# Few-shot Learning and Data Augmentation for Cross-Domain UAV Fingerprinting

Tianya Zhao
tzhao010@fiu.edu
Florida International University

Ningning Wang
nwang012@fiu.edu
Florida International University

Shiwen Mao
smao@ieee.org
Auburn University

Xuyu Wang
xuywang@fiu.edu
Florida International University

## Abstract

In this paper, we propose a novel approach to cross-domain unmanned aerial vehicle (UAV) authentication using radio frequency (RF) fingerprinting based on prototypical networks (PTNs). UAVs present a unique challenge for RF fingerprinting due to their hovering motion, which creates more diverse signal domains compared to other RF devices like Wi-Fi. This results in a severe domain shift problem, where well-trained models struggle to generalize to unseen domains. To address this issue without incurring significant costs in data collection and model retraining, we employ PTNs, a few-shot learning paradigm that enhances cross-domain performance and system viability. We further improve our method's effectiveness by incorporating fine-tuning with data augmentation, maintaining system viability while improving performance. Comprehensive experimental results demonstrate that our approach significantly mitigates domain shift, achieving up to a 20% improvement in cross-domain accuracy for UAV fingerprinting.

## CCS Concepts

• **Security → UAV security**.

## Keywords

Unmanned aerial vehicle (UAV) radio frequency (RF) fingerprinting, few-shot learning, domain shift

**ACM Reference Format:**
Tianya Zhao, Ningning Wang, Shiwen Mao, and Xuyu Wang. 2024. Few-shot Learning and Data Augmentation for Cross-Domain UAV Fingerprinting. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3636534.3698248

## 1 Introduction

In recent years, unmanned aerial vehicles (UAVs) have become increasingly prevalent across various domains, prompting a growing need for study into their security issues [2]. One of the fundamental topics is device authentication. However, traditional authentication methods may not be suitable for UAV classification. Classic techniques based on angle of arrival (AOA) and time difference of arrival (TDOA) are often ineffective due to the mobile nature of even legitimate APs within the operation area [8].

In response to these challenges, radio frequency (RF) fingerprinting has emerged as a promising authentication method in related domains [4, 15]. This technique involves classifying the inherent physical imperfections in the analog circuitry of RF emitters that arise during the manufacturing process. These imperfections slightly affect the transmitted signals without compromising device performance, creating unique fingerprints for each device. Compared to traditional authentication methods, RF fingerprinting offers enhanced robustness against tampering and spoofing, thereby improving the security of RF devices [11]. By leveraging powerful deep neural networks (DNNs), RF fingerprinting can achieve high performance and be easily deployed. These characteristics make RF fingerprinting particularly well-suited for UAV authentication, addressing the limitations of conventional methods and providing a more reliable solution for this emerging security challenge.

**Challenges.** Despite the promise of RF fingerprinting for UAV authentication, several significant challenges persist. First, while DNN-based RF fingerprinting systems can achieve high performance in known domains, they often

*ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA*

ACM ISBN 979-8-4007-0489-5/24/11
https://doi.org/10.1145/3636534.3698248

struggle with the domain shift problem, performing poorly in unknown domains such as different times or distances [5]. Second, unlike stationary RF devices like Wi-Fi routers, UAVs are constantly hovering or moving in the air. This mobility introduces complex channel variations and exacerbates the domain shift problem, making it even more challenging to maintain consistent fingerprinting performance across varying conditions [8]. Third, gathering a large dataset of UAV signals from new domains to retrain DNNs is both impractical and resource-intensive. This limitation hinders adapting UAV fingerprinting systems to new environments or conditions. Given these challenges, there is a need for a lightweight solution that can improve cross-domain fingerprinting performance without extensive retraining or unseen domain data collection, thereby significantly enhancing the viability of RF fingerprinting for UAV authentication.

