A Dynamic-Length Single-Step Sampling Network for Adaptive Inference in Efficient Radio Frequency Fingerprint Identification

1st Kaicong Yu
Institute of System Engineering
Academy of Military Science
Beijing, China
yukaicong@alumni.nudt.edu.cn

2nd Qiaoyun Sun*
Department of Information
Beijing City University
Beijing, China
sunqy_bupt@126.com

3rd Yi Fang
Laboratory of Electromagnetic Space
Cognition and Intelligent Control
Beijing, China
fangyi950129@163.com

4th Kai Xie
Institute of System Engineering
Academy of Military Science
Beijing, China
sherlockcongcong@outlook.com

5th Jian Yang
Laboratory of Electromagnetic Space
Cognition and Intelligent Control
Beijing, China
yuhengzi_8205@163.com

***Abstract*—Radio Frequency Fingerprint Identification (RFFI) has emerged as a promising physical layer authentication technique for Internet of Things (IoT) security. While deep learning-based methods have achieved impressive accuracy, their high computational costs pose significant challenges for deployment on resource-constrained platforms. To address this limitation, we propose a Dynamic-Length Single-Step Sampling Network (DSSN) that adaptively allocates computational resources based on sample difficulty. Our method employs a lightweight policy network to process and extract signal features in a single forward pass and compute importance scores for signal fragments. Through an adaptive threshold mechanism, we achieve adaptive variation in fragment selection, dynamically determining the optimal number of fragments for each sample. To better utilize information from discarded fragments, we introduce a fragment condensing module that employs a one-way nearest neighbor algorithm to merge pruned fragments into retained ones through weighted aggregation. The policy network is trained via reinforcement learning with a composite reward function incorporating contrastive reward, cost penalty, and accuracy reward. It achieves 81.36% accuracy with 1.42× speedup on ADS-B and 93.71% accuracy with 1.60× speedup on LTE-RFF, surpassing state-of-the-art lightweight methods.**

***Index Terms*—Deep Learning, Interpretability,Radio Frequency Fingerprint Identification,Adaptive Inference.**

I. Introduction

With the rapid growth of Internet of Things (IoT) devices [1], various types of wired or wireless terminals [2] are widely used, emphasizing the need for secure communication. The latest trend in security mechanisms is the adoption of specific emitter identification (SEI) in IoT systems, which is a physical layer-based technique that protects data transmission security by exploiting the inherent characteristics of communication channels. Specific emitter identification [3], also known as Radio Frequency Fingerprint Identification (RFFI), has become a promising approach for authentication by extracting subtle features from received signals, which originate from the inherent hardware imperfections [4] in device radio signals. With its advantages of being non-replicable, difficult to tamper with, and key-independent, RFFI has become an important technical means for solving IoT device identity recognition and security issues.

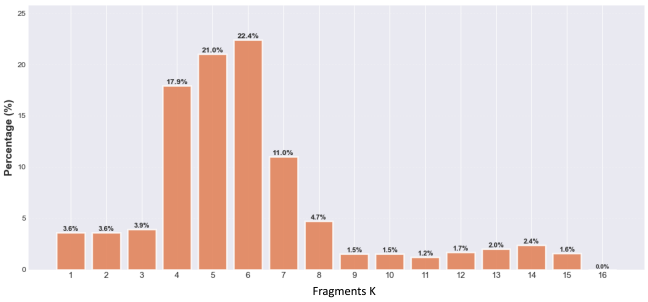

Fig. 1: We empirically demonstrate the existence and distribution patterns of sample difficulty variation on the ADS-B dataset. The distribution of k values (the number of patches selected for each sample during testing) directly proves that samples have varying recognition difficulties, thereby validating that our network can adaptively allocate computational resources for each input signal.

In recent years, deep learning [5]-[6] has transformed the SEI field, with residual neural networks and transformer architectures demonstrating powerful feature extraction capabilities for radio frequency signals, and significant progress [7]-[8] has been made in various methods for identifying complex

* Corresponding author.

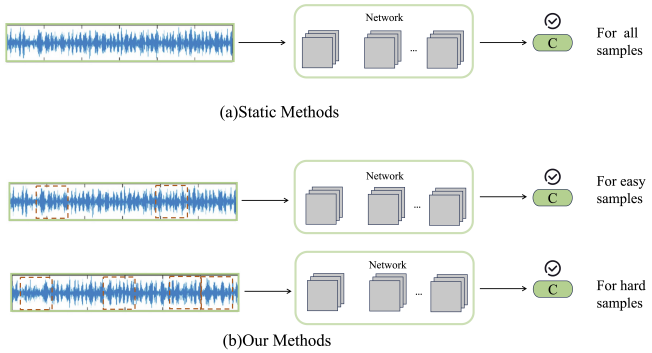

Fig. 2: Difference between our methods and static methods.

signals. While these models demonstrate strong performance on standard evaluation datasets, there exists a direct correlation between model sophistication and both accuracy gains and computational requirements. This creates significant challenges when attempting to implement such models in environments with limited resources, including mobile devices.

