CVCAM-K: A Specific Emitter Identification Method with Complex-Valued Convolutional Attention Mamba and KNN

1st Jiajun Yu
Information Engineering University
Zhengzhou, China
yjj020415@163.com

2nd Hao Zhang
Information Engineering University
Zhengzhou, China
haozhang0126@163.com

3rd Dan Qu*
Information Engineering University
Zhengzhou, China
qudan_xd@163.com

***Abstract*—Specific emitter identification (SEI) aims to achieve unique identification of emitters through subtle signal characteristics. Recently, Mamba-based methods have gained attention for selective state-space modeling, which achieves linear complexity—improving efficiency over Transformer's square complexity in long signal processing. However, Mamba primarily models temporal correlations and lacks cross-dimensional selectivity for multi-channel features from Complex-Valued Neural Network (CVNN), limiting feature representation. Moreover, existing methods mostly use trained parametric models for classification, ignoring intrinsic data correlations. To address these issues, this paper proposes the CVCAM-K method. The method constructs the Complex-Valued Convolutional Attention Mamba (CVCAM) model, introducing the Convolutional Block Attention Module (CBAM) into CVNN to enhance the signals fed to Mamba. Further, the method innovatively integrates the K-nearest neighbor (KNN) algorithm to make decisions based on both parametric models and data correlations. Experiments on an open-source automatic dependent surveillance-broadcast (ADS-B) dataset for 10-class recognition show that our method achieves 85.3% average accuracy using only 10% of training set, outperforming mainstream approaches. With full training set, accuracy reaches 99.3%, matching the best benchmarks.**

***Index Terms*—Automatic dependent surveillance-broadcast (ADS-B), complex-valued convolutional attention Mamba (CVCAM), K-nearest neighbor (KNN), specific emitter identification (SEI)**

I. Introduction

Specific emitter identification (SEI) enables device authentication by extracting physical layer features from signals [1], which demonstrates promising applications in both civilian and military domains [2], [3]. The recognition performance of SEI critically depends on feature extraction. Traditional SEI research often relies on domain knowledge, resulting in issues such as limited features and poor generalization capabilities [4]. Deep learning (DL) has driven advancements in signal processing [5]. In 2018, Merchant K et al. [6] demonstrated the feasibility of DL methods for the SEI field by extracting features directly from I/Q sequences using convolutional neural network (CNN).

Currently, Transformer-based SEI has been widely applied. Huang et al. [7] first applied Transformer to channel-robust SEI, verifying its strong robustness to channel variations. Deng et al. [8] proposed a lightweight Transformer, which reduced parameter count while improving recognition accuracy. Wang et al. [9] enhanced local and global feature modeling through a CNN-Transformer architecture, thereby advancing SEI performance. However, the square complexity of Transformer imposes an efficiency bottleneck when processing long signals. In recent years, Mamba [10], based on state space model (SSM), has achieved sequence modeling capabilities comparable to Transformer with linear complexity, finding widespread application across multiple fields [11], [12], [13], [14]. Liu et al. [15] first applied Mamba for SEI by designing the complex-valued convolutional Mamba (CVCM). CVCM achieved an accuracy of 96.58% on the 30-class automatic dependent surveillance-broadcast (ADS-B) dataset [16], demonstrating its effectiveness.

However, CVCM has some limitations. Mamba's selective mechanism primarily addresses the time dimension and struggles to optimize information across different channels after complex-valued convolutions, resulting in insufficient feature representation. Furthermore, similar to most DL methods, its parameterized decision mechanism solidifies boundaries while ignoring data correlations, leading to degraded generalization performance under limited data conditions.

Aiming at these problems, we propose a SEI method with complex-valued convolutional attention Mamba (CVCAM) and K-nearest neighbor (KNN) algorithm, named CVCAM-K. CVCAM-K makes two key improvements based on CVCM. First, it introduces the convolutional block attention module (CBAM) [17] to enhance the key features. Second, it integrates KNN [18] to construct a decision-making framework that combines parameterized models with data correlations. We evaluated the proposed method on an open-source real-world ADS-B dataset. In the 10-class SEI task, CVCAM-K achieved an accuracy of 85.3% with only 10% of the training set, showing a clear improvement in average accuracy over the compared baseline methods. With the full training set, accuracy reached 99.3%. These results demonstrate the effectiveness of the proposed CVCAM-K method on the evaluated dataset.

II. Methods

A. Proposed CVCAM-K

Fig. 1 shows the overall process of CVCAM-K, which primarily consists of two stages: training and inference. The training stage aims to optimize model parameters and construct the feature database. After the raw signal is fed into CVCAM, features are first extracted via the complex-valued convolutional attention network (CVCAN), followed by sequence modeling using Mamba. Finally, global feature vectors are obtained through attention pooling layer, which are

*Dan Qu is the corresponding author.