**Our Solution.** To address these challenges, we propose a solution based on prototypical networks (PTNs) [7] for cross-domain UAV fingerprinting. First, we redesign the PTN to optimize a feature extractor that can identify stable fingerprint features across different domains using similarity metrics. As a few-shot learning (FSL) paradigm, PTN requires only a small amount of labeled data during inference to improve accuracy. Second, we implement a fine-tuning process for the trained feature extractor to accommodate the diverse unseen domains generated by UAVs' hovering nature. This allows the model to adapt to complex new domains. Third, we design and apply a data augmentation technique during the fine-tuning stage to further enhance classification accuracy. Overall, our contributions are as follows:

- To the best of our knowledge, this is the first work to deploy FSL to mitigate domain shift issues in UAV fingerprinting. Our approach eliminates the need for extensive data collection and cumbersome model training processes, thereby enhancing classification performance and overall system viability.
- We carefully design data augmentation and employ fine-tuning with only a few data, which further enhances system accuracy without significantly increasing overhead.
- Our comprehensive experimental evaluation shows that our proposed method can improve the classification accuracy by about 20% in the best case, demonstrating the effectiveness of our approach.

## 2 Background and Related Work

### 2.1 RF Fingerprinting

RF fingerprinting has emerged as a promising technique for identifying wireless devices based on their unique hardware imperfections. This physical layer identification method offers enhanced resistance to spoofing and replay attacks [9]. The development of powerful deep learning techniques has enabled automatic extraction of RF fingerprint features, leading to widespread adoption in various device identification applications [5, 12]. Typically, DNN-based RF fingerprinting systems use raw in-phase/quadrature (I/Q) data as input, leveraging the DNN's ability to effectively process and classify complex signal characteristics. The process of RF fingerprinting generally involves two key components: feature extraction and multi-class identification. Accurate feature extraction is crucial for successfully distinguishing different RF fingerprints [6, 16].

In UAV fingerprinting, Soltani *et al.* propose a multi-classifier scheme with a two-step score-based aggregation method and data augmentation to enhance cross-domain performance [8]. However, this method may be time-intensive for training and deployment. Zhao *et al.* employ auxiliary classifier Wasserstein generative adversarial networks (ACWGANs) for feature identification [14]. Cai *et al.* develop a lightweight backbone network using lightweight multiscale convolution (LMSC) blocks, reducing model size while improving feature extraction capabilities in a simulation environment [1].

### 2.2 Few-shot Learning

FSL offers a significant advantage over traditional deep learning approaches by enabling models to generalize to new classes and domains using only a limited number of examples [13]. This capability makes FSL particularly valuable in scenarios where data is scarce or costly to obtain. FSL allows for rapid adaptation to new tasks in data-constrained environments, a feature that has led to its deployment in various related domains [10, 15]. Consequently, it is well-suited for cross-domain UAV fingerprinting systems.

In this paper, we use a base dataset $\mathcal{E}_{\text{base}}$ to train a feature extractor $f_\theta$. Then, we create a support set $\mathcal{E}_{\text{support}}$ consisting of a small number of labeled samples and a query set $\mathcal{E}_{\text{query}}$ containing data that we need to classify. The $N$-way $K$-shot learning scheme, a common approach in FSL, refers to training the model on $N$ classes with $K$ labeled examples per class [13].

Overall, there are some key distinctions between our work and previous studies. First, we address the domain shift issue by considering both time and distance variations. Second, we adapt the PTN structure to better suit UAV fingerprinting, enabling lightweight deployment. Third, we design a data augmentation strategy combined with fine-tuning specifically tailored for UAV fingerprinting to enhance performance.

## 3 Methodology

Fig. 1 provides an overview of our proposed cross-domain UAV fingerprinting system, which consists of two main stages. During the training stage, a feature extractor is trained to extract fingerprints using the base set. In the inference stage, the feature extractor is rapidly fine-tuned to generate prototypes for UAV classification. This section will explain each stage of the system in detail.

## 3.1 Extractor Training

To address the domain shift challenge, we modify the PTN to train an effective feature extractor for cross-domain UAV fingerprinting. Our goal is to train a robust feature extractor capable of generalizing across different domains. We employ ResNet-18 [3] as the feature extractor $f_\theta$, as convolutional neural networks (CNNs) have demonstrated their ability to extract fingerprints from I/Q data [5]. To adapt ResNet-18 for our specific task, we modify only the first input layer to accommodate the dimensions of our I/Q data, which has a size of $2 \times 256$. To ensure that the embedding vectors lie on a hypersphere with a constant radius, we add an $L_2$-norm layer before the final output layer as follows:

$$f_\theta(\mathbf{x}_i) = \frac{f'_\theta(\mathbf{x}_i)}{\left\| f'_\theta(\mathbf{x}_i) \right\|_2}, \tag{1}$$

where $f'_\theta(\mathbf{x}_i)$ is the output embedding before the $L_2$-norm $\|\cdot\|_2$, and $f_\theta(\mathbf{x}_i)$ represents the final feature embedding.

