

Channel-aware Partial Fine-Tuning for AirPooling based Semantic Communications

Nan Wang, Yinglei Teng, *Senior Member, IEEE*, Jiachen Sun,

Beijing Key Laboratory of Space-ground Interconnection and Convergence

Beijing University of Posts and Telecommunications (BUPT), Xitucheng Road No.10, Beijing, China, 100876.

Email: wangnan_26@bupt.edu.cn, lilytengtt@gmail.com, sunjiachen@bupt.edu.cn

Abstract—Recently, multiple devices in the Internet of Things (IoT) collaborate to perceive the environment, offering richer information for intelligent decisions. However, due to the limited network resources, the aggregation of massive sensory data over the dynamic channel environments presents a critical challenge to the real-time learning performance. In this work, we propose a channel-aware partial fine-tuning scheme for the AirPooling based semantic communication system, enabling rapid delivery of accurate models in dynamic channel environments. To achieve efficient information aggregation, Over-the-Air Pooling (AirPooling) is leveraged to facilitate the pooling of multiple single-view features during concurrent transmission, with channel encoding and decoding performed via autoencoder-based neural networks. In addition, to address learning performance degradation caused by deteriorating channels, a partial fine-tuning method is proposed, which utilizes layer freezing to reduce computation overhead. Simulation results show that in dynamic channel environments, the proposed scheme achieves higher inference performance with lower fine-tuning overhead compared to baselines.

Index Terms—over-the-air computation, semantic communication, model fine-tuning, multi-view learning.

I. INTRODUCTION

THE convergence of Internet of Things (IoT) and artificial intelligence (AI) has ushered in an era of unprecedented intelligence, where valuable information can be rapidly extracted from massive data for intelligent decision-making [1]. In many AI-empowered emerging applications, many heterogeneous devices collaborate to sense their surroundings and upload the multi-view or multi-modal data to the edge server for comprehensive analysis. However, compared to single-source analysis, multi-source data processing not only needs to address challenges such as data alignment [2] and incompleteness [3] but also imposes greater transmission demand on the network for data aggregation. Therefore, developing efficient data management schemes to enable low-latency collaborative learning has become a primary concern.

To alleviate the impact of the wireless channel on the end-to-end task performance, semantic communications [4], based on joint source-channel coding and decoding (JSCC), not only guarantees the semantic integrity of data transmission, but also greatly reduces the transmission overhead. For multimodal tasks, Zhang *et al.* [5] proposed a unified end-to-end framework, in which the encoded data from different modalities are transmitted through their respective channels to the receiver, followed by aggregation and input to the

channel decoder for subsequent processing. For the 3D visual reconstruction task, Fu *et al.* [6] transmitted encoded signals from multiple views to the receiver, where each view's signal is decoded in parallel via deconvolution and then fed into a joint context transfer module to enhance reconstruction quality by exploiting inter-view correlations. Despite coupling learning and communication, the existing “receive-then-aggregate” in collaborative semantic communications may either face degraded decoding performance due to failures in single-source information sampling and transmission or suffer from higher latency caused by the use of separate transmission channels.

In fact, many collaborative tasks focus on the aggregated information rather than the individual source data. In this regard, over-the-air (OTA) computation [7] has emerged as a highly promising communication-efficient solution, leveraging signal superposition to fuse multiple single-source data during concurrent transmissions. Recent works have been exploring the OTA technique to expedite end-to-end multi-view learning. Liu *et al.* [9] proposed an over-the-air pooling (AirPooling) scheme, which leverages the waveform superposition property of a multiple access channel to perform multi-view pooling during transmission, and optimizes the configuration parameters to balance the aggregation error and the approximation error of Max-AirPooling. Chen *et al.* [10] investigated the impact of channel conditions and view quantity on the inference accuracy of the AirPooling scheme and theoretically demonstrated that the AirPooling scheme achieves superior noise suppression capability compared to the OMA scheme. However, all the OTA transceiver designs in the aforementioned works concentrate on the bit error, which is not fully equivalent to task-specific performance such as inference accuracy in learning tasks. The lack of research on the impact of OTA aggregation error on final task performance serves as the motivation for our current work.