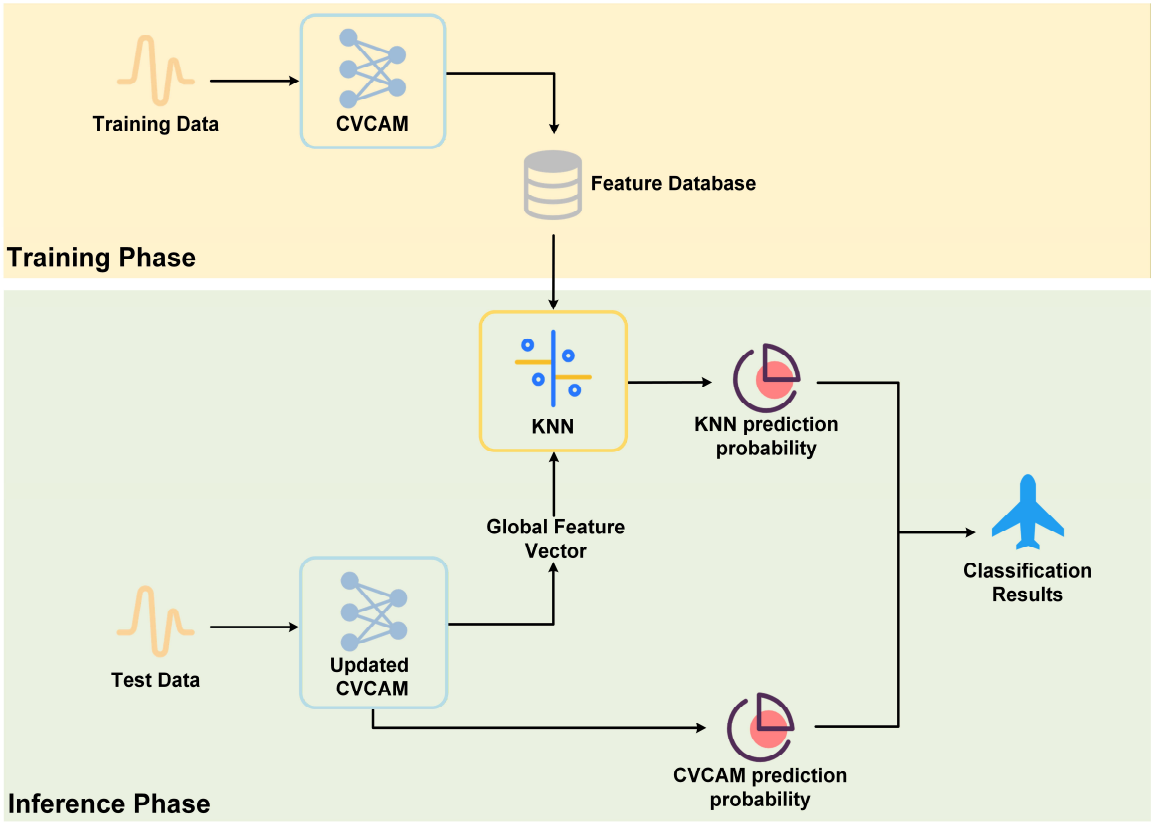


Fig. 1. Overall Process of CVCAM-K. It consists of two stages: training and inference.

