

# Exploiting Large Language Models for Single-Channel Mix Source Separation in Anti-Jamming Wireless Communications

Boya Li<sup>ID</sup>, Luliang Jia<sup>ID</sup>, Lin Zhang<sup>ID</sup>, Nan Qi<sup>ID</sup>, Senior Member, IEEE, Wanbin Tang<sup>ID</sup>, and Fumiuki Adachi<sup>ID</sup>, Life Fellow, IEEE

**Abstract**—In anti-jamming wireless communications, single-channel mix source separation (SCMSS) is an effective way to combat full-band jamming. Conventional SCMSS methods typically depend on prior knowledge of jamming, and existing deep learning-based SCMSS methods require extensive training samples, limiting their applications in practical scenarios. Alternatively, we exploit large language models (LLMs) to design a novel SCMSS method, including a new LLM-based deep neural network (DNN), and a new fine-tuning algorithm. By harnessing the extremely powerful feature extraction and cross-domain knowledge transfer capabilities of LLMs, our method can effectively separate target signals from full-band jamming after fine-tuning with a small number of labeled data samples. Experimental results demonstrate superior bit error ratio (BER) and generalization performance over traditional and existing deep learning-based anti-jamming methods with significantly reduced training samples.

**Index Terms**—Single-channel mix source separation (SCMSS), anti-jamming, LLMs, fine-tuning.

## I. INTRODUCTION

IN MODERN wireless communication, anti-jamming technologies like direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS) are crucial for reliable transmission [1], [2]. However, their effectiveness is primarily limited to narrow-band jamming [3]. In adversarial environments such as military or congested networks, wide-band jamming severely degrades performance. To address this challenge, single-channel mix source separation (SCMSS) has emerged as a promising solution.

Traditional SCMSS methods primarily fall into two categories. The first is the virtual multi-channel approach [4], which transforms the single-channel signal into multiple virtual channels for separation using techniques like independent

Received 9 October 2025; accepted 22 November 2025. Date of publication 26 November 2025; date of current version 19 December 2025. This work was supported in part by National Natural Science Foundation of China under Grants 62571094, 62471491, 62271253 and National Key Laboratory of Wireless Communications Foundation under Grant IFN202411. The associate editor coordinating the review of this letter and approving it for publication was O. Simpson. (*Corresponding author:* Lin Zhang.)

Boya Li, Lin Zhang, and Wanbin Tang are with the National Key Laboratory of Wireless Communications, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: liboya.ee@std.uestc.edu.cn; linzhang1913@gmail.com; wbtang@uestc.edu.cn).

Luliang Jia is with the School of Space Information, Space Engineering University, Beijing 101416, China (e-mail: jialuts@163.com).

Nan Qi is with the Key Laboratory of Dynamic Cognitive System of Electromagnetic Spectrum Space, Ministry of Industry and Information Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: nanqi.common@gmail.com).

Fumiuki Adachi is with the International Research Institute of Disaster Science (IRIDeS), Tohoku University, Sendai 980-8577, Japan (e-mail: adachi@ceei.tohoku.ac.jp).

Digital Object Identifier 10.1109/LCOMM.2025.3637306

component analysis (ICA) [5]; its performance hinges on the independence of these virtual signals. The second category employs a finite symbol set, utilizing algorithms such as particle filtering (PF) [6] or per-survivor processing (PSP) [7] to estimate symbols and channel parameters. While the latter generally achieves better performance, it comes at the cost of complexity that grows exponentially with the signal modulation order.

Deep learning (DL) is applied to SCMSS to improve separation performance with reduced complexity. The core idea is to use large amounts of data samples and train a deep neural network (DNN) to capture distinct amplitude/phase features of target signals and jamming and learn a mixed signal-jamming waveform separation skill. Time-domain separation networks, such as Conv-TasNet [8], along with time-frequency domain separation networks such as DCCRN [9], have demonstrated that the well-trained DNN has an excellent mix source separation capability. Nevertheless, it is typically challenging to collect enough data samples for the training from scratch, especially in adversarial environments, constraining the practical applications of DL-based methods.

