whole dataset, in which PSNR was compared across image patches with different levels of semantic importance. Both the training and the testing phase assumed 10 dB SNR. As the figure shows, SA-OOSC achieved superior overall channel bandwidth efficiency, with its 0.0201 CBR lowering than other baseline methods. Despite the low channel bandwidth occupation, SA-OOSC consistently achieved the best reconstruction performance in both high and medium semantic importance patches. In low-importance patches, OOSC slightly outperformed SA-OOSC, while in background patches, both SA-OOSC and OOSC exhibited low PSNR, as these methods intentionally allocated fewer resources to background content. NTSCC and Deep JSCC achieved higher PSNR in the background, although the information therein was less important. From an overall (non-background) perspective, SA-OOSC achieved an average PSNR of 35.55 dB, which was comparable to OOSC (35.01 dB) but significantly higher than NTSCC (31.68 dB) and Deep JSCC (29.24 dB). These results clearly demonstrate that SA-OOSC could prioritize the reconstruction of patches with scenario-relevant information, maintaining a high image quality with the lowest CBR.

Fig. 6 benchmarks SA-OOSC with baseline methods under various SNR. Here we analyzed the average PSNR of different methods across different patches as SNR varies, with all models trained with 10 dB SNR. For patches with high semantic importance, SA-OOSC consistently achieved the best PSNR. In medium-importance patches, SA-OOSC maintained the highest PSNR across all SNR settings, demonstrating strong noise robustness in critical target areas. For low-importance patches, OOSC and SA-OOSC performed similarly, both outperforming NTSCC and Deep JSCC. For background patches, our method did not perform as well as NTSCC and Deep JSCC. Yet, due to limited impacts of background patches on decision-making, the minor degradation in reconstruction quality is acceptable.

VI. Conclusion

This paper proposes a MLLM-empowered semantic identification method with task scenario understanding. Meanwhile, we develop SA-OOSC, an image semantic communication framework that leverages the MLLM-based identification for knowledge distillation, enabling the framework to understand task-relevant scenarios. Experiments show that SA-OOSC achieves superior coding efficiency while preserving all important scenario-relevant information, demonstrating robust image reconstruction across various channel conditions. Further, we open-source the MLLM-annotated dataset with manual sampling and cross-checking, which represents an vital data contribution to the community of semantic communication.

References

[1] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11 891–11 915, 2021.

[2] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[3] T. M. Getu, G. Kaddoum, and M. Bennis, "Semantic communication: A survey on research landscape, challenges, and future directions," *Proc. IEEE*, vol. 112, no. 11, pp. 1649–1685, 2024.

[4] Y. Shao, Q. Cao, and D. Gündüz, "A theory of semantic communication," *IEEE Trans. Mob. Comput.*, 2024.

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

[6] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, "Nonlinear transform source-channel coding for semantic communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2300–2316, 2022.

[7] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[8] 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, 2023.

[9] Y. Wang, S. Guo, Y. Deng, H. Zhang, and Y. Fang, "Privacy-preserving task-oriented semantic communications against model inversion attacks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 10 150–10 165, 2024.

[10] H. Ngo, H. Fang, and H. Wang, "Cooperative perception with v2v communication for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11 122–11 131, 2023.

[11] F. Zhang, Y. Du, Y. Xiang, X. Liu, and S. C. Liew, "SA-OOSC: A multimodal llm-distilled semantic communication framework for enhanced coding efficiency with scenario understanding," *arXiv preprint arXiv:2509.07436*, 2025.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] K. Chen, Y. Du, T. You, M. Islam, Z. Guo, Y. Jin, G. Chen, and P.-A. Heng, "Llm-assisted multi-teacher continual learning for visual question answering in robotic surgery," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 772–10 778.

[14] S. Wang, J. Dai, X. Qin, Z. Si, K. Niu, and P. Zhang, "Improved nonlinear transform source-channel coding to catalyze semantic communications," *IEEE J. Sel. Top. Signal Process.*, vol. 17, no. 5, pp. 1022–1037, 2023.

[15] F. Zhang, Y. Du, K. Chen, Y. Shao, and S. C. Liew, "Out-of-distribution in image semantic communication: A solution with multimodal large language models," *IEEE Trans. Mach. Learn. Commun. Netw.*, pp. 997–1013, 2025.