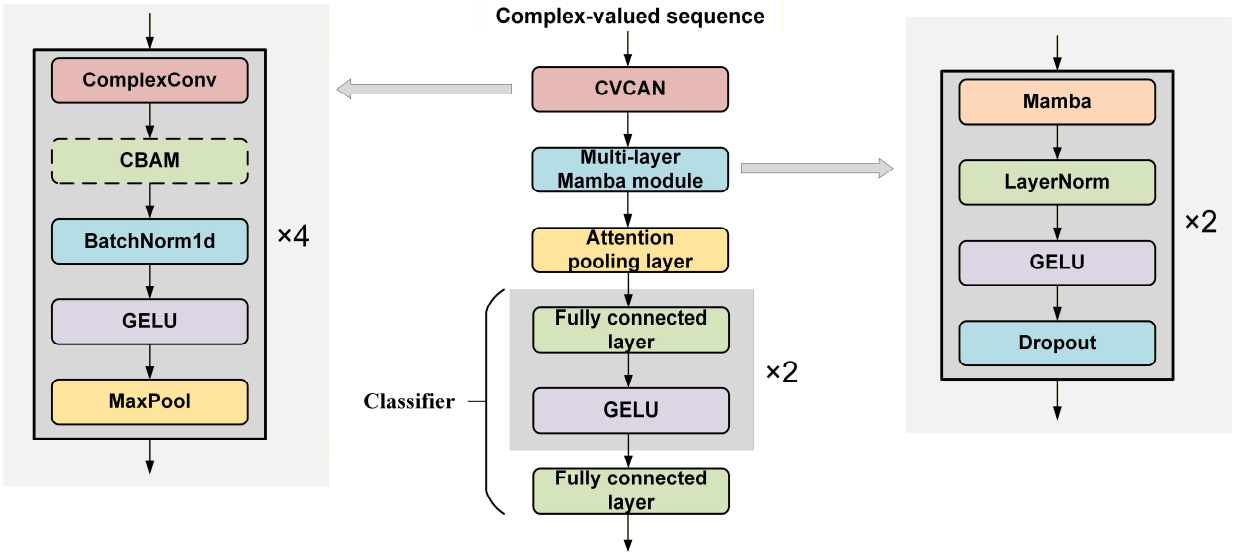


Fig. 2. Architecture of CVCAM. The diagrams on the left and right show the detailed structures of CVCAN and the multi-layer Mamba module respectively.

used to update parameters for training and are stored in the feature database alongside corresponding labels. The inference phase employs a dual-path hybrid decision mechanism. The CVCAM path produces a parameterized probability, while the KNN path generates a non-parameterized probability based on data correlations. These probabilities are then weighted and fused to produce a composite probability, with the highest value indicating the recognition result.

B. Overall Structure of CVCAM

As shown in Fig. 2, CVCAM comprises four key components: the CVCAN, a multi-layer Mamba module, an attention pooling layer, and a fully-connected classifier.

1) CVCAN

For complex-valued signals, we design CVCAN as the core feature extractor, which consists of four complex convolutional blocks.

After the complex-valued signal $\mathbf{X}$ is fed into CVCAN, it first passes through the complex-valued convolution layer, which transforms $\mathbf{X}$ into the output feature map $\mathbf{F}$.

To enhance the feature representation capability, we introduce CBAM after the first and third complex-valued convolutional layers. This approach enables higher computational efficiency while maintaining performance. As shown in Fig. 3, CBAM enhances discriminative features through channel attention module (CAM) and spatial attention module (SAM).

Specifically, upon receiving $\mathbf{F}$, CAM performs global average pooling (GAP) and global max pooling (GMP) on it to generate two distinct channel descriptor vectors, $\mathbf{z}_{\text{avg}}$ and $\mathbf{z}_{\max}$:

$$\mathbf{z}_{\text{avg}} = \text{AvgPool}(\mathbf{F}), \tag{1}$$

$$\mathbf{z}_{\max} = \text{MaxPool}(\mathbf{F}). \tag{2}$$

Subsequently, $\mathbf{z}_{\text{avg}}$ and $\mathbf{z}_{\max}$ are fed into a shared multilayer perceptron (MLP) for nonlinear transformation and fusion. After activation by the Sigmoid function, the channel weight vector $\mathbf{M}_{\text{c}}(\mathbf{F})$ is generated. The channel-refined feature map $\tilde{\mathbf{F}}_{\text{c}}$ is then obtained via element-wise multiplication:

$$\mathbf{M}_{\text{c}}(\mathbf{F}) = \sigma\big(\text{MLP}(\mathbf{z}_{\text{avg}}) + \text{MLP}(\mathbf{z}_{\max})\big), \tag{3}$$

$$\tilde{\mathbf{F}}_{\text{c}} = \mathbf{M}_{\text{c}}(\mathbf{F}) \otimes \mathbf{F}, \tag{4}$$

where, σ denotes the Sigmoid function, $\otimes$ denotes the element-wise multiplication operation.

SAM performs GAP and GMP on $\tilde{\mathbf{F}}_{\text{c}}$ along the channel dimension, generating two 2D feature maps, $\mathbf{f}_{\text{avg}}$ and $\mathbf{f}_{\max}$:

$$\mathbf{f}_{\text{avg}} = \text{AvgPool}_{\text{channel}}(\tilde{\mathbf{F}}_{\text{c}}), \tag{5}$$

$$\mathbf{f}_{\max} = \text{MaxPool}_{\text{channel}}(\tilde{\mathbf{F}}_{\text{c}}). \tag{6}$$

Concatenate $\mathbf{f}_{\text{avg}}$ and $\mathbf{f}_{\max}$ along the channel dimension, then apply the 1D convolution operation followed by Sigmoid function to generate the spatial weight matrix $\mathbf{M}_{\text{s}}(\tilde{\mathbf{F}}_{\text{c}})$:

$$\mathbf{M}_{\text{s}}(\tilde{\mathbf{F}}_{\text{c}}) = \sigma\big(f^{n\times1}([\boldsymbol{f}_{\text{avg}}; \boldsymbol{f}_{\max}])\big), \tag{7}$$

where, $f^{n\times1}$ denotes the 1D convolution operation with a kernel size of n. The final spatially refined feature map $\tilde{\mathbf{F}}_{\text{s}}$ is computed via element-wise multiplication and then normalized:

$$\tilde{\mathbf{F}}_{\text{s}} = \mathbf{M}_{\text{s}}(\tilde{\mathbf{F}}_{\text{c}}) \otimes \tilde{\mathbf{F}}_{\text{c}}. \tag{8}$$

