GenID: A Generalizable Physical-Layer Device Identification Method for Out-of-Distribution Environments

Yawei Zhang[1] , **Lanting Fang**[1*] , **Di Yao**[2] , **Kaiyu Feng**[1] and **Shuliang Wang**[1]
[1]Beijing Institue of Technology
[2]Institute of Computing Technology, Chinese Academy of Sciences
z15334025555@163.com, ltfang@bit.edu.cn, yaodi@ict.ac.cn, {fengky, slwang2011