In many complex signal identification scenarios, it is necessary to process large amounts of communication channel data in a timely manner under limited computational resources. For example, in the vicinity of important military facilities [9], monitoring systems need to perform radio frequency fingerprint authentication on hundreds of drones flying in the airspace simultaneously, and when unauthorized or illegal drone intrusions are detected, the system needs to identify their identity features in real-time.Furthermore, RFFI methods in edge gateways need to complete device identity verification within milliseconds to meet low-latency communication requirements. This necessitates fast and accurate radio frequency fingerprint identification methods to eliminate the efficiency bottleneck of deep learning approaches.

Most existing network acceleration methods focus on reducing computational redundancy through network design [10]-[11], such as model compression [12], knowledge distillation [13]-[14], and early exit mechanisms [15]-[16]. These methods can adjust network architectures or parameters during training or inference stages to adapt to input samples, enabling the system to require less computational cost during identification.

A promising alternative is to reduce the computational complexity of emitter identification while maintaining recognition accuracy by adaptively utilizing multiple patches to represent signals in a single step according to the varying difficulty levels of each sample.As shown in Figure 1, we observe that each sample exhibits distinct semantic information across different spatial locations. Certain signal fragments may contain substantial information irrelevant to the classification task, leading to redundant computations that do not warrant excessive computational resources. We contend that for "simple" input samples, short sequence representations suffice for accurate predictions. Irrelevant fragments can be skipped to enhance the network's inference efficiency.

Specifically,we design a RFFI framework for dynamically locating and attending to important fragments of each sample. The core challenge faced by the method is how to determine the importance of each fragment in feature signal recognition. We train a simple yet effective policy network to quickly browse through the entire signal to obtain coarse global information, which then guides the selection of the most valuable fragment combinations for subsequent identification. Due to the non-differentiability of fragment selection, we employ reinforcement learning methods to automatically learn dynamic fragment selection strategies. Since training the policy network in a vast search space consumes expensive computational resources, we introduce a threshold-based approach that allows the network to prune fragments whose scores evaluated by the policy network fall below a learnable threshold. This enables the fragment length for each sample to adaptively vary according to the input sequence.

To achieve a more efficient strategy with better performance, we believe that information from unselected signal fragments should be better utilized. We propose a signal condensing module that partitions the post-decision fragments into reserved and pruned subsets. Through a nearest-neighbor matching strategy, we establish one-to-many mappings between reserved fragments and pruned fragments, employing a similarity-based fusing mechanism that selectively merges pruned tokens into their corresponding reserved fragments via weighted aggregation. This ensures that the number of reserved fragments remains constant. The main difference between our method and static methods is illustrated in Figure 2.

Extensive experiments on two benchmark datasets demonstrate the effectiveness of our approach. On the ADS-B dataset, DSSN achieves 81.36% accuracy with a 1.42× speedup, while on the LTE-RFF dataset, it attains 93.71% accuracy with a 1.60× speedup, surpassing all state-of-the-art lightweight methods in both accuracy and efficiency. These results validate that our adaptive inference framework can effectively reduce computational redundancy while maintaining superior recognition performance, making it particularly suitable for deployment on resource-constrained IoT platforms.We will make our code available upon acceptance.

II. Related Work

A. DL-based Methods for RFFI

Deep learning–based SEI methods that use deep neural networks can automatically extract features, enabling them to process raw or transformed signals and directly identify transmitters. For example, Hua [17] proposes the first method that applies knowledge graphs and adaptive feature combination to specific emitter identification, which significantly reduces the computational complexity of deep learning methods while maintaining an identification accuracy above 99.2%.In another study, Peng [18]-[19] utilized sample implicit regularization and label implicit regularization with depthwise separable convolutions for classification, demonstrating high practical applicability.

B. Efficient Networks

Research on efficient neural networks can be categorized into static methods and dynamic methods from the perspective of inference paradigm. In static methods, there are network architectures obtained through manual or automatic design [20]-[21], network models obtained through pruning and quantization, as well as networks trained through knowledge distillation.

MSDNet [22] achieves efficient image classification by incorporating multiple early-exit classifiers within a single deep convolutional neural network, combined with dense connectivity and multi-scale features.DRNet [23] reduces CNN computational costs by dynamically selecting the optimal input resolution for each image based on its difficulty level. The method uses a lightweight resolution predictor to choose from candidate resolutions. SANet [24] proposes a spatially adaptive inference method that reduces CNN computational costs by computing features only at sparsely sampled locations and reconstructing the rest through interpolation. Similarly,SwiftNet [25] integrates branch classifiers into backbone networks and employs an entropy-based multidimensional early-exit strategy across layers, categories, and fragments, enabling high-confidence samples to exit early.

III. Method

A. Overview

Figure 3 illustrates an overview of our approach. For the input signal, we initially partition it into a fixed number of candidate fragments (T fragments), and then employ a lightweight policy network to extract features from each candidate fragment and compute importance scores. Based on a learnable adaptive threshold, we achieve differentiable automated optimal fragment selection through Gumbel-Softmax reparameterization, where this process is realized via reinforcement learning (incorporating three weighted optimization objectives: contrastive reward, cost reward, and accuracy reward). The fragment condensing module leverages a feature similarity mechanism to enable discarded fragments to learn complementary information from retained fragments, further enhancing the representation capability of the integrated fragment. Subsequently, the classifier f_C takes the integrated fragment (comprising the concatenated sequence of multiple selected fragments) as input and predicts the action class through a backbone network. Notably, our method requires only a single forward pass to complete fragment sampling without iterative optimization, thereby significantly reducing inference latency. The subsequent sections will elaborate on the design principles, training strategies, and experimental validation of these components.