In this paper, we propose a channel-aware partial fine-tuning scheme for the AirPooling based semantic communication system to expedite the collaborative learning. First, a collaborative learning framework for multi-view object recognition is constructed, where multiple sensing devices send the joint source-channel coded signals of the single-view features to the edge server for subsequent fusion and recognition. Then, to improve the communication efficiency, the AirPooling scheme enables concurrent transmission of single-view encoded features over a shared channel, allowing the receiver to directly

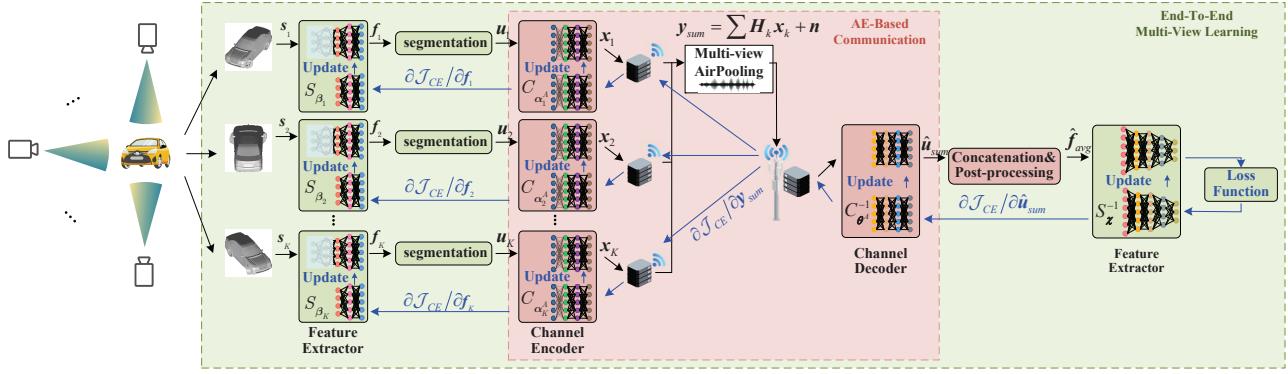


Fig. 1: The framework of collaborative semantic communication system.

obtain a fused feature for subsequent classification. In addition, to mitigate the performance degradation of the pre-trained model under changing channel conditions, we propose a partial fine-tuning approach with layer freezing, effectively reducing transmission and computation overhead. Experimental results show that in dynamic environments, the proposed scheme achieves higher inference performance with lower fine-tuning overhead compared to baselines.

II. COLLABORATIVE SEMANTIC COMMUNICATION SYSTEM

As shown in Fig. 1, we consider a multi-device collaborative learning system for multi-view object recognition, which consists of an edge server equipped with N_R antennas and K sensing devices equipped with N_T antennas. Object recognition is implemented by a multi-view convolutional neural network (MVCNN) [11], which splits the standard convolutional neural network (CNN) into a feature extractor model and a global classifier model, deployed on the sensing devices and edge server, respectively. The feature extractor extracts viewpoint-specific features, while the global classifier makes global predictions based on aggregated features pooled from multiple single views.

A. Feature Extractor and Channel Encoder

Define the image captured from the k -th viewpoint as s_k , then the extracted single-view feature map $f_k \in \mathbb{R}^{N_F}$ can be expressed as

$$f_k = S_{\beta_k}(s_k), \quad (1)$$

where $S_{\beta_k}(\cdot)$ is the feature extractor network with the parameter β_k .