Recently, we noticed that the pre-trained large language models (LLMs) [10], [11] have excellent universal feature extraction and cross-domain adaptation capabilities even without task-specific information, and have been widely validated in the natural language processing-related field. In other words, if we embed the LLM into the SCMSS DNN, the LLM can extract correctly the features of target signals and jamming with a small amount of labeled data samples, thus largely reducing the required data samples compared with the existing DL-based SCMSS method. For this purpose, we design a novel LLM-based SCMSS DNN as well as a new fine-tuning algorithm in this letter, and demonstrate the advantages through experiments. To our best knowledge, this is the first attempt to exploit the LLM for the SCMSS.

## II. PROBLEM DESCRIPTION

Suppose there is a wireless transceiver and a jammer. Once the wireless transceiver link is active, the jammer can emit a full-band jamming to corrupt the link. Thus, the received mixed signal-jamming waveform  $\mathbf{x}(t)$  at the receiver can be expressed in the baseband equivalent representation as

$$\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{j}(t) + \mathbf{n}(t), \quad (1)$$

where  $\mathbf{s}(t)$ ,  $\mathbf{j}(t)$ , and  $\mathbf{n}(t)$  denote the signal, jamming, and noise components, respectively. In practice, the continuous waveform  $\mathbf{x}(t)$  is converted into a discrete-time sequence  $\mathbf{x} \in$

$\mathbb{C}^T$  through A/D sampling.  $s$  and  $j$  denote the discrete-time sequences of the  $s(t)$  and  $j(t)$ , respectively. In general, the receiver may fail to recover  $s$  due to the jamming  $j$ . Our goal is to develop a smart algorithm for the receiver to recover  $s$  from  $x$ , which can be treated as a typical SCMSS problem, i.e., separating  $s$  and  $j$  from  $x$ . Here, we intend to exploit LLM to construct a novel SCMSS DNN as well as a new fine-tuning algorithm, such that a small number of labeled data samples are enough for the SCMSS DNN to learn a good mix source separation skill from the mixed signal.

### III. LLM-BASED SCMSS ALGORITHM

In this part, we will first introduce the principle of the algorithm design, and then elaborate the details.

#### A. Principle of the Algorithm Design

Existing SCMSS DNN typically cascades an encoder module, a separation module, and a decoder module. In particular, the separation module is the core, and is responsible for extracting feature differences of the target signal and jamming, and then achieving separation; the encoder (decoder) is designed for data pre (post) processing before (after) the separation module.

In general, large amounts of labeled data samples are required to train the SCMSS DNN from scratch. Here, our idea is to introduce the LLM, which has excellent cross-domain knowledge transfer and universal feature extraction capabilities, into the SCMSS DNN to extract the multi-dimensional feature differences of the target signal and jamming. Then, it is possible that the LLM-based SCMSS DNN can be well-trained with a small number of labeled data samples. Therefore, the proposed LLM-based SCMSS DNN generally has four modules as shown in Fig. 1(a), including an encoder module, an LLM module, a separation module and a decoder module. In the following, we will elaborate on the designs of these modules.

#### B. Encoder Module

To learn and exploit the correlations between real and imaginary components at each time step, the real and imaginary parts of the input signal  $x$  are stacked in channel dimension to form a two-dimensional vector  $\tilde{x} \in \mathbb{R}^{2 \times T}$ . It is worth noting that the dimension of  $\tilde{x}$  is too low to fully trigger the excellent feature extraction capability of the LLM. Alternatively, we project  $\tilde{x}$  into a high-dimensional feature tensor before passing it to the LLM. In particular, we adopt a multi-channel convolutional layer with learnable hyper-parameters to achieve the projection. Here we do not consider complex-valued neural networks and keep the formulation real-valued for simplicity. Mathematically, the output feature tensor of the encoder can be expressed as

$$\mathbf{E} = h_{encoder}(\tilde{x}) = conv1d(\tilde{x}), \quad (2)$$

where  $conv1d(\cdot)$  denotes a 1-D convolution. It is worth noting that, although we feed the real and imaginary parts of the signal into the network for simplicity, the network can be redesigned to take the complex-valued signal as input [12].