B. Network Architecture

1) Policy Network: The policy network π serves as a lightweight feature extraction and decision module, designed to obtain global contextual features of all signal fragments and determine which fragments can be used to form a salient fragment set representing the signal. The policy network adds minimal computation relative to the downstream classifier.

Formally, given a candidate set $\{s_1, s_2, \ldots, s_T\}$ uniformly partitioned along the sequence dimension with size $C \times L$, they are first resized to a lower resolution $C \times \tilde{L}$, and then fed into the policy network for global feature extraction. Through multi-layer convolution and adaptive pooling, deep features $\boldsymbol{z}^S = \{z_1^S, z_2^S, \ldots, z_T^S\}$ are extracted from each candidate fragment at downsampled signal resolution, where z_i^S encodes the global contextual information of the i-th fragment.

Subsequently, the policy network maps the global features to normalized importance scores through a lightweight fully connected layer f_L and Sigmoid activation function ϕ:

$$\boldsymbol{p}^L = \{p_1^L, p_2^L, \ldots, p_T^L\} = \phi(f_L(\{z_1^S, z_2^S, \ldots, z_T^S\})) \quad (1)$$

where $p_i^L \in [0, 1]$ represents the importance probability of the i-th fragment, quantifying the contribution of that fragment to the final classification task.

2) Backbone Network: Our method employs ResNet1D as the backbone network, which consists of multiple stacked residual blocks. It receives K selected one-dimensional signal fragments from our method and outputs the recognition result of the signal. Specifically, the backbone network f processes K fragments $\{s_1, s_2, \ldots, s_K\}$ after selection and merging (where $K \leq N$, and N is the total number of candidate fragments). The selected fragments are transposed and reshaped along the channel dimension, then concatenated into a continuous sequence, i.e.,

$$p = f(\text{concat}(\{s_1, s_2, \ldots, s_K\})), \quad (2)$$

where p denotes the probability scores for each class. Notably, the classifier f_C accounts for the majority of the computational overhead in our framework.

C. Adaptive Threshold Learning for Fragment Selection

We apply a threshold-based fragment selection mechanism that filters important fragments as the input signal passes through the policy network, reducing the number of fragments K fed into the classifier. This is schematically illustrated in Figure 3. For fragment selection, due to the enormous search space that would significantly increase training cost, we must define a metric to determine important fragments. A fragment is retained only when its importance score is higher than the threshold θ. Note that the importance score s_i refers to the normalized probability p_i^L obtained from the policy network.

Our approach introduces a selection mechanism through a binary filter $M(\cdot) : \{1, \ldots, N\} \rightarrow \{0, 1\}$, determining the retention status of each fragment:

$$M(s_i) = \begin{cases} 1, & \text{if } s_i > \theta \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where N is the total number of candidate fragments, and s_i represents the $i-th$ fragment's score. The selected fragments will be fed into the backbone network for recognition, thereby reducing computational complexity.