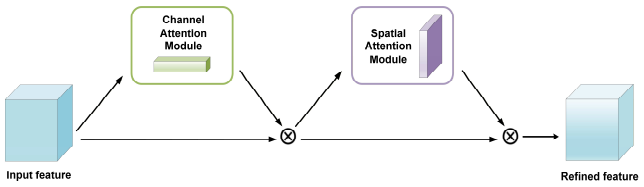


Fig. 3. Architecture of CBAM.

The refined features from CBAM are then nonlinearly transformed via the gaussian error linear unit (GELU) activation function and downsampled by a max-pooling layer. These high-dimensional features are subsequently fed into the multi-layer Mamba module.

2) Multi-layer Mamba module

We employ Mamba as the core component for sequence modeling and design the multi-layer Mamba module, which consists of Mamba, LayerNorm (layer normalization), GELU activation function, and Dropout layer. This module is stacked twice.

The high-dimensional features extracted by CVCAN are fed into the multi-layer Mamba module after dimension transposition. Mamba models sequential dependencies, followed by LayerNorm, GELU activation function, and Dropout layer to enhance representations and prevent overfitting. Finally, the multi-layer Mamba module produces advanced sequence features, which are then fed into the attention pooling layer and classifier.

3) Attention pooling layer and classifier

Advanced sequence features are aggregated by an attention pooling layer. This layer calculates weights for each time step through a learnable function and produces a fixed-dimensional global feature vector via a weighted sum. This vector is then fed into a three-layer fully connected classifier, which maps it to an output space with the same number of classes through nonlinear transformations. The output result is converted by the LogSoftmax function into the log probability $p_{CVCAM}(y_i|x)$. The class with the highest probability value is selected as the final recognition result.

C. Enhancing Inference with KNN

In DL-based SEI, the decision-making mechanism of parameterized models faces inherent limitations. When training data is scarce or the test environment changes, relying on fixed boundaries for decision-making leads to degraded generalization capabilities, which in turn causes unstable performance. To mitigate this, we introduce a KNN decision fusion mechanism which provides additional and effective information by referencing the labels of its nearest neighbors in the training set. Specifically, it dynamically generates a probability distribution based on the labels of the K-nearest neighbors by calculating the similarity between the test sample and samples in the feature database.

The KNN method consists of three steps: feature database construction, nearest neighbor search, and KNN distribution construction.

1) Feature database construction

Given the signal-class data pairs $(x, y^*) \in D$ from the training set, where x denotes the input signal and y^* denotes the class to which the signal belongs. The feature database is constructed as follows:

$$(K, V) = \bigcup_{(x,y^*)\in D} \{(f(x), y^*)\}, \tag{9}$$

where, the key K is a high-level representation of x obtained by processing it through the projection function $f(\cdot)$. In this paper, it refers to the output of the attention pooling layer. The value V denotes the category corresponding to x.

2) Nearest neighbor search

During the testing phase, given an input signal x, we use $f(x)$ as the query value and search for k nearest key-value pairs in the feature database constructed in the previous step based on the cosine distance d:

$$N = \{(d_j, v_j), j \in \{1,2,\ldots,k\} | (key_j, v_j) \in (K, V)\}, \tag{10}$$

TABLE I. COMPARISON OF ACCURACY AMONG DIFFERENT METHODS

Methods	Training Dataset Proportion				
	5%	10%	20%	50%	100%
CVNN[19]	61.20%	73.40%	92.50%	97.70%	99.20%
DRCN[20]	54.70%	74.90%	93.20%	97.20%	98.90%
SSRCNN[21]	49.70%	78.90%	91.40%	97.40%	99.10%
SimMIM[22]	66.30%	78.20%	92.70%	97.50%	98.90%
TripleGAN[23]	47.30%	64.30%	91.10%	97.20%	99.10%
MAT-PA[24]	73.70%	84.90%	93.60%	97.30%	99.30%
RC-SEI[25]	49.70%	63.50%	67.20%	93.40%	**99.40%**
CVCAM-K	**74.20%**	**85.30%**	**93.70%**	**98.20%**	99.30%

$$d_j = d\left(key_j, f(x)\right), \tag{11}$$

where, key_j denotes the j-th nearest neighbor representation retrieved from the feature database, v_j is the class corresponding to this representation, and $d(\cdot)$ denotes the cosine distance.