After the feature extraction, the single-view features are encoded into transmit signals through channel encoding. We consider an autoencoder (AE) based communication framework [12], in which neural networks are employed to learn the encoding and decoding of the transmitted message. The single-view feature is divided into five vectors, each of which is independently encoded and transmitted to the edge server. The source vector for each transmission is denoted by $u_k \in \mathbb{R}^{N_U}$, i.e., $N_U = N_F/\tau$. Assume a closed-loop MIMO system where

the channel state information (CSI) is known to both the sensing devices and edge server. The source vector u_k is concatenated with the channel state and then fed into the channel encoder network $C_{\alpha_k}(\cdot) : \mathbb{R}^{N_U} \times \mathbb{C}^{N_R \times N_T} \rightarrow \mathbb{C}^{N_T \times 1}$,

$$\mathbf{x}_k = C_{\alpha_k}(u_k, \mathbf{H}_k), \quad (2)$$

where α_k is the channel encoder parameter and $\mathbf{H}_k \in \mathbb{C}^{N_R \times N_T}$ is the CSI from the sensing device k to the edge server¹. For the power constraint, a normalization layer is added before the output of the channel encoder to ensure that the transmit power of each sensing device does not exceed P_T , i.e., $\mathbb{E}[\|u_k\|^2] \leq P_T$.

To avoid multi-user interference, orthogonal multiple access (OMA) is employed to transmit the encoded signals over their respective channels to the edge server,

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}, \quad (3)$$

where \mathbf{n} denotes the additive white Gaussian noise with the distribution $\mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_{N_R})$.

B. Channel Decoder and Global Classifier

Similar to the channel encoder, the received signal at the edge server is input to the channel decoder network $C_{\theta}^{-1}(\cdot) : \mathbb{C}^{N_R \times 1} \times \mathbb{C}^{N_R \times N_T} \rightarrow \mathbb{R}^{N_L}$ through the concatenation with the channel state,

$$\hat{\mathbf{u}}_k = C_{\theta}^{-1}(\mathbf{y}_k, \mathbf{H}_k), \quad (4)$$

where θ is the channel decoder parameter. After τ timeslots, the recovered signals are recombined into single-view features $\hat{f}_k \in \mathbb{R}^{N_F}$, and multiple single-view features are average-pooled into an aggregated feature,

$$\hat{f}_{avg} = \frac{\sum_{k=1}^K \hat{f}_k}{K}. \quad (5)$$

¹The complex-valued CSI is first transformed into a real-valued vector of length $2N_R N_T$, which is then fed into the channel encoder network. The channel encoder outputs a real-valued vector of length $2N_T$, where the first half is treated as the real part and the second half as the imaginary part to form the complex-valued source vector.

Then, the aggregated feature is fed into the global classifier to obtain the predicted results,

$$\hat{\mathbf{p}} = S_{\chi}^{-1}(\hat{\mathbf{f}}_{avg}), \quad (6)$$

where $S_{\chi}^{-1}(\cdot)$ is the global classifier network with the parameter χ .

C. End-to-End Training Procedure

For the multi-view object recognition task, the categorical cross-entropy loss function is adopted as the training criterion for the end-to-end system, rather than training the learning and communication modules separately. The loss function is defined by

$$\mathcal{J}_{CE}(\beta, \alpha, \theta, \chi) = - \sum_{z=1}^Z \mathbf{p}(z) \log [\hat{\mathbf{p}}(z)], \quad (7)$$

where Z denotes the total number of classes in the dataset, and $\hat{\mathbf{p}}(z)$ represents the one-hot encoding of the ground truth labels.

III. MULTI-VIEW AIRPOOLING AND PARTIAL FINE-TUNING SCHEME

Based on the aggregation requirements of multi-view learning, we utilize an AirPooling aggregation method to further enhance the learning efficiency. Then, considering the performance degradation of the pre-trained model caused by deteriorating channel conditions, a channel-aware partial fine-tuning scheme is designed for the end-to-end learning system.

A. AirPooling for Multi-View Aggregation

In our multi-view learning system, as the focus is on the pooled feature rather than on each individual one, the “receive-then-aggregate” of single-view features lead to unnecessary resource consumption. Therefore, leveraging the superposition property of the multiple-access channel, we utilize the OTA computation to achieve multi-view pooling during the transmission process. Similar to the OMA mechanism, the scalar \mathbf{u}_k is encoded via the OTA channel encoder network $C_{\alpha_k^{OTA}}(\cdot) : \mathbb{R}^{N_U} \times \mathbb{C}^{N_R \times N_T} \rightarrow \mathbb{C}^{N_T \times 1}$,

$$\mathbf{x}_k = C_{\alpha_k^{OTA}}(\mathbf{u}_k, \mathbf{H}_k). \quad (8)$$

Then, multiple sensing devices simultaneously transmit the encoded signals to the edge server,

$$\mathbf{y}_{sum} = \sum_{k=1}^K \mathbf{H}_k \mathbf{x}_k + \mathbf{n}. \quad (9)$$