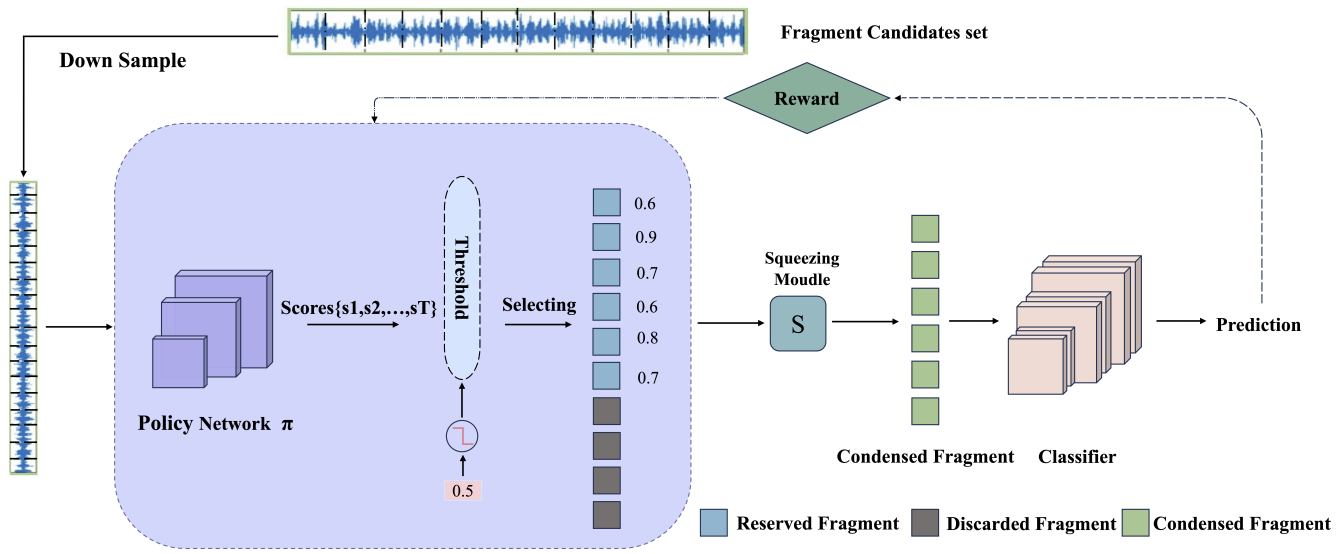

Fig. 3: Method Overview. For a given set of signal candidate fragments, our architecture first inputs all candidate fragments into the policy network π for importance scoring. Through an adaptive threshold mechanism, based on the score distribution output by the policy network, we derive an optimal fragment selection strategy that activates a subset of K key fragments. To further exploit and compress redundant information, we introduce a fragment condensing mechanism that fuses information from discarded fragments into retained fragments, forming a condensed signal representation. Subsequently, the condensed fragments are concatenated and fed into a classifier to obtain recognition results. Based on the classification results, we construct a comprehensive reward function to perform end-to-end optimization of the policy network through reinforcement learning.

Establishing an appropriate threshold presents a significant challenge. The threshold exhibits variation not only among different datasets but also across various tasks. To overcome this, we make the threshold a learnable parameter. Nevertheless, several obstacles arise. The binary characteristic of M means that the rejected fragments cannot contribute to the gradient flow. Additionally, since M is non-differentiable, gradients cannot backpropagate to the threshold. We resolve these issues by implementing a soft sampling strategy that mimics the behavior of hard sampling while maintaining gradient propagation capabilities.

Our soft sampling strategy replaces the hard filter M with a continuous approximation using the sigmoid functioon σ:

$$\tilde{M}(s_i) = \sigma\left(\frac{s_i - \theta}{T}\right), \tag{4}$$

where T is the temperature parameter, $\theta = \theta_{\text{base}} + \theta_{\text{offset}}$ is the learnable threshold, θ_{base} is the base threshold (set to 0.5), and θ_{offset} is the learnable offset. When the temperature T is sufficiently small, $\tilde{M}(s_i)$ closely approximates the hard mask $M(s_i)$.

During training, we apply the soft mask to features:

$$\tilde{z}_i^S = \tilde{M}(s_i) \cdot z_i^S, \tag{5}$$

where z_i^S is deep feature. If the importance score s_i of a fragment is far below the threshold, its weight approaches zero, thus having little impact on the final prediction. Consequently, while the soft sampling strategy functionally mirrors hard sampling, its differentiable nature enables backpropagation and facilitates gradient-based threshold optimization.

D. Fragment Condensing

After filtering candidate fragments, we introduce a fragment condensing module. The selected candidate fragments contribute most to the final recognition, and we aim to design a process that can both retain most key fragments and integrate the remaining information while maintaining the overall model performance. We condense the features of discarded fragments into retained fragments without generating additional fragments. Specifically, we employ a one-way nearest neighbor matching algorithm to map discarded fragments to selected fragments through a many-to-one pattern.

For each discarded fragment in Q_d, we identify its nearest fragment from the selected fragment set Q_r as its target fragment:

$$x^{target} = \underset{x_j \in Q_r}{\operatorname{argmax}}\, c_{i,j}. \tag{6}$$

A similarity matrix $c_{i,j}$ for all $i \in Q^r$ and $j \in Q^d$ represents the interactions between the fragments for matching.

Note that since the fragment matching process is unidirectional and follows a many-to-one pattern, a single target fragment may receive multiple discarded fragments, while

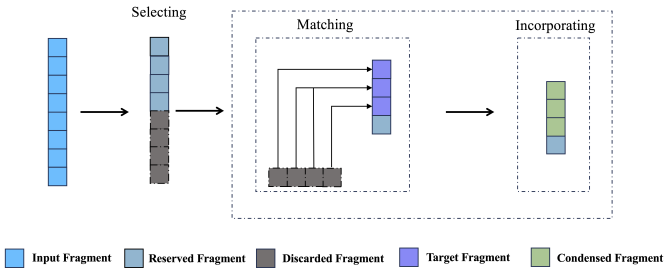

Fig. 4: To leverage information in discarded fragments, we propose a fragment condensation method to better compress discarded fragments into target fragments.

some selected fragments may not be matched with any discarded fragments. A mask matrix $M \in \mathbb{R}^{N^d \times N^r}$ is then constructed to capture the matching results, where its elements are computed as:

$$m_{i,j} = \begin{cases} 1, & \text{if } x_j \text{ is the target fragment of } x_i; \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where N^d and N^r represent the counts of discarded and reserved fragments, respectively. Through this masking mechanism, the following fusion process can be performed via standard matrix operations on Q^r and Q^d, while eliminating the interference from unmatched fragment pairs. We define the similarity matrix as follows:

$$c_{i,j} = \frac{x_i^T x_j}{\|x_i\|\|x_j\|}, \quad \text{for } i \in Q^d, j \in Q^r. \tag{8}$$