3) KNN distribution construction

We convert the cosine distances between the k nearest neighbors retrieved and the query sample into a categorical probability distribution, such that samples with closer distances receive higher probabilities:

$$p_{\mathrm{KNN}}(y_i|x) \propto \sum_{(d_j,v_j)\in N} \mathbb{I}_{y_i=v_j} \exp\left(\frac{-d_j}{\tau}\right), \tag{12}$$

where, τ is the temperature coefficient, and $\mathbb{I}$ is the indicator function. For the y_i -th class, the prediction probability $p_{\mathrm{KNN}}(y_i|x)$ of KNN is the sum of the probabilities for that class after applying the softmax function.

Although KNN alone can perform recognition, its effectiveness relies heavily on the quality of the data. To enhance recognition robustness, we combine KNN's probability distribution with CVCAM's probability distribution using a weight λ. The final probability that x belongs to y_i-th class is:

$$p(y_i|x) = \lambda p_{\mathrm{CVCAM}}(y_i|x) + (1-\lambda) p_{\mathrm{KNN}}(y_i|x). \tag{13}$$

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Data

ADS-B is a technology that broadcasts dynamic flight information of aircraft in real time, carrying the unique identification information of the aircraft (ICAO code). The ADS-B dataset is widely used due to its large scale and open-source nature. Our experiment uses ADS-B data comprising 10-class aircraft signals, with 3,081 signals in the training set and 1,000 signals in the test set. Each signal is 4,800 sampling points in length. To evaluate model performance under varying data availability, we train our model using five different subsets of the training data: 5%, 10%, 20%, 50%, and 100% of the total training samples. Additionally, 20% of the training data is selected to form the validation set. All data is randomly selected.

B. Settings

In the experiment, the operating system is Ubuntu 18.04 LTS, and GPU is NVIDIA Tesla V100. We implement our approach in PyTorch (v2.4.0 with Python 3.9) by optimizing the parameters using Adam. The loss function selected is the cross-entropy loss function. We train the model for 300 iterations and the batch size is 32. Learning rate is 0.001. Dropout rate is 0.3. For Mamba, the model dimension is 512, the SSM state dimension is 32, the local convolution width is 4, and the expansion factor is 2. For KNN, we set τ=1.0, k=5 and λ=0.8. These parameters were selected on the validation set.

C. Results and analysis

The CVCAM-K method is compared with seven latest SEI methods, including CVNN [19], DRCN [20], SSRCNN [21], SimMIM [22], TripleGAN [23], MAT-PA [24], and RC-SEI [25]. For fairness, all methods were trained and tested under identical conditions of data, iteration, batch size, and learning rate. The average accuracy obtained by averaging the results from multiple experiments is shown in TABLE I. The CVCAM-K method demonstrates outstanding performance across varying training data volumes. In the presence of limited data (5% and 10% training data), it achieves accuracies of 74.20% and 85.30%, outperforming other methods. As training data increases, its performance advantage remains substantial. Only when using the full dataset did it fall slightly below RC-SEI's 99.40%. This minor gap may stem from RC-SEI's design for complete datasets, which prioritizes achieving maximum accuracy under abundant data conditions.

D. Ablation Experiments

To investigate the contributions of the core components in our method, we designed ablation experiments focusing on analyzing the performance gains of CBAM and the effectiveness of the KNN decision fusion mechanism.