After feature extraction, a final classifier $C(\cdot)$ is added on the top and can be adjusted according to the number of UAVs. The feature extractor is trained using a traditional supervised learning scheme, utilizing our base set $\mathcal{E}_{\text{base}} = \{\mathbf{x}_i, y_i\}_{i=1}^{B}$ as the training data. We employ the classic multi-class cross-entropy loss function to guide parameter optimization:

$$\mathcal{L} = -\sum_{i}^{B} y_i \cdot log(C(f_\theta(\mathbf{x}_i))). \tag{2}$$

## 3.2 Model Inference

Once a feature extractor is well-trained, we can generate prototypes for inference. The prototype for each UAV is determined by averaging all the feature embedding vectors belonging to that class. The computation of prototypes can be expressed as follows:

$$\mathbf{c}_i = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathcal{E}_{\text{data}}}^{n} f_\theta(\mathbf{x}_i), \tag{3}$$

where $\mathbf{c}_i$ denotes the prototypes of the UAV $y_i$, and $n$ denotes the number of samples in each class in the dataset. These prototypes encapsulate the essential characteristics of specific UAV classes, providing a generalized representation that remains relatively invariant across different domains. By computing prototypes for each UAV, we obtain stable representations crucial for the classification process, allowing for accurate and reliable identification across diverse domains.

Once the prototypes of each UAV have been established, the model can use them to generate predictions for new samples. This is accomplished by comparing the feature embedding of a new sample to the prototypes of each class. In our experimental setup, we quantify this comparison using cosine similarity, which is calculated as follows:

$$\mathbf{D} = d(\mathbf{c}, f_\theta(\mathbf{x}_i)) = \frac{\mathbf{c} \cdot f_\theta(\mathbf{x}_i)}{\|\mathbf{c}\|_2 \, \|f_\theta(\mathbf{x}_i)\|_2}, \tag{4}$$

where $\mathbf{D}$ represents the similarity matrix between the input sample $\mathbf{x}_i$ and the prototypes of all known UAV classes. The function $d(\mathbf{c}, f_\theta(\mathbf{x}_i))$ calculates the cosine similarity between a class prototype $\mathbf{c}$ and the feature embedding $f\theta(\mathbf{x}_i)$ of the input sample. Cosine similarity values range from $-1$ to 1, with values closer to 1 indicating higher similarity. The model assigns the input sample to the class whose prototype yields the highest similarity score, thus determining the identification result.

## 3.3 Few-shots Fine-tuning

The hovering and mobility capabilities of UAVs introduce additional complexities to different domains, necessitating fine-tuning to enhance system performance. However, our system relies on a limited number of samples from the support set, which may impede the model's ability to generalize effectively to complex tasks. To address this issue, we incorporate data augmentation during the fine-tuning process. Data augmentation is a widely adopted technique that enhances DNNs' generalization capabilities, enabling more accurate predictions on previously unseen data.

In this paper, we recognize that UAV domains significantly impact data magnitude. Consequently, we leverage this domain-specific data information to design our augmentation strategy as follows:

$$\mathbf{x}_j^{s\prime} = \mathbf{x}_j^s + \alpha \cdot N(\mu(\mathbf{x}_j^s), \sigma(\mathbf{x}_j^s); L), \tag{5}$$

where $L$ denotes the size of the data, $\mathbf{x}_j^{s\prime}$ is the augmented data derived from the support set $\mathcal{E}_{\text{support}}$, and $\mu(\mathbf{x}_j^s)$ and $\sigma(\mathbf{x}_j^s)$ represents the mean and standard deviation of the data $\mathbf{x}_j^s$, respectively. By integrating this augmentation strategy with fine-tuning, our well-trained feature extractor can better generalize to new domains.