#### C. LLM Module

Here, the open-source Generative Pre-trained Transformer 2 (GPT-2) is adopted as the LLM backbone for good balance between performance and size [13]. The backbone of GPT-2 consists of learnable positional embedding layers and stacked transformer decoders, where the number of stacks and feature sizes can be flexibly adapted. As in Fig. 1(a),  $N_L$  represents the stack number. Each layer consists of a multi-head self-attention layer, a feed-forward layer, and a normalization layer. Correspondingly, the extracted features from  $\mathbf{E}$  by the GPT-2 can be expressed by

$$\mathbf{G} = h_{GPT-2}(\mathbf{E}), \quad (3)$$

where  $h_{GPT-2}(\cdot)$  denotes the backbone network of the GPT-2.

#### D. Separation Module

The separation module is responsible for achieving a two-stage signal-jamming feature separation. In the first stage, the separation module learns a feature separation policy in terms of the encoder output feature tensor based on the data from the LLM. In the second stage, the separation module multiplies the feature separation policy by the encoder output feature tensor. Here, the feature separation policy can be treated as a time-domain mask of the encoder output.

1) *Feature Separation Policy Learning Stage*: We notice that the temporal convolution network (TCN) [14] has a multi-scale receptive field and contributes to a good feature separation performance in audio signal processing. Motivated by this, we integrate the TCN into the first stage to achieve the signal and jamming feature separation learning.

The TCN is established by cascading  $R$  TCN blocks, each of which contains  $M$  cascaded convolutional units, namely the conv unit, as shown in Fig. 1 (B), where  $B$  and  $H$  represent the different number of channel, and  $L$  represents the length of feature in each channel. Each unit consists of a conv1D operation followed by a depth-wise convolution (D-conv1D) operation, with nonlinear activation function and normalization added between each two convolution operations. Therefore, any conv unit can be referred to as  $conv(i, j)$ , satisfying  $1 \leq i \leq R$  and  $1 \leq j \leq M$ . In particular, the dilation factor of  $conv(i, j)$  is  $2^{i-1}$ , which ensures an extended temporal receptive field to effectively capture long-range signal dependencies [14].

Since the TCN is a stacked convolution network in nature, the computational complexity of the TCN increases exponentially with the input channel number. Then, a bottleneck layer, namely, the bottleneck-D, is adopted to decrease the channel number of the LLM output to  $B$  before it enters the TCN, to reduce the computational complexity of the TCN. Correspondingly, a bottleneck layer, namely bottleneck-I, is adopted to increase the TCN output channel number to  $H$  and recover the full feature of signals and jamming. Furthermore, a Parametric Rectified Linear Unit (PReLU) layer and a sigmoid layer are introduced into the network to enhance its nonlinear representation capacity. Mathematically, the feature separation policy is formulated as