Unlike the OMA mechanism, AirPooling does not require to detect individual signals at the edge server. Instead, the aggregated signal is treated as the sum of single-view signals and is directly fed into the OTA channel decoder $C_{\theta^{OTA}}^{-1}(\cdot) : \mathbb{C}^{N_R \times 1} \times \mathbb{C}^{N_R \times N_T} \rightarrow \mathbb{R}^{N_U}$,

$$\hat{\mathbf{u}}_{sum} = C_{\theta^{OTA}}^{-1}(\mathbf{y}_{sum}, \mathbf{H}_{sum}), \quad (10)$$

where $\mathbf{H}_{sum} = \sum_{k=1}^K \mathbf{H}_k$ is the sum of the channel gain. After recovering the completed feature over τ timeslots, the

edge server only needs to perform a simple post-processing to obtain the aggregated feature,

$$\hat{\mathbf{f}}_{avg} = \frac{\hat{\mathbf{f}}_{sum}}{K}. \quad (11)$$

Compared to the time-consuming procedure of the OMA scheme, which requires $\mathcal{O}(K)$ transmission resource blocks and $\mathcal{O}(K)$ channel decoding operations, the AirPooling scheme only involves $\mathcal{O}(1)$ transmission resource block and $\mathcal{O}(1)$ channel decoding operation, significantly improving the execution efficiency of the end-to-end model.

B. Partial Fine-Tuning for Deteriorating Channels

Different from the traditional learning-communication separate design, the proposed semantic communication system for multi-view recognition adopts a joint source-channel encoding approach for processing view information. Therefore, during practical deployment, the inference accuracy depends not only on the captured view information but also on the current channel conditions. Let $\mathbf{x}_k = \phi_z + \mathbf{n}'_k$, where ϕ_z is the centroid of class z , and $\mathbf{n}'_k \sim \mathcal{CN}(\mathbf{0}, \sigma_v^2 \mathbf{I}_{N_T})$ represents the view-specific noise. Then, the received aggregated signal is characterized by the following distribution:

$$\mathbf{y}_{sum} \sim \mathcal{CN}\left(\mathbf{0}, \left(K\|\phi_z\|^2 + KN_T\sigma_v^2 + \sigma^2\right) \mathbf{I}_{N_R}\right). \quad (12)$$

As shown in the above equation, increasing channel noise leads to a larger variance in the aggregated signal, resulting in a more dispersed signal distribution. Such dispersion hinders the ability of the subsequent neural network to accurately map the signal to the correct class, ultimately leading to a decline in prediction accuracy. Fig. 2 presents the visualization of final prediction results under different channel conditions during inference. It can be observed that as the channel conditions deteriorate, the predictions become more scattered and disorganized, with the boundaries between classes gradually becoming blurred. Since the neural network is unable to effectively distinguish between these classes, the accuracy decreases accordingly. Therefore, when the system detects that the channel conditions have deteriorated to a certain extent, it is necessary to fine-tune the original pre-trained model to ensure the performance of the real-time model.

However, transmitting all view data to the edge server for model updating raises privacy concerns, and full end-to-end fine-tuning consumes substantial computation resources on the device side. Since channel variations primarily impact the later layers of the model, a partial fine-tuning approach is proposed, utilizing the layer freezing technique to prevent the parameters of earlier layers (such as the feature extractor) from participating in gradient updates, thereby reducing computation overhead. Consider the end-to-end learning model consists of L layers, where the last l layers are frozen. The remaining unfrozen trainable layers are updated using a gradient descent algorithm, and the model update process can be formulated as follows,

$$\mathbf{w}'_i = \begin{cases} \mathbf{w}_i, & i > L - l, \\ \mathbf{w}_i - \gamma \frac{\partial \mathcal{J}_{CE}}{\partial \mathbf{w}_i}, & \text{otherwise}, \end{cases} \quad (13)$$