## 3.4 Summary

Alorithm 1 describes the pseudocode for training the feature extractor $f_\theta$. The process begins with a base set $\mathcal{E}_{\text{base}}$ for initial training and a support set $\mathcal{E}_{\text{support}}$ for subsequent fine-tuning and inference. First, we use the base set to train a feature extractor that can generate stable UAV fingerprints. Next, we apply specially designed data augmentation techniques to expand the support set during the inference stage. This augmented dataset is then used to fine-tune the feature

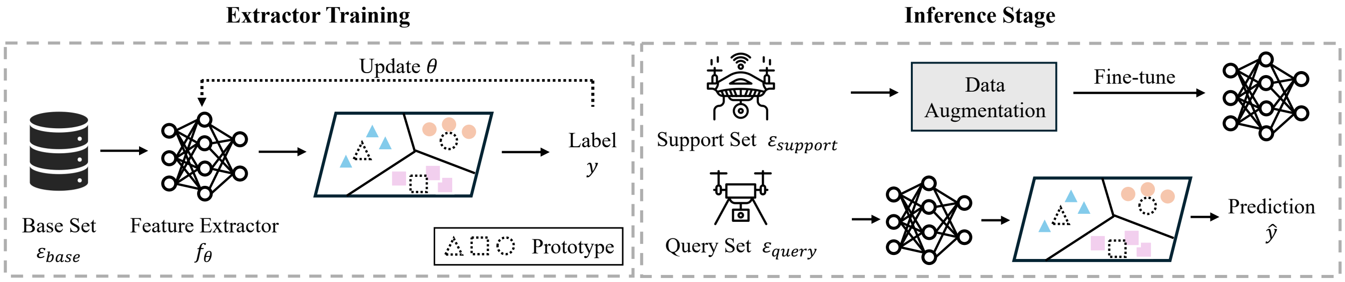

**Figure 1: Overview of our proposed cross-domain UAV fingerprinting approach. To improve performance in unseen domains, we employ fine-tuning and data augmentation techniques. The final prediction is generated by comparing prototypes with feature embeddings extracted from the input I/Q samples.**

**Algorithm 1** PTN training and fine-tuning

```
INPUT: Base set $\mathcal{E}_{\text{base}} = \{(\mathbf{x}_i, y_i)\}$, support set $\mathcal{E}_{\text{support}} = \{(\mathbf{x}^s_j, y^s_j)\}$, feature extractor $f_\theta$, classifier $C$, learning rate $lr$, fine-tuning learning rate $lr'$, hyperparameter $\alpha$
OUTPUT: fine-tuned feature extractor $f_\theta$
    Step 1: Train with base set
for number of epoch do
    for $(\mathbf{x}_i, y_i) \in \mathcal{E}_{\text{base}}$ do
        $\mathcal{L} \leftarrow CrossEntropy(C(f_\theta(\mathbf{x}_i)), y_i)$
    end for
    $\theta \leftarrow \theta - lr \cdot \nabla_\theta \mathcal{L}$
end for
    Step 2: Construct augmented set
for $(\mathbf{x}^s_j, y^s_j) \in \mathcal{E}_{\text{support}}$ do
    $\mathbf{x}^{s\prime}_j \leftarrow \mathbf{x}^s_j + \alpha \cdot N(\mu(\mathbf{x}^s_j), \sigma(\mathbf{x}^s_j); L)$
    $y^{s\prime}_j \leftarrow y^s_j$
    $\mathcal{E}_{\text{augment}} \leftarrow \mathcal{E}_{\text{support}} + \{(\mathbf{x}^{s\prime}_j, y^{s\prime}_j)\}$
end for
    Step 3: Fine-tune the feature extractor
for number of epoch do
    for $(\mathbf{x}^a_k, y^a_k) \in \mathcal{E}_{\text{augment}}$ do
        $\mathcal{L} \leftarrow CrossEntropy(C(f_\theta(\mathbf{x}^a_k)), y^a_k)$
    end for
    $\theta \leftarrow \theta - lr' \cdot \nabla_\theta \mathcal{L}$
end for
return $f_\theta$
```

extractor, adapting it to new domains. The final predictions are made by comparing the cosine similarities between feature embeddings and each prototype **c**. The class with the highest similarity to the embedding is assigned to the input.