$$\mathbf{M} = h_{sep}(\mathbf{G}), \quad (4)$$

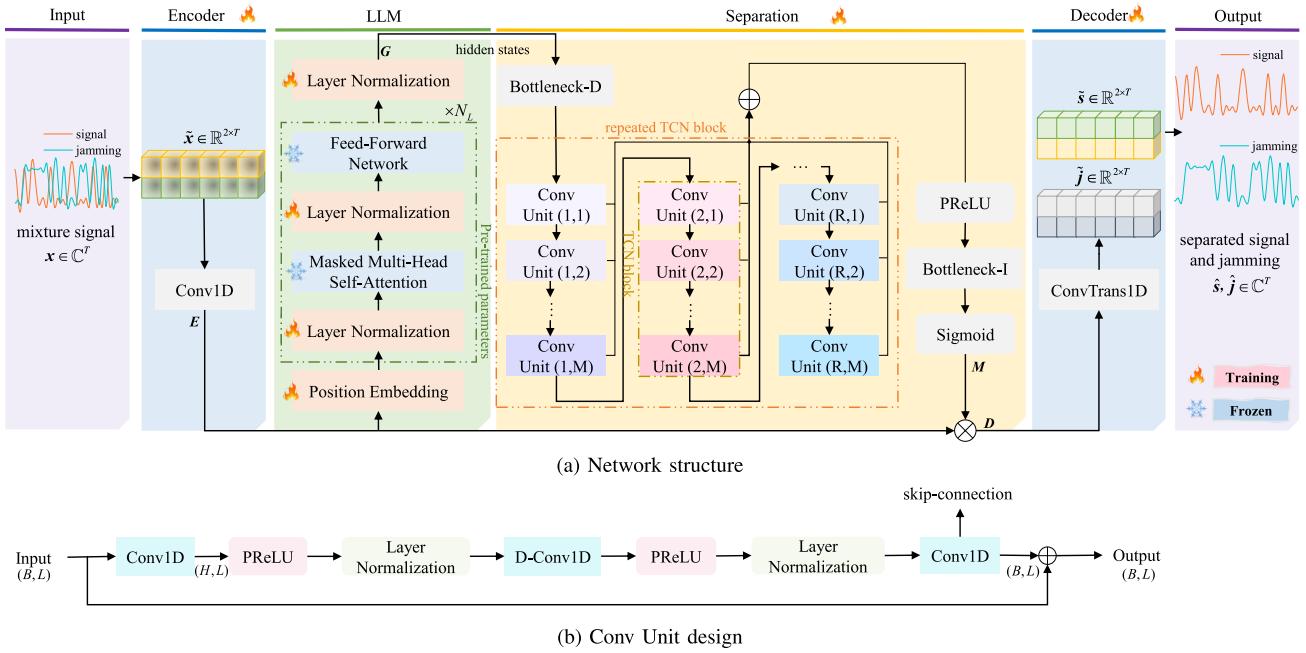


Fig. 1. (a) The block diagram of the proposed network structure. (b) The design of Conv unit.

where  $h_{sep}(\cdot)$  is the separation module.

2) *Feature Separation Stage*: The learned policy  $M$  is multiplied by the encoder output feature tensor via a skip-connection to disentangle signals and jamming. Accordingly, the feature separation result can be written as

$$\mathbf{D} = \mathbf{E} \odot M, \quad (5)$$

where  $\odot$  denotes the element-wise multiplication.

#### E. Decoder Module

The decoder is responsible for transforming the separated features of both the signals and jamming into the corresponding I/Q representations, i.e.,

$$[\tilde{s}, \tilde{j}] = h_{decoder}(\mathbf{D}) = convTrans1d(\mathbf{D}). \quad (6)$$

where  $convTrans1d(\cdot)$  is a one-dimensional transpose convolution, which is widely used to recover low-dimensional time-domain signals from high-dimensional features. Moreover,  $\tilde{s} \in \mathbb{R}^{2 \times T}$  and  $\tilde{j} \in \mathbb{R}^{2 \times T}$ , respectively, denote the separated source signal and jamming. Finally, we recover signal and jamming by combining the real and imaginary parts into complex-value representations, i.e.,  $\hat{s} \in \mathbb{C}^T$  and  $\hat{j} \in \mathbb{C}^T$ .

#### F. Fine-Tuning Method

According to [15], adjusting only the normalization and positional embedding layers in the LLM can achieve near-optimal performance compared with training from scratch. To further reduce training cost and enable cross-domain transfer, we freeze the multi-head attention and feed-forward layers and fine-tune the remaining layers.

In particular, we adopt the mean squared error (MSE) loss function for the fine-tuning, i.e.,

$$MSE = (s - \tilde{s})^2 + (j - \tilde{j})^2. \quad (7)$$

TABLE I  
HYPER-PARAMETERS OF THE PROPOSED METHOD

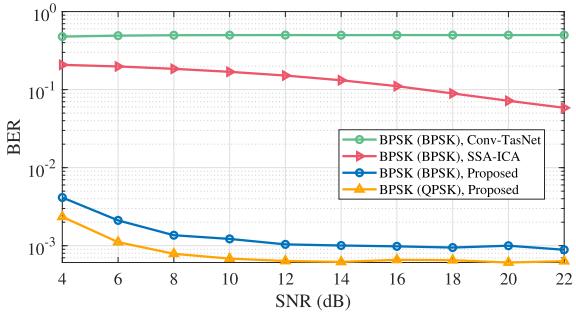
Symbol	Description	value
N	Number of channels in encoder	768
K	Kernel size in encoder	64
B	Number of channels in bottleneck and the skip-connection path's conv units	192
H	Number of channels in conv units	768
P	Kernel size in conv units	3
M	Number of conv units in each block	8
R	Number of TCN blocks	3
G	Number of GPT-2 layers	6