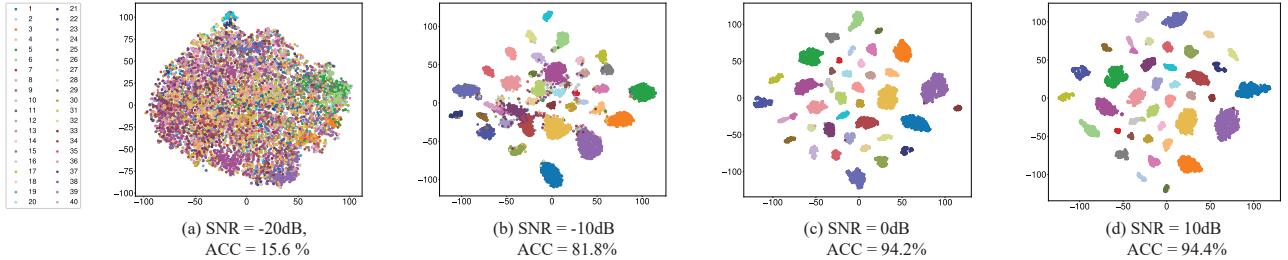


Fig. 2: The framework of collaborative semantic communication system.

where w_i is denoted as the model parameters of the i -th layer, and γ is the learning rate.

C. Implementation Issue of Partial Fine-tuning

As depicted in Fig. 1, by utilizing AirPooling and layer freezing, we propose a partial fine-tuning scheme for the AirPooling-based semantic communication system, which ensures stable high performance of the end-to-end model even under deteriorating channel conditions. Assuming that all the frozen layers are located on the model deployed at the sensing device, the detailed procedure is as follows.

(1) Device-side Model Forward Propagation: Using the local feature extractor and channel encoder, the sensing device encodes the image data captured from the specific angle into a single-view feature for transmission.

(2) Multi-view AirPooling: With the over-the-air computation technique, all sensors concurrently transmit encoded signals over a shared channel while accomplishing multi-view pooling.

(3) Server-side Model Forward Propagation: The edge server processes the received signals through the channel decoder and global classifier to derive the prediction.

(4) Server-side Model Back Propagation: Based on the predictions and corresponding labels, the loss is computed and used to update the server-side model parameters via gradient descent.

(5) Downlink Broadcasting of Received Signal's Gradients: When back propagation reaches the first layer of the channel decoder, the gradient of the received signal is broadcast to the sensing devices to enable the backward update of the device-side models.

(6) Device-side Model Forward Propagation: After receiving the gradient of the received signal, each sensing device updates its unfrozen network parameters using the gradient descent algorithm.

Although freezing certain layers may lead to a drop in accuracy, the proposed partial fine-tuning scheme offers advantages in terms of computational load and memory usage. Specially, assume frozen layers include L_ξ convolutional layers and L_ζ linear layers. Let C_i^I , C_i^O represent the input and output channels of the convolutional layers, G_i the kernel size, and W_i , H_i the width and height of the output feature map. Denote N_i^I and N_i^O as the number of input and output neurons for each linear layer. Then, the number of operations saved by

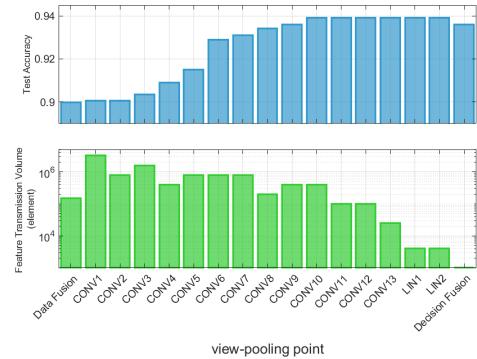


Fig. 3: System performance comparison with $SNR = -20dB$. All the values are normalized.

our partial fine-tuning method can be calculated as $FLOPs = \sum_{i=1}^{L_\xi} 2 \times C_i^I \times C_i^O \times W_i \times H_i \times G_i^2 + \sum_{i=1}^{L_\zeta} 2 \times N_i^I \times N_i^O$. Furthermore, freezing model parameters eliminates the need for devices to allocate memory for storing gradients and intermediate activations in the frozen layers, thereby reducing memory consumption.