# 4 Experimental Evaluation

## 4.1 Experiment Setup

In all experiments, the learning rate was set to 0.001. $K_{shot}$, $N_{query}$, fine-tuning epochs and fine-tuning learning rate were set to 5, 15, 10, and 0.0001 respectively. The value of $N_{way}$ was set to the size of all labels for the datasets. The experiments were conducted on a server with an Intel Xeon E5-2650L v4 CPU and 8 NVIDIA GeForce GTX 1080Ti GPU.

## 4.2 UAV Dataset

This paper conducts experiments on a public UAV fingerprinting dataset [8]. The dataset comprises 7 identical DJI M100 UAVs as transmitters, with an Ettus USRP X310 equipped with a UBX 160 USRP daughterboard serving as the receiver to collect I/Q samples from DJI's non-standard, proprietary waveform. The data collection encompasses 4 non-overlapping bursts at different times and 4 distinct distances (6, 9, 12, and 15 feet). To address the domain shift problem, we construct our base set (5112 I/Q samples) using data from distances of 6 and 9 feet, taken from bursts 1, 2, and 3. This configuration allows us to evaluate our UAV fingerprinting system across various domain partitions, considering both time and distance variations.

## 4.3 Cross-domain Results

Table 1 presents the cross-domain UAV fingerprinting results. The high classification accuracy in the source domain demonstrates CNN's strong capability for extracting UAV fingerprints. For unseen domains, results are presented as 'a-b', where 'a' represents distance in feet and 'b' represents burst time. '6-4' and '9-4' indicate tests at distances of 6 and 9 feet during burst 4, which occurs in a different time domain than the base set. Classification accuracy decreases by approximately 14% and 6% respectively in these cases. Notably, accuracy drops more significantly for unseen distances compared to different time domains. The most severe case is '15-3', where accuracy falls to only about 32%, which is inadequate for reliable UAV fingerprinting. These results suggest that while the CNN performs well in known domains, its performance degrades in unseen scenarios, particularly with changes in distance.

**Table 1: Cross-domain UAV fingerprinting results. 'Source' indicates classification accuracy in the source domain. The remaining 'a-b's show classification accuracy in unseen domains, where 'a' represents the distance in feet and 'b' represents burst time.**

| | Source | 6-4 | 9-4 | 12-1 | 12-2 | 12-3 | 12-4 | 15-1 | 15-2 | 15-3 | 15-4 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet-18 | 0.9482 | 0.8004 | 0.8804 | 0.8285 | 0.6370 | 0.6057 | 0.4383 | 0.5399 | 0.4458 | 0.3261 | 0.5515 |
| PTN | 0.9629 | 0.8238 | 0.8657 | 0.9011 | 0.5933 | 0.5467 | 0.5362 | 0.5778 | 0.4809 | 0.4422 | 0.6493 |
| + Fine-tuning | 0.9746 | 0.9079 | 0.9270 | 0.9407 | 0.7148 | 0.6296 | 0.5619 | 0.6481 | 0.6698 | 0.5259 | 0.6978 |
| + Augmentation | 0.9746 | 0.9365 | 0.9333 | 0.9593 | 0.7259 | 0.6333 | 0.5968 | 0.6926 | 0.6952 | 0.5593 | 0.7333 |