## IV. EXPERIMENTS

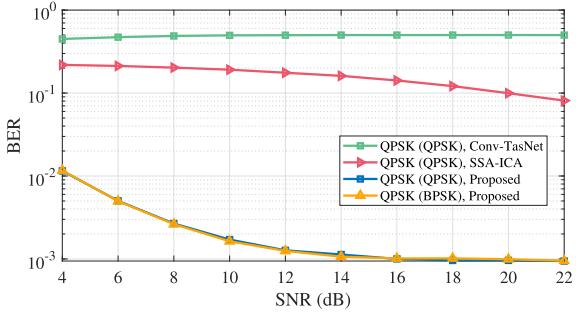
### A. Datasets

To evaluate the effect of the modulation type on the algorithm performance, we construct respectively a BPSK and a QPSK modulated signal-jamming-noise mixed dataset, in each of which the signal is BPSK (QPSK) modulated and the jamming can be either BPSK or QPSK modulated. In the BPSK (QPSK) modulated dataset, signal-to-noise ratio (SNR) ranges from 4 dB to 22 dB with 2 dB interval, and the signal-to-jammer ratio (SJR) is fixed at -6 dB, which maximizes the minimum Euclidean distance of constellation points and can be easily generalized to other SJR values in experiments.

For each SNR, there are 400 data samples. Each sample consists of 8192 BPSK (QPSK) modulated symbols mixed with BPSK (QPSK) jamming symbols under 4× up-sampling. The label of each sample is the expected ground-truth signal and jamming components. Here, we consider perfect synchronization of the signal at the receiver, which can be achieved with proper synchronization sequences. Meanwhile, we take different symbol offsets between the signal and jamming into



(a) The BER of BPSK modulated signal with BPSK or QPSK modulated jamming



(b) The BER of QPSK modulated signal with BPSK or QPSK modulated jamming

Fig. 2. The BER comparison of SSA-ICA, Conv-TasNet, and the proposed method under different SNRs.

account. Datasets are split 75% for training and 25% for validation. Testing datasets are generated at a specific SNR for inference.

### B. Experimental Configuration

Experiments were conducted on an NVIDIA GeForce RTX 4080 SUPER GPU and an Intel Core i7-13700KF CPU using PyTorch 2.4.0. We used the Adam optimizer with a  $10^{-3}$  initial learning rate, halved if the validation MSE plateaued for two epochs.

### C. Experimental Results and Discussions

1) *Separation Performance Comparison*: Fig. 2 illustrates the BER performance of different SCMSS algorithms with different SNRs when the training sample size is 10. Algorithms include the proposed method, the traditional method of singular spectrum analysis-independent component analysis (SSA-ICA) [5] and DL-based Conv-TasNet [8]. The hyper-parameters in Conv-TasNet are the same as those of the proposed method as shown in Tab. I. The BER performance of QPSK modulated signal and BPSK modulated signal are shown in Fig. 2. In Fig. 2(a), the BER of SSA-ICA slowly decreases as the SNR increases from 4 to 22 dB and demonstrates a limited capability to suppress full-band jamming. This is because its linear modeling framework cannot adequately characterize the non-linear distortion mechanisms intrinsic to full-band jamming. In addition, the BER of Conv-TasNet remains almost 0.5. This is because a small number of training samples, e.g., 10, lead to a poor representation capability

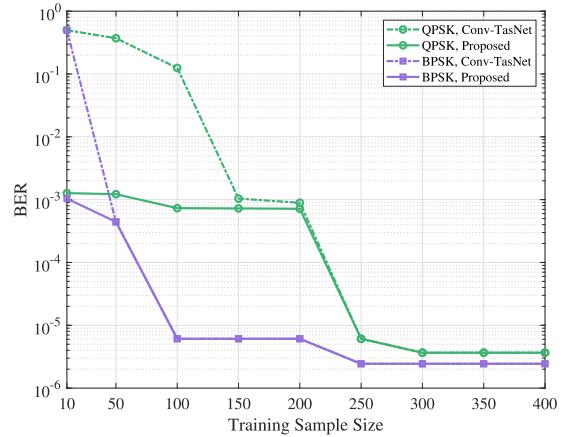


Fig. 3. The BER comparison of SSA-ICA, Conv-TasNet, and our proposed method under different training sample sizes at SNR=12 dB.