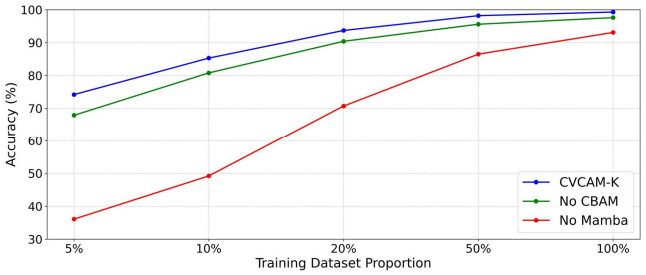
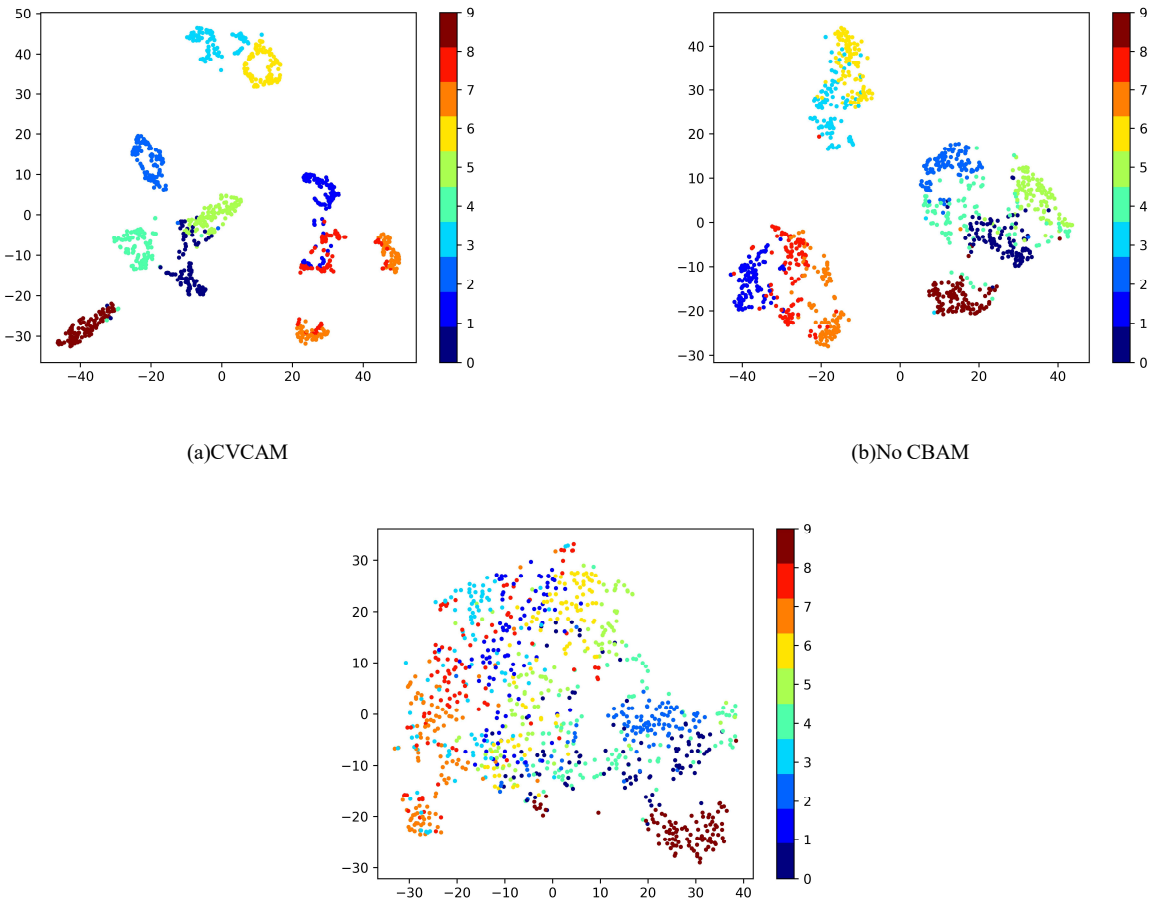


Fig. 4. Effect of three configurations on average accuracy.



(a)CVCAM

(b)No CBAM

(c)No Mamba

Fig. 5. Visualization of semantic features of three different configurations.

1) Performance Gains of CBAM

To evaluate the contribution of CBAM, we designed ablation experiments comparing the performance of three configurations: removing CBAM (No CBAM), removing multi-layer Mamba module (No Mamba), and complete CVCAM-K structure. All other settings remain the same. As shown in Fig. 4, CBAM brings significant improvements across different data scales. The performance gain of CBAM narrows gradually as the proportion of training data increases, but it still provides a decisive 1.70%-point accuracy lead over the model without CBAM when using the full dataset, pushing the final accuracy to 99.30%. These results further validate CVCAM's outstanding performance for SEI.

TABLE II.

SILHOUETTE COEFFICIENT AMONG THREE CONFIGURATIONS

Parameter	**Methods**		
	CVCAM	**No CBAM**	**No Mamba**
Silhouette Coefficient	**0.284**	0.203	0.025

To thoroughly analyze the quality of features learned by CVCAM, we employed t-distributed Stochastic Neighbor Embedding technique for dimensionality reduction visualization, as shown in Fig. 5. Additionally, we calculated the silhouette coefficient and present the results in TABLE II. We only show the results when the ratio is 10% because of the limited space. The results show that features extracted by complete CVCAM exhibit distinct intra-class clustering and inter-class separation structures in the 2D space. After removing CBAM, the inter-class boundaries become blurred, and the silhouette coefficient decreases accordingly. After removing multi-layer Mamba module, features failed to form effective class boundaries, with a silhouette coefficient of only 0.025. These results further confirm that CVCAM, which integrates Mamba and CBAM, can learn more discriminative feature representations, providing a geometric explanation for the outstanding recognition performance.