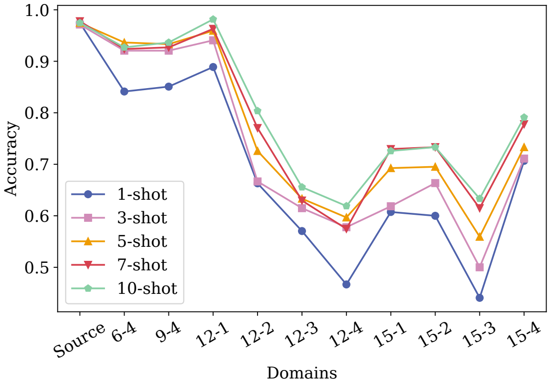

**Figure 2: Results for our proposed cross-domain UAV fingerprinting under different k-shot settings.**

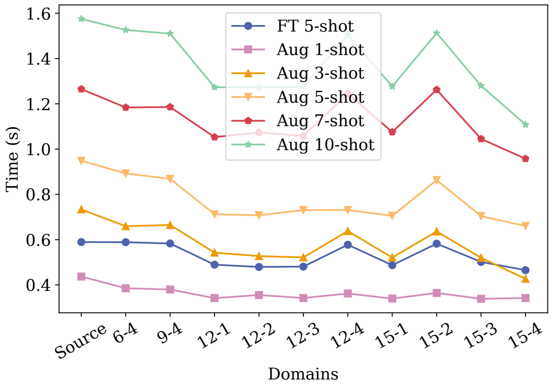

**Figure 3: Time required for fine-tuning and data augmentation. 'Aug k-shot' indicates k support samples used for fine-tuning with augmentation.**

In our FSL experiments, we use 5 examples from the support set to learn about unseen domains and 15 examples from the query set for prediction. While the vanilla PTN method slightly outperformed the CNN approach overall, it performed worse in the '9-4', '12-2', and '12-3' scenarios. These results highlight the limitations of vanilla PTN, despite its ability to incorporate some target domain information from a small number of examples.

To address these limitations, we apply fine-tuning, which increases classification accuracy for all cross-domain cases. The improvement is particularly significant for the '15-2' and '15-3' cases, where accuracy increased by about 20%. Furthermore, applying our designed data augmentation technique further improves classification accuracy across all cases. The most notable improvement is observed for '15-4', which has both different time and distance domains than the source domains, increasing accuracy by about 4% compared to fine-tuning alone and 18% compared to the CNN approach.

Fig. 2 presents the classification accuracy of our proposed FSL and data augmentation-aided UAV fingerprinting method under various k-shot settings. Fine-tuning and augmentation with 1-shot have limited effectiveness due to the constraints of using only a single support set sample. Nevertheless, they still outperform the vanilla PTN in 5-shot settings. Generally, increasing the number of support samples improves classification accuracy. However, this improvement is not linear. For instance, the accuracy gain from 7-shot to 10-shot is minimal, whereas the increase from 1-shot to 3-shot is substantial. This non-linear improvement demonstrates that our method achieves effective results without requiring a large number of domain samples, making it a lightweight solution. We will further validate this efficiency in the following subsection.

## 4.4 Deployment Overhead

For UAV fingerprinting, extensive fine-tuning on unseen domains is impractical due to time constraints, highlighting the need for a lightweight cross-domain solution. Fig. 3 shows the time required to fine-tune 5 support samples over 10 epochs and predict 15 query samples across various scenarios. In the case of 1-shot PTN with fine-tuning and data augmentation, the process takes around 0.35 seconds, making it practical for unseen domain adaptation. Notably, vanilla fine-tuning for the 5-shot case consumes roughly the same amount of time as fine-tuning with data augmentation in the 3-shot case. The most time-consuming scenario is the 10-shot PTN, averaging about 1.37 seconds. This overhead is

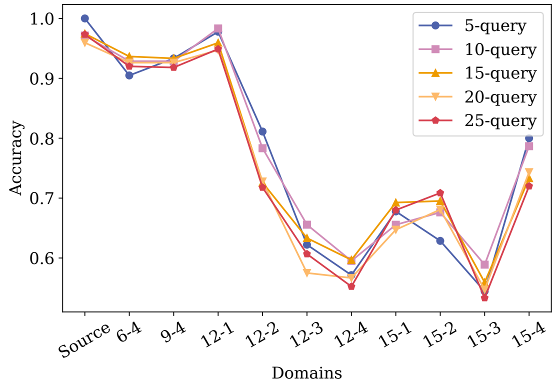

**Figure 4: Stability analysis under different numbers of query samples.**

considered acceptable since the process involves 10 epochs of fine-tuning and 15 sample predictions.

Combining these time costs and previous k-shot results, we conclude that deploying 10-shot learning is unnecessary, as it only shows a minimal increase in accuracy compared to 5-shot learning while being about 0.6 seconds slower. Our method thus demonstrates a significant increase in cross-domain accuracy without incurring large overhead, striking an optimal balance between performance and efficiency.