The other factor affecting practical implementation is the view-pooling point in MVCNN. Figure 3 shows the test accuracy and the feature transmission volume by the edge device when selecting different view-pooling points, with VGG16 used as the standard CNN model. It can be observed that placing the view-pooling point in the later layers of VGG16 yields better recognition performance. This is because, when the view-pooling is located in deeper layers, each single-view feature can be extracted through more network layers, capturing more semantically rich information, thereby facilitating the identification of complementarities between multiple views. However, in the collaborative learning, the selection of the view-pooling point not only affects the amount of transmitted data but also introduces variations in the computation overhead on sensing devices. When view-pooling is performed at deeper layers, each sensing device is required to handle more computation, which should be taken into account during both model fine-tuning and inference stages.

IV. SIMULATION AND RESULTS

In the simulation, we consider a collaborative semantic communication system consisting of $K = 12$ sensing devices and an edge server, with the antenna size $N_R = N_T = 16$.

TABLE I: Fine-tuning performance with respect to the number of frozen layer l .

	Frozen Layers	Trained Parameters Size (MB)	Computation Overhead (GFLOPS)	Fine-Tuning Latency(s)	Downloading Overhead (Mbit)	Test Accuracy	
						$SNR = -20dB$	$SNR = -15dB$
Full fine-tuning	$l = 0$	139.77	147.00	226.00	3.21	0.906	0.932
	$l = 2$	139.73	143.09	216.52		0.890	0.929
	$l = 4$	139.51	143.09	210.80		0.886	0.927
Device-side Model Fine-tuning	$l = 6$	138.63	131.97	205.47	3.21	0.882	0.926
	$l = 8$	136.85	126.42	197.47		0.871	0.922
	$l = 10$	132.13	119.02	194.12		0.857	0.921
	$l = 12$	130.54	99.08	174.97		0.845	0.920
	$l = 0$	129.46	85.51	165.53		0.842	0.917
Device-side Model Frozen	$l = 16$	127.86	65.37	147.42		0.839	0.917
	$l = 18$	126.79	52.01	141.05	0	0.828	0.916
	$l = 20$	122.07	50.16	138.78		0.793	0.910
	$l = 22$	16.95	49.03	102.40		0.647	0.882

To implement multi-view learning, VGG16 is adopted as the standard CNN backbone, with the view-pooling point placed after the tenth convolutional layer (CONV10). We consider ModelNet40 [11] dataset for multi-view object recognition, where each of the 12 views per object is captured by a sensor. Both the channel encoder and decoder networks consist of four fully connected layers. Each MIMO channel is assumed to be Rayleigh fading, and the hardware noise factor is set as 6 dB. The pretrained model is trained with $SNR = KP_T/\sigma^2 = 5$ dB and a learning rate of 0.001. Fine-tuning is performed for 30 epochs with a learning rate set to 0.0001.

For performance comparison, we considered fine-tuning with OMA scheme and separate source-channel coding (SSCC) with AirPooling scheme. In the OMA scheme, all sub-channels are equally allocated to each sensing device, and thus the transmission latency of OMA can be expressed as $T^{OMA} = KN_F t_s / [B/\Delta f]$. And in the SSCC with AirPooling scheme, the beamforming design is derived using differential geometry to minimize the MSE [13].

As shown in Table I, we compare the performance of different fine-tuning schemes in terms of learning accuracy and resource overhead. As the number of frozen layers increases, the model's learning performance slightly declines, while the fine-tuning latency is correspondingly reduced. Compared to the full fine-tuning scheme, the partial fine-tuning scheme with $l = 10$ reduces the computational complexity by 19.03%. Even under extremely harsh channel conditions with $SNR = -20$ dB, this scheme achieves a test accuracy only 0.049 lower than that of the full fine-tuning. Furthermore, when $SNR = -15$ dB, the test accuracy remains above 0.90 even with up to $\lambda = 20$, at which point the entire model on the sensing device is frozen, and the transmission of aggregated signal's gradient can also be omitted.