Each reserved fragment $\boldsymbol{z}_j$ undergoes an update by incorporating information from the discarded fragments:

$$\boldsymbol{u}_j = w_j \boldsymbol{z}_j + \sum_{\boldsymbol{x}_i \in Q^d} w_i \boldsymbol{z}_i, \tag{9}$$

where w_i represents the contribution weight of each discarded fragment $\boldsymbol{z}_i \in Q^d$, w_j denotes the weight assigned to the reserved fragment itself, and $\boldsymbol{u}_j$ corresponds to the resulting updated fragment. The contribution weight w_i is determined by both the mask value $m_{i,j}$ and the similarity score $c_{i,j}$:

$$w_i = \frac{\exp(c_{i,j}) m_{i,j}}{\sum_{\boldsymbol{x}_i \in Q^d} \exp(c_{i,j}) m_{i,j} + \mathrm{e}}. \tag{10}$$

The reserved fragment maintains the highest contribution weight w_j, since its self-similarity $c_{j,j}$ equals unity:

$$w_j = \frac{\mathrm{e}}{\sum_{\boldsymbol{x}_i \in Q^d} \exp(c_{i,j}) m_{i,j} + \mathrm{e}}. \tag{11}$$

Based on these formulations,reserved fragments that are not selected as target fragments preserve their original values, whereas discarded fragments are merged into their corresponding target fragments, effectively replacing them.

Notably, our matching and fusion mechanism guarantees that the total number of output fragments equals that of the reserved fragments, thus preserving a consistent dimensionality for computational efficiency during inference.

E. Training Algorithm

We design a three-stage training scheme to improve the accuracy of our proposed method.

Stage I: Pretrained Classifier We train the backbone network ResNet on the task dataset to minimize its cross-entropy loss, which lays the foundation for training the policy network.

Stage II: Joint Optimization of Policy Network and Classifier

In this stage, we train the policy network π using reinforcement learning methods while fine-tuning the classifier f_c. We use two independent optimizers to update different network components separately. Specifically:

1) Reward Mechanism: For the input candidate fragments $\{s_1, s_2, \ldots, s_T\}$, the policy network π first predicts importance scores for each fragment and generates sampling masks based on the threshold. To evaluate the action benefits of the policy network, we construct three reward signals:

Contrastive Reward

The contrastive reward measures the performance advantage of policy sampling over random sampling:

$$R_{\text{contrast}} = \text{conf}_\pi - \text{conf}_{\text{rand}} \tag{12}$$

$\text{conf}_\pi = \boldsymbol{p}_y^L$ represents the confidence with the true label y after the fragments selected by the policy network pass through the classifier $\text{conf}_{\text{rand}} = \mathbb{E}_{\{s_t\} \sim \text{RandomSample}(s)}[p_y^L]$ represents the confidence after randomly and uniformly sampling the same number of fragments.

With this contrastive reward function, Our goal is for the policy network π to identify fragments that yield the greatest confidence improvements. However, randomly selecting fragments may also consistently lead to large rewards. For example, when certain key features are present in the input sequence, the confidence value naturally increases regardless of the selection strategy. We hypothesize that such cases could mislead the optimization of π, as all actions receive positive reinforcement from the substantial rewards, even if the action is not truly superior. To address this issue, we adopt the contrastive reward function. By comparing the action taken by the policy network with the average performance of a random policy, only when strategically selecting fragments leads to significant changes in the classifier's prediction will meaningful reward variations be produced.

Therefore, a positive value indicates that the policy significantly outperforms random selection, a negative value suggests that the policy needs improvement, while a value close to zero indicates that the policy provides no advantage beyond the random baseline.

Cost Reward

The cost reward encourages the model to select fewer fragments to accelerate inference:

$$R_{\text{e}} = \frac{\sum_{i=1}^{T} \|\tilde{M}(s_i)\|}{T} \tag{13}$$

where K is the number of actually selected fragments and T is the total number of candidate fragments. This reward

term is added to the total reward with a negative weight (i.e., $-\beta \cdot r_{\text{efficiency}}$), so that selecting fewer fragments yields higher rewards, thus achieving computational acceleration.

Accuracy Reward

The accuracy reward directly reflects the correctness of the classification result:

$$R_{\text{a}} = \begin{cases} 1, & \text{if } \arg\max(f_c(\{s_1, \ldots, s_T\})) = y \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

This reward provides the most direct performance feedback signal based on whether the selected fragments lead to a correct classification.

2) Objective Function: To maximize the reward at each time fragment, we define the following optimization objective, where the objective function to be optimized at each time step is given by:

$$\max \mathcal{J}(\theta) = \max \sum_{i=1}^{T} \left(\mathbb{E}_{a_i \sim \pi_{i,\theta}}[R_i(a_i)] \right) \tag{15}$$

where $a_i = 1$ indicates that the i-th candidate fragment is incorporated into the classification process, while $a_i = 0$ indicates that the fragment is discarded. The parameters θ of the policy network are updated via gradient ascent, with the step size controlled by the learning rate α:

$$\theta \leftarrow \theta + \alpha \nabla_\theta [\mathcal{J}(\theta)] \tag{16}$$

Following the theoretical framework of the policy gradient algorithm, the loss function can be derived as:

$$\mathcal{L} = -\sum_{i=1}^{T} \left(R_i(a_i) \log \pi_{i,\theta}(a_i | \mathcal{S}_t) \right) \tag{17}$$

Regularization

To prevent the learnable threshold from degenerating to excessively low values that would trivially retain all fragments for minimizing classification loss, we introduce a regularization term to penalize excessive fragment selection.