TABLE II  
PERFORMANCE COMPARISON IN ABLATION EXPERIMENTS

Metric	proposed	w/o LLM	w/o TCN
BER	<b>0.00193</b>	0.371	0.0026

of the Conv-TasNet. However, the BER of the proposed method decays quickly from around  $0.04$  to  $9 \times 10^{-4}$  as the SNR increases from 4 to 12 dB and remains constant as the SNR increases further. This is reasonable since the pre-trained LLM, with strong feature extraction and cross-domain transfer capabilities, can still capture sufficient features even with limited training samples. Besides, we observe that the BER of the proposed algorithm for distinct modulations of the signal and jamming is lower than that for same modulations of the signal and jamming, as shown in both Fig. 2(a) and Fig. 2(b). By further comparing Fig. 2(a) and Fig. 2(b), we notice that it is harder for the proposed algorithm to separate QPSK component than BPSK component due to the denser constellation and smaller Euclidean distance.

The BER performance of the proposed method and Conv-TasNet under different sample sizes is shown in Fig. 3. In general, the proposed method achieves lower BER with few training samples, while performing comparably to Conv-TasNet with larger datasets. For QPSK, achieving a BER of  $10^{-2}$  requires only 10 samples for the proposed method versus about 140 for Conv-TasNet. For BPSK, fewer than 10 samples are needed compared to about 30 for Conv-TasNet. This indicates that the proposed method can largely reduce the required number of training samples and is especially suitable to the adversarial environment in which the training samples are scarce.

2) *Ablation Experiments*: Ablation studies, shown in Table II, validate the effectiveness of the LLM module and dilated TCN blocks by removing relevant components. BER values are averaged across SNRs from 4 to 22 dB. For the model without LLM and TCN blocks, BER values are approximately 0.371 and 0.0026, respectively, demonstrating that LLM is crucial for its powerful feature extraction capability.

3) *Symbol Offsets Analysis*: Fig. 4 illustrates the BER performance of the proposed method with different symbol offsets

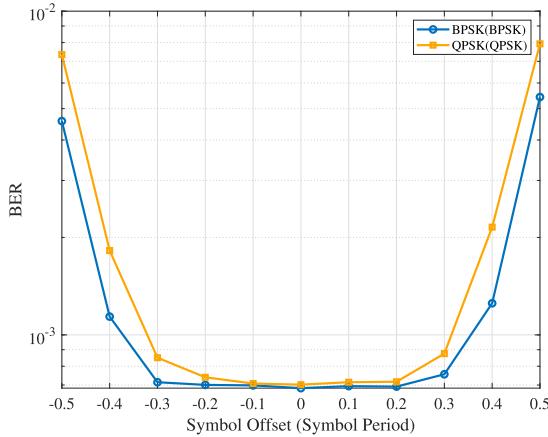


Fig. 4. The BER performance of the proposed method under different symbol offsets between the signal and jamming at  $\text{SNR} = 12 \text{ dB}$ .

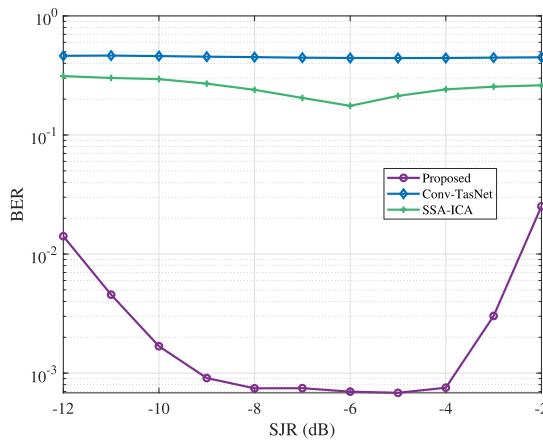


Fig. 5. The generalization behaviors of different methods at  $\text{SNR} = 12 \text{ dB}$ .

between the signal and jamming when the training sample size is 100. Taking the QPSK modulated signal and jamming as an example, BER is generally lower than 0.01 when the symbol offset varies between  $-0.5$  and  $0.5$  symbol period, with a ‘U’ shape curve, achieving the minimum at zero offset. This figure also verifies that the well-trained DNN model can accommodate the interference with different symbol offsets. The reason lies in the designed DNN with a flexible temporal receptive field.