In Fig. 4, we further investigate the learning performance versus the number of fine-tuning sensors under different fine-tuning schemes. As the number of fine-tuning sensing devices increases, the test accuracy of the fine-tuned model gradually improves. This is because the model can better learn to extract

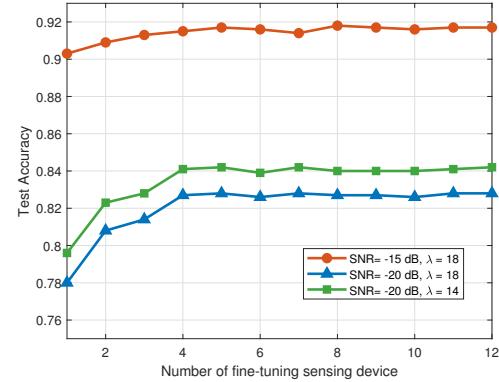


Fig. 4: System performance comparison with $SNR = -20dB$. All the values are normalized.

meaningful semantic information from more view features affected by the current channel noise. However, when the number of fine-tuning sensing devices reaches 4, the performance improvement tends to saturate, and adding more devices yields negligible gains in accuracy. Moreover, under more severe channel conditions, involving more devices in fine-tuning has a more pronounced impact on model performance. Therefore, in practical deployments, the selection of fine-tuning devices should take into account their energy efficiency to achieve a balance between learning performance and resource consumption.

Next, we compare the learning performance under different schemes in Fig. 5. It can be observed that the AirPooling-based scheme demonstrates stronger robustness to noise compared to the OMA-based scheme, maintaining high accuracy even when SNR drops to -5 dB. In contrast, the no fine-tuning with OMA scheme suffers a significant degradation in accuracy when the SNR falls below 0 dB. This is because, in the OMA-based scheme, each single-view feature is individually affected by channel noise during transmission and then sequentially processed by channel decoders to recover the original signal. Such multiple decoding stages can accumulate errors, leading

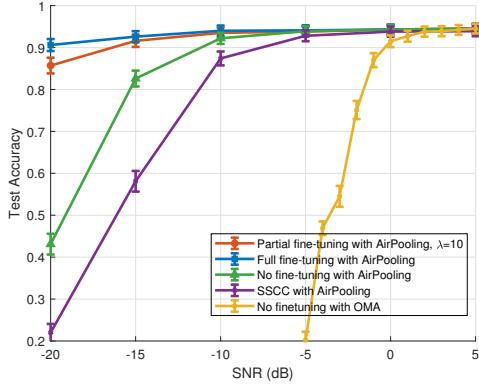


Fig. 5: System performance comparison with $SNR = -20dB$. All the values are normalized.

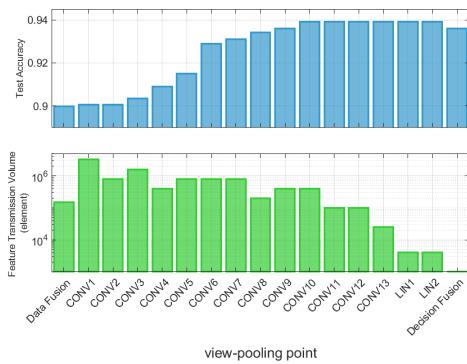


Fig. 6: System performance comparison with $SNR = -20dB$. All the values are normalized.

to increased deviation. In the low SNR regime, ($SNR \downarrow -10$ dB), the proposed partial fine-tuning scheme significantly outperforms both the no fine-tuning scheme and the SSCC with AirPooling scheme. This improvement stems from the proposed scheme's ability to perceive both data semantics and channel noise, enabling it to adjust network parameters according to the current channel conditions for optimal learning performance. In contrast, the no fine-tuning scheme lacks adaptability to worsening channel conditions, resulting in a sharp decline in test accuracy as the channel quality further deteriorates.