2) Effectiveness of the KNN Decision Fusion Mechanism

To systematically evaluate the effectiveness of the KNN fusion strategy, we designed ablation experiments, setting λ to 0.4, 0.6, 0.8, and 1.0, while keeping other parameters and experimental settings consistent. The experimental results are shown in Fig. 6. The highest recognition accuracy is achieved at λ =0.8. As λ increases or decreases, the recognition accuracy declines. When λ=1.0, the output entirely relies on CVCAM, ignoring data correlations; when λ =0.4, the recognition capability of CVCAM is excessively weakened. This result indicates that combining the output of a well-trained CVCAM model with the local distribution information provided by KNN can achieve the best decision-making

effect, demonstrating the effective complementarity between param-eterized and non-parameterized methods.

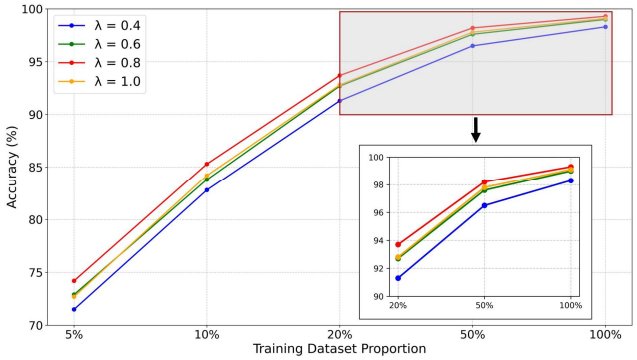


Fig. 6. Effect of different λ on average accuracy.

IV. Conclusion

This paper proposes an innovative SEI method named CVCAM-K to address the issues of insufficient selectivity in processing multi-channel features and the neglect of data correlations by the parameterized decision mechanism in Mamba-based methods. CVCAM-K introduces CBAM to construct the CVCAN network, which effectively enhances key features. It also integrates KNN to build a recognition framework that combines parameterized and non-parame-terized decision-making. Experimental results demonstrate that this work provides an efficient and reliable solution for the SEI field.

Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 62171470), Natural Science Foundation of Henan Province (No.252300420990), Youth Talent Recruitment Project in the Military Science and Technology Field (No.2022-JCJQ-QT-028) and Outstanding Youth Science Foundation of Henan Province (No. 242300421174).

References

[1] L. Ding, S. Wang, F. Wang, and W. Zhang, “Specific Emitter Identifi cation via Convolutional Neural Networks,” IEEE Commun. Lett., vol . 22, no. 12, pp. 2591–2594, Dec. 2018, doi: 10.1109/LCOMM.2018.2 871465.

[2] N. Mazhar, R. Salleh, M. Zeeshan, and M. M. Hameed, “Role of Devi ce Identification and Manufacturer Usage Description in IoT Security: A Survey,” IEEE Access, vol. 9, pp. 41757–41786, 2021, doi: 10.110 9/ACCESS.2021.3065123.

[3] A. E. Spezio, “Electronic warfare systems,” IEEE Trans. Microw. The ory Tech., vol. 50, no. 3, pp. 633–644, Mar. 2002, doi: 10.1109/22.98 9948.

[4] G. Yan, X. Fu, Y. Wang, Q. Zhang, and G. Gui, “Radio frequency fin gerprint identification towards statistical and deep learning features: R eview, recent results and future directions,” Peer--Peer Netw. Appl., v ol. 18, no. 3, p. 116, May 2025, doi: 10.1007/s12083-024-01902-9.

[5] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional Radio Mod ulation Recognition Networks,” June 10, 2016, arXiv: arXiv:1602.041 05. doi: 10.48550/arXiv.1602.04105.

[6] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, “Deep Learnin g for RF Device Fingerprinting in Cognitive Communication Network s,” IEEE J. Sel. Top. Signal Process., vol. 12, no. 1, pp. 160–167, Feb. 2018, doi: 10.1109/JSTSP.2018.2796446.

[7] K. Huang, J. Yang, H. Liu, and P. Hu, “Channel-Robust Specific Emit ter Identification Based on Transformer,” Highlights Sci. Eng. Techno l., vol. 7, pp. 71–76, Aug. 2022, doi: 10.54097/hset.v7i.1019.

[8] P. Deng, S. Hong, J. Qi, L. Wang, and H. Sun, “A Lightweight Transf ormer-Based Approach of Specific Emitter Identification for the Auto matic Identification System,” IEEE Trans. Inf. Forensics Secur., vol. 1 8, pp. 2303–2317, 2023, doi: 10.1109/TIFS.2023.3266627.

[9] B. Wang, N. Gao, and F. Wang, “Specific Emitter Identification based on CNN and Transformer,” in 2023 5th International Academic Exch ange Conference on Science and Technology Innovation (IAECST), D ec. 2023, pp. 571–575. doi: 10.1109/IAECST60924.2023.10502919.