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{reg}} \tag{18}$$

where $\mathcal{L}_{cls}$ is the original cross-entropy loss, $\mathcal{L}_{\text{reg}} = \frac{1}{T} \sum_{i=1}^{T} \|\tilde{M}(s_i)\|$ is the L1 loss, T denotes the number of candidate fragments, and λ is the hyperparameter.

IV. Experiment

A. Datasets

ADS-B signal dataset [26] :It is a large scale real world radio signal dataset based on the Automatic Dependent Surveillance- Broadcast system. It contains 20,000 training samples and 5,000 test samples.

Long-term evolution (LTE) signal dataset [27] : It contains signals from 10 LTE devices, with 150,000 training samples and 50,000 test samples in total. This dataset is specifically designed for wireless communication device authentication.

TABLE I: Comparison of different methods on ADS-B and LTE-RFF datasets

Method	Datasets			
	ADS-B		LTE-RFF	
	Acc.	FLOPs (speedup)	Acc.	FLOPs (speedup)
Resnet [28]	81.78	602M (1.00×)	94.25	248M (1.00×)
ShuffleNet V2 [29]	80.48	436M (1.38×)	92.63	194M (1.28×)
SANet [30]	80.52	451M (1.33×)	92.31	132M (1.88×)
BNAS [31]	79.75	581M (1.04×)	91.05	234M (1.06×)
DRN [32]	80.27	467M (1.29×)	91.92	196M (1.27×)
CHEX [33]	80.82	518M (1.16×)	92.83	203M (1.22×)
MRCNN [34]	79.62	569M (1.06×)	90.81	225M (1.10×)
DSSN(ours)	**81.36**	**425M (1.42×)**	**93.71**	**155M (1.60×)**

B. Settings

For each dataset, we uniformly split the sequence length into 16 fragments and train using an online reinforcement learning approach with a dual-optimizer strategy: Adam optimizers update the classifier at a learning rate of 1×10^{-3} and update the policy network at a learning rate of 1×10^{-4} every 5 batches. Training is conducted for 300 epochs with a batch size of 128. The reward function consists of three weighted components (contrastive reward weight $\alpha = 1.0$, cost reward weight $\beta = 2.0$, and accuracy reward weight $\gamma = 2.0$). The main network loss is cross-entropy plus 0.01 times L_1 regularization, with the initial threshold value set to 0.5.

C. Main Results

Our experimental results on the ADS-B and LTE-RFF datasets are shown in Table I. DSSN achieves the highest accuracy among all lightweight methods on both datasets while maintaining excellent computational efficiency.

On the ADS-B dataset, DSSN achieves 81.36% accuracy with 425M FLOPs, delivering a 1.42× speedup. While the accuracy is 0.42 percentage points lower than the baseline ResNet, DSSN reduces computation by 29.4%. Notably, DSSN outperforms the second-best CHEX (80.82%, 1.16×) and SANet (80.52%, 1.33×) in both accuracy and overall efficiency.

On the LTE-RFF dataset, DSSN demonstrates even stronger performance, achieving 93.71% accuracy with only 155M FLOPs and a 1.60× speedup. DSSN surpasses all lightweight methods, outperforming the second-best CHEX (92.83%) by 0.88 percentage points while reducing computational cost by 23.6%. Although SANet achieves a higher speedup ratio (1.88×) with 132M FLOPs, its accuracy (92.31%) is 1.40 percentage points lower than DSSN. Compared to the baseline ResNet, DSSN reduces computational cost by 37.5% with only a 0.54 percentage point accuracy drop.

Overall, as illustrated in Figure 5, DSSN (marked with a star) is positioned in the upper-left optimal region of the accuracy-FLOPs scatter plots for both datasets, achieving the

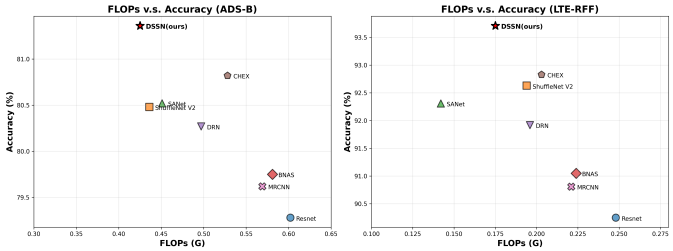

Fig. 5: Comparison of DSSN with state-of-the-art methods on ADS-B and LTE-RFF datasets.

TABLE II: Results of different fragmentation size on the ADS-B dataset

Fragmentation	4	8	16	32
Acc.	80.7	81.17	**81.36**	80.92
MFLOPs(speedup)	438(1.01×)	426(1.03×)	**425**(1.03×)	441(1.00×)

best accuracy-efficiency trade-off among lightweight methods. This validates that our adaptive fragment selection mechanism can effectively reduce computational redundancy while maintaining superior recognition performance.

D. Ablation Study

Various fragmenting methods We conducted experiments on the ADS-B dataset with different fragment partitioning strategies, and the results are shown in the Table II. We observed that different fragment sizes have minimal impact on FLOPs and accuracy. For simplicity, we uniformly divided the signals into 16 fragments, as it achieved the best balance between accuracy and computational cost.