Fig. ?? shows the learning performance and resource consumption under different schemes. It can be observed that the proposed scheme achieves enhanced inference performance at the expense of additional fine-tuning overhead. Compared to SSCC with AirPooling scheme, our proposed partial fine-tuning scheme not only achieves lower test loss but also has a shorter model execution latency. This is attributable to the fact that the SSCC with AirPooling scheme involves lots of matrix inversion operations, which have higher computational complexity compared to simple linear neural networks. In addition, compared with the partial fine-tuning with OMA scheme, the partial fine-tuning with AirPooling scheme, although slightly less accurate, saves considerable feature trans-

mission and subsequent decoding latency due to its “compute-when-communicate” property.

V. CONCLUSION

In this work, we propose a channel-aware partial fine-tuning scheme for the AirPooling based semantic communication system to address the issues of aggregation latency and semantic distortion. With the AirPooling mechanism, the single-view encoded symbols from multiple sensors are concurrently transmitted to the edge server over a shared channel. And to ensure model performance over the varying channels, a partial fine-tuning scheme is designed by freezing layers less affected by channel variations to reduce fine-tuning overhead. Experiments show that the proposed scheme can ensure better model performance in the dynamic channel environment with lower communication and computation overhead.

ACKNOWLEDGMENT

This work was supported in part by the National Key R&D Program of China (No. 2021YFB3300100), and the National Natural Science Foundation of China (No. 62171062) and (U24A20234).

REFERENCES

- [1] X. Yan, S. Hu, Y. Mao, Y. Ye, and H. Yu, “Deep multi-view learning methods: A review,” *Neurocomputing*, vol. 448, pp. 106–129, 2021.
- [2] Z. Chen, J. Chen, W. Zhang, L. Guo, Y. Fang, Y. Huang, Y. Zhang, Y. Geng, J. Z. Pan, W. Song *et al.*, “Meaformer: Multi-modal entity alignment transformer for meta modality hybrid,” in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 3317–3327.
- [3] G. A. Khan, J. Khan, T. Anwar, Z. Ashraf, M. H. Javed, and B. Diallo, “Weighted concept factorization based incomplete multi-view clustering,” *IEEE Trans. Artif. Intell.*, 2024.
- [4] W. Yang, H. Du, Z. Q. Liew, W. Y. B. Lim, Z. Xiong, D. Niyato, X. Chi, X. Shen, and C. Miao, “Semantic communications for future internet: Fundamentals, applications, and challenges,” *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 213–250, 2022.
- [5] G. Zhang, Q. Hu, Z. Qin, Y. Cai, G. Yu, and X. Tao, “A unified multi-task semantic communication system for multimodal data,” *IEEE Trans. Commun.*, 2024.
- [6] Z. Z. J. Y. Fu, Yuechun and T. Q. Quek, “A semantic communication scheme for distributed holographic-type communications with multi-view images,” *IEEE Commun. Lett.*, 2024.
- [7] Y. Sun, S. Zhou, Z. Niu, and D. Gündüz, “Dynamic scheduling for over-the-air federated edge learning with energy constraints,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 227–242, 2021.
- [8] G. Zhu, Y. Wang, and K. Huang, “Broadband analog aggregation for low-latency federated edge learning,” *IEEE Trans. Wirel. Commun.*, vol. 19, no. 1, pp. 491–506, 2019.
- [9] Z. Liu, Q. Lan, A. E. Kalør, P. Popovski, and K. Huang, “Over-the-air multi-view pooling for distributed sensing,” *IEEE Trans. Wirel. Commun.*, 2023.
- [10] X. Chen, K. B. Letaief, and K. Huang, “On the view-and-channel aggregation gain in integrated sensing and edge ai,” *IEEE J. Sel. Areas Commun.*, 2024.
- [11] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.
- [12] J. Song, C. Häger, J. Schröder, T. J. O’Shea, E. Agrell, and H. Wyneersch, “Benchmarking and interpreting end-to-end learning of mimo and multi-user communication,” *IEEE Trans. Wirel. Commun.*, vol. 21, no. 9, pp. 7287–7298, 2022.
- [13] G. Zhu and K. Huang, “Mimo over-the-air computation for high-mobility multimodal sensing,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6089–6103, 2018.