[10] A. Gu and T. Dao, “Mamba: Linear-Time Sequence Modeling with Se lective State Spaces,” May 31, 2024, arXiv: arXiv:2312.00752. doi: 1 0.48550/arXiv.2312.00752.

[11] J. Wang, S. Zhao, Z. Luo, Y. Zhou, S. Li, and G. Pan, “EEGMamba: An EEG foundation model with Mamba,” Neural Netw., vol. 192, p. 1 07816, Dec. 2025, doi: 10.1016/j.neunet.2025.107816.

[12] L. Lumetti, V. Pipoli, K. Marchesini, E. Ficarra, C. Grana, and F. Bol elli, “Taming Mambas for 3D Medical Image Segmentation,” IEEE A ccess, vol. 13, pp. 89748–89759, 2025, doi: 10.1109/ACCESS.2025.3 570461.

[13] H. Jiahao, M. M. Ur Rahman, T. Al-Naffouri, and T.-M. Laleg-Kirati, “Mamba-CAM-Sleep: A Mamba-based Channel Attention Model for Sleep Staging Classification,” in 2025 47th Annual International Con ference of the IEEE Engineering in Medicine and Biology Society (E MBC), Copenhagen, Denmark: IEEE, July 2025, pp. 1–6. doi: 10.110 9/EMBC58623.2025.11251806.

[14] R. Chao et al., “An Investigation of Incorporating Mamba for Speech Enhancement,” Oct. 07, 2025, arXiv: arXiv:2405.06573. doi: 10.4855 0/arXiv.2405.06573.

[15] L. Yang and X. Wu, “Specific Emitter Identification of ADS-B Signal Based on Mamba,” in 2024 6th International Conference on Electroni cs and Communication, Network and Computer Technology (ECNCT), Guangzhou, China: IEEE, July 2024, pp. 136–141. doi: 10.1109/EC NCT63103.2024.10704402.

[16] Y. Tu et al., “Large-scale real-world radio signal recognition with dee p learning,” Chin. J. Aeronaut., vol. 35, no. 9, pp. 35–48, Sept. 2022, doi: 10.1016/j.cja.2021.08.016.

[17] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” July 18, 2018, arXiv: arXiv:1807.06521. do i: 10.48550/arXiv.1807.06521.

[18] A. Aubry, A. Bazzoni, V. Carotenuto, A. De Maio, and P. Failla, “Cu mulants-based Radar Specific Emitter Identification,” in 2011 IEEE I nternational Workshop on Information Forensics and Security, Iguacu Falls, Brazil: IEEE, Nov. 2011, pp. 1–6. doi: 10.1109/wifs.2011.6123 155.

[19] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, O. A. Dobre, and H. V. Po or, “An Efficient Specific Emitter Identification Method Based on Co mplex-Valued Neural Networks and Network Compression,” IEEE J. Sel. Areas Commun., vol. 39, no. 8, pp. 2305–2317, Aug. 2021, doi: 1 0.1109/jsac.2021.3087243.

[20] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, “Tr ansfer Learning for Semi-Supervised Automatic Modulation Classific ation in ZF-MIMO Systems,” IEEE J. Emerg. Sel. Top. Circuits Syst., vol. 10, no. 2, pp. 231–239, June 2020, doi: 10.1109/JETCAS.2020.2 992128.

[21] Y. Dong, X. Jiang, L. Cheng, and Q. Shi, “SSRCNN: A Semi-Supervi sed Learning Framework for Signal Recognition,” IEEE Trans. Cogn. Commun. Netw., vol. 7, no. 3, pp. 780–789, Sept. 2021, doi: 10.1109/ TCCN.2021.3067916.

[22] K. Huang, J. Yang, H. Liu, and P. Hu, “Deep Learning of Radio Frequ ency Fingerprints from Limited Samples by Masked Autoencoding,” I EEE Wirel. Commun. Lett., pp. 1–1, 2024, doi: 10.1109/LWC.2022.3 184674.

[23] J. Gong, X. Xu, Y. Qin, and W. Dong, “A Generative Adversarial Net work Based Framework for Specific Emitter Characterization and Ide ntification,” in 2019 11th International Conference on Wireless Comm unications and Signal Processing (WCSP), Xi’an, China: IEEE, Oct. 2 019, pp. 1–6. doi: 10.1109/WCSP.2019.8927888.

[24] X. Fu et al., “Semi-Supervised Specific Emitter Identification Method Using Metric-Adversarial Training,” Nov. 28, 2022, arXiv: arXiv:22 11.15379. doi: 10.48550/arXiv.2211.15379.

[25] M. Wang, S. Fang, Y. Fan, and S. Hou, “Resource-Constrained Specif ic Emitter Identification Based on Efficient Design and Network Com pression,” Sensors, vol. 25, no. 7, p. 2293, Apr. 2025, doi: 10.3390/s2 5072293.