Effectiveness of adaptive threshold

To validate the performance of the adaptive threshold method, we systematically evaluated the impact of different fragment candidate quantity configurations on the LTE-RFF dataset. In the experiments, we compared fixed fragment quantity methods with the proposed Adaptive Threshold method, with results shown in Table III.

The experimental results reveal a severe efficiency bottleneck in fixed fragment configurations. When fragments increase from 4 to 12, accuracy improves marginally from 91.13% to 91.68% (only 0.55 percentage points), while computational overhead surges from 147 to 376 MFLOPs (a 155.8% increase). Notably, increasing from 8 to 12 fragments yields merely a 0.14 percentage point accuracy gain but requires 30.6% additional computation, indicating that excessive fragments accumulate redundant information without effective performance gains.

In contrast, the proposed adaptive threshold method achieves the highest accuracy. By dynamically determining the optimal number of fragments for each sample based on content complexity, it intelligently allocates computational resources, using fewer fragments for simple samples and more for complex ones, thereby maximizing both efficiency and accuracy.

TABLE III: Effectiveness of Adaptive Threshold on the lte-rff dataset

Fragment Candidates	4	6	8	12	AT(ours)
Acc.	91.13	91.45	91.54	91.68	**91.71**
MFLOPs	**147**	192	288	376	155

TABLE IV: Comparison between Fragment Condensing and the current parameter-free method

Method	ADS-B	LTE-RFF
-	79.48	92.42
Random	79.63	93.25
MaxPool1D	80.41	92.86
AvgPool1D	80.85	93.52
Fragment Condensing(ours)	**81.36**	**93.71**

Ablation study of fragment condensing

To further demonstrate the superiority of our proposed fragment condensing method, we compared it with other parameter-free compression methods. The results in the Table IV show that our method provides performance improvements that are significantly superior to existing compression methods. Fragment compression offers a higher performance ceiling by preserving more of the original information.

Effectiveness of the Contrastive Reward

To assess the effectiveness of the proposed contrastive reward mechanism, we conduct controlled experiments comparing two reward formulations. The Monte Carlo estimator for employs single-sample approximation for computational efficiency. We train two DSSN variants: one not adopting rewards, and another adopting our contrastive formulation that normalizes gains against random selection baselines.

Experimental results in Table V reveal consistent superiority of the contrastive approach. The accuracy improvements are 0.44 percentage points on ADS-B (81.36% vs 80.92%) and 0.57 percentage points on LTE-RFF (93.84% vs 93.27%). This validates our hypothesis that contrasting learned policies against stochastic baselines mitigates spurious reward signals arising from inherently informative fragments, thereby enabling more effective policy optimization.

V. Conclusions

This paper proposes an accurate and efficient Dynamic-Length Single-Step Sampling Network (DSSN) for Radio Frequency Fingerprint Identification that adaptively allocates computational resources based on sample difficulty. By employing a lightweight policy network trained via reinforcement learning to avoid heavy computational overhead, an adaptive

TABLE V: Comparison Between Contrastive and Incremental Reward Functions

Reward	ADS-B	LTE-RFF
Increments	80.92	93.27
Contrastive	81.36	93.84

threshold mechanism for dynamic fragment selection, and a fragment condensing module for information preservation, our method achieves superior performance on benchmark datasets while significantly reducing computational cost.

Currently, the rapid development of large language models has made them a focal point of attention, and in the near future, large electromagnetic models will also be widely promoted and applied. However, due to the absolute volume of tokens in the Transformer architecture causing enormous inference overhead, we hope to propose a training method: using small electromagnetic models to guide large electromagnetic models to focus only on important tokens, thereby significantly reducing computational load while preserving the essential information of correct answers. Meanwhile, for relatively simple samples, the small electromagnetic model can directly determine the correct answer without invoking the large model, making full use of the computational cost of the small model.

REFERENCES

[1] X. Qi, A. Hu, and T. Chen, "Lightweight radio frequency fingerprint identification scheme for v2x based on temporal correlation," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1056–1070, 2023.

[2] J. Zhang, R. Woods, M. Sandell, M. Valkama, A. Marshall, and J. Cavallaro, "Radio frequency fingerprint identification for narrowband systems, modelling and classification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3974–3987, 2021.

[3] Z. Cai, Y. Wang, G. Gui, and J. Sha, "Toward robust radio frequency fingerprint identification via adaptive semantic augmentation," *IEEE Transactions on Information Forensics and Security*, 2024.

[4] P. Deng, S. Hong, J. Qi, L. Wang, and H. Sun, "A lightweight transformer-based approach of specific emitter identification for the automatic identification system," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2303–2317, 2023.

[5] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE communications letters*, vol. 22, no. 12, pp. 2591–2594, 2018.

[6] B. Wang, N. Gao, and F. Wang, "Specific emitter identification based on cnn and transformer," in *2023 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*. IEEE, 2023, pp. 571–575.

[7] Y. Zhang, Q. Zhang, H. Zhao, Y. Lin, G. Gui, and H. Sari, "Multisource heterogeneous specific emitter identification using attention mechanism-based rff fusion method," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 2639–2650, 2024.