4) *Generalization Analysis:* Fig. 5 compares the generalization of Conv-TasNet and the proposed LLM-based method with 100 training samples. The DNN is trained only on the QPSK dataset at  $\text{SJR} = -6 \text{ dB}$  and tested at  $\text{SJR}$  between  $-12 \text{ dB}$  and  $-2 \text{ dB}$ . For comparison, SSA-ICA is also included. The figure shows that the proposed method generalizes better than SSA-ICA and Conv-TasNet, which fails with 100 samples and yields a constant BER of 0.5. After fine-tuning, the proposed method achieves a BER below 0.01 when the jamming is neither too strong ( $> -12 \text{ dB}$ ) nor too weak

( $< -2 \text{ dB}$ ). Extending LLM-based multi-channel separation to handle extreme SJR cases will be our future work.

## V. CONCLUSION

In this letter, we designed a novel LLM-based SCMSS DNN as well as a new fine-tuning algorithm. By leveraging the powerful feature extraction and the cross-domain knowledge transfer capabilities of the LLM, the proposed method can achieve better SCMSS performance than the conventional SCMSS algorithm and the existing DL-based SCMSS method. Specially, it is demonstrated that the LLM module is able to largely reduce the required training sample size, making the proposed method quite suitable to the adversarial environment in which the training samples are scarce.

## REFERENCES

- [1] L. Jia et al., “Game theory and reinforcement learning for anti-jamming defense in wireless communications: Current research, challenges, and solutions,” *IEEE Commun. Surveys Tuts.*, vol. 27, no. 3, pp. 1798–1838, Jun. 2025.
- [2] X. Lin et al., “Secure beamforming and anti-jamming coalition formation for air-terrestrial integrated ad-hoc networks,” *IEEE Trans. Wireless Commun.*, vol. 24, no. 8, pp. 6722–6736, Aug. 2025.
- [3] C. Han, K. An, Z. Lin, S. Chatzinotas, and J. Wang, “Endogenous anti-jamming communications for SAGIN: A network perspective,” *IEEE Netw.*, early access, Mar. 2025, doi: [10.1109/MNET.2025.3551248](https://doi.org/10.1109/MNET.2025.3551248).
- [4] P. He, T. She, W. Li, and W. Yuan, “Single channel blind source separation on the instantaneous mixed signal of multiple dynamic sources,” *Mech. Syst. Signal Process.*, vol. 113, pp. 22–35, Dec. 2018.
- [5] Z. Wu, D. Yuan, F. Zhang, and M. Yao, “Improving accuracy of low-cost vehicle attitude estimation by denoising MIMU based on SSA and ICA,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.
- [6] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, Jun. 1999.
- [7] R. Raheli, A. Polydoros, and C.-K. Tzou, “Per-survivor processing: A general approach to MLSE in uncertain environments,” *IEEE Trans. Commun.*, vol. 43, nos. 2–4, pp. 354–364, Feb. 1995.
- [8] Y. Lue et al., “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.
- [9] Y. Hu et al., “DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2472–2476.
- [10] B. Liu, X. Liu, S. Gao, X. Cheng, and L. Yang, “LLM4CP: Adapting large language models for channel prediction,” *J. Commun. Inf. Netw.*, vol. 9, no. 2, pp. 113–125, Jun. 2024.
- [11] S. Chen et al., “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.
- [12] P. Guo, M. Yu, L. Shen, Z. Lin, K. An, and J. Wang, “Single-channel blind source separation in wireless communications: A complex-domain deep learning approach,” *IEEE Wireless Commun. Lett.*, vol. 13, no. 6, pp. 1645–1649, Jun. 2024.
- [13] A. Radford et al., “Language models are unsupervised multitask learners,” in *OpenAI Blog*, vol. 1, no. 8, p. 9, Apr. 2019.
- [14] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 156–165.
- [15] J. E. Hu et al., “LoRA: Low-rank adaptation of large language models,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2021, pp. 1–20.