## 4.5 Stability Evaluation

Fig. 4 presents the performance of our proposed method under varying numbers of query samples, ranging from 5 to 25. The results demonstrate a remarkably consistent performance across this range, with only minimal variations observed as the number of query samples increases. This tight performance bound across different query settings is a strong indicator of our method's stability and robustness in the context of cross-domain UAV fingerprinting.

## 5 Conclusion

This paper proposes a lightweight cross-domain UAV fingerprinting method to mitigate domain shift issues. To achieve this, we first modify PTN to generate stable fingerprint features. Then, we devise specific data augmentation and apply fine-tuning to further improve performance on the unseen domains. Comprehensive experimental results validate our method's effectiveness in mitigating domain shift, demonstrating robust UAV identification across varied conditions.

## Acknowledgments

This work is supported in part by the NSF (CNS-2415209, CNS-2105416, IIS-2306791, CNS-2107164, CNS-2321763, CNS-2317190, and CNS-2319343).

## References

[1] Zhenxin Cai, Yu Wang, Qi Jiang, Guan Gui, and Jin Sha. 2024. Toward Intelligent Lightweight and Efficient UAV Identification With RF Fingerprinting. *IEEE Internet of Things Journal* (2024).

[2] Azade Fotouhi, Haoran Qiang, Ming Ding, Mahbub Hassan, Lorenzo Galati Giordano, Adrian Garcia-Rodriguez, and Jinhong Yuan. 2019. Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges. *IEEE Communications surveys & tutorials* 21, 4 (2019), 3417–3442.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[4] Wei Nie, Zhi-Chao Han, Mu Zhou, Liang-Bo Xie, and Qing Jiang. 2021. UAV detection and identification based on WiFi signal and RF fingerprint. *IEEE Sensors Journal* 21, 12 (2021), 13540–13550.

[5] Kunal Sankhe, Mauro Belgiovine, Fan Zhou, Shamnaz Riyaz, Stratis Ioannidis, and Kaushik Chowdhury. 2019. ORACLE: Optimized radio classification through convolutional neural networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 370–378.

[6] Guanxiong Shen, Junqing Zhang, Alan Marshall, Linning Peng, and Xianbin Wang. 2021. Radio frequency fingerprint identification for LoRa using spectrogram and CNN. In *Proc. IEEE Conf. Computer Communications (INFOCOM)*. IEEE, 1–10.

[7] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017).

[8] Nasim Soltani, Guillem Reus-Muns, Batool Salehi, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury. 2020. RF fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms. *IEEE transactions on vehicular technology* 69, 12 (2020), 15518–15531.

[9] Naeimeh Soltanieh, Yaser Norouzi, Yang Yang, and Nemai Chandra Karmakar. 2020. A review of radio frequency fingerprinting techniques. *IEEE J. Radio Freq. Identification* 4, 3 (2020), 222–233.

[10] Guomin Sun. 2021. RF transmitter identification using combined Siamese networks. *IEEE Transactions on Instrumentation and Measurement* 71 (2021), 1–13.

[11] Qiao Tian, Yun Lin, Xinghao Guo, Jinming Wen, Yi Fang, Jonathan Rodriguez, and Shahid Mumtaz. 2019. New security mechanisms of high-reliability IoT communication based on radio frequency fingerprint. *IEEE Internet of Things Journal* 6, 5 (2019), 7980–7987.

[12] Ningning Wang, Tianya Zhao, Shiwen Mao, and Xuyu Wang. 2024. AI Generated Wireless Data for Enhanced Satellite Device Fingerprinting. In *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 88–93.

[13] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)* 53, 3 (2020), 1–34.

[14] Caidan Zhao, Caiyun Chen, Zhibiao Cai, Mingxian Shi, Xiaojiang Du, and Mohsen Guizani. 2018. Classification of small UAVs based on auxiliary classifier Wasserstein GANs. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 206–212.

[15] Tianya Zhao, Xuyu Wang, and Shiwen Mao. 2024. Cross-domain, Scalable, and Interpretable RF Device Fingerprinting. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 2099–2108.

[16] Tianya Zhao, Xuyu Wang, Junqing Zhang, and Shiwen Mao. 2024. Explanation-guided backdoor attacks on model-agnostic rf fingerprinting. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 221–230.