[8] H. Wan, Q. Wang, X. Fu, Y. Wang, H. Zhao, Y. Lin, H. Sari, and G. Gui, "Vc-sei: Robust variable-channel specific emitter identification method using semi-supervised domain adaptation," *IEEE Transactions on Wireless Communications*, vol. 23, no. 12, pp. 18 228–18 239, 2024.

[9] X. Li, Y. Zhao, and G. Cheng, "Bitfl: Bitstream-based lightweight federated learning with differential privacy for radio frequency fingerprint identification of drones," *Computer Networks*, p. 111859, 2025.

[10] Y. Zniyed, T. P. Nguyen *et al.*, "Enhanced network compression through tensor decompositions and pruning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 3, pp. 4358–4370, 2024.

[11] K. Xie, C. Liu, X. Wang, X. Li, G. Xie, J. Wen, and K. Li, "Neural network compression based on tensor ring decomposition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 3, pp. 5388–5402, 2024.

[12] D. Gurevin, M. Shan, S. Huang, M. A. Hasan, C. Ding, and O. Khan, "Prunegnn: Algorithm-architecture pruning framework for graph neural network acceleration," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 108–123.

[13] S. Sun, W. Ren, J. Li, R. Wang, and X. Cao, "Logit standardization in knowledge distillation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 15 731–15 740.

[14] J. Gou, Y. Chen, B. Yu, J. Liu, L. Du, S. Wan, and Z. Yi, "Reciprocal teacher-student learning via forward and feedback knowledge distillation," *IEEE transactions on multimedia*, vol. 26, pp. 7901–7916, 2024.

[15] Y. Sepehri, P. Pad, A. C. Yüzügüler, P. Frossard, and L. A. Dunbar, "Hierarchical training of deep neural networks using early exiting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 6271–6285, 2024.

[16] Z. Zeng, Y. Hong, H. Dai, H. Zhuang, and C. Chen, "Consistentee: A consistent and hardness-guided early exiting method for accelerating language models inference," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 506–19 514.

[17] M. Hua, Y. Zhang, J. Sun, B. Adebisi, T. Ohtsuki, G. Gui, H.-C. Wu, and H. Sari, "Specific emitter identification using adaptive signal feature embedded knowledge graph," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4722–4734, 2023.

[18] Y. Peng, X. Zhang, L. Guo, C. Ben, Y. Liu, Y. Wang, Y. Lin, and G. Gui, "Enhanced specific emitter identification with limited data through dual implicit regularization," *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 26 395–26 405, 2024.

[19] Y. Zhang, Y. Peng, J. Sun, G. Gui, Y. Lin, and S. Mao, "Gpu-free specific emitter identification using signal feature embedded broad learning," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 13 028–13 039, 2023.

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[21] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 815–825.

[22] G. Huang, D. Chen, T. Li, F. Wu, L. Van Der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," *arXiv preprint arXiv:1703.09844*, 2017.

[23] M. Zhu, K. Han, E. Wu, Q. Zhang, Y. Nie, Z. Lan, and Y. Wang, "Dynamic resolution network," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 319–27 330, 2021.

[24] Z. Xie, Z. Zhang, X. Zhu, G. Huang, and S. Lin, "Spatially adaptive inference with stochastic feature sampling and interpolation," in *European conference on computer vision*. Springer, 2020, pp. 531–548.

[25] Z. Wen, Q. Li, Y. Wang, L. Xu, H. Shao, G. Sun, and S. Wang, "Swiftnet: A cost-efficient deep learning framework with diverse applications," *IEEE Transactions on Industrial Informatics*, 2024.

[26] T. Ya, L. Yun, Z. Haoran, W. Yu, G. Guan, M. Shiwen *et al.*, "Large-scale real-world radio signal recognition with deep learning," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 35–48, 2022.

[27] X. Yang and D. Li, "Led-rff: Lte dmrs-based channel robust radio frequency fingerprint identification scheme," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1855–1869, 2023.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[29] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[30] Y. Han, G. Huang, S. Song, L. Yang, Y. Zhang, and H. Jiang, "Spatially adaptive feature refinement for efficient inference," *IEEE Transactions on Image Processing*, vol. 30, pp. 9345–9358, 2021.

[31] Z. Ding, Y. Chen, N. Li, D. Zhao, Z. Sun, and C. P. Chen, "Bnas: Efficient neural architecture search using broad scalable architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 5004–5018, 2021.

[32] M. Zhu, K. Han, E. Wu, Q. Zhang, Y. Nie, Z. Lan, and Y. Wang, "Dynamic resolution network," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 319–27 330, 2021.

[33] Z. Hou, M. Qin, F. Sun, X. Ma, K. Yuan, Y. Xu, Y.-K. Chen, R. Jin, Y. Xie, and S.-Y. Kung, "Chex: Channel exploration for cnn model compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 287–12 298.

[34] T. Cui, R. Li, Z. Li, L. Shi, and H. Zhang, "Multi-resolution convolutional neural network for specific emitter identification," in *Proceedings of the 2024 6th International Symposium on Signal Processing Systems*, 2024, pp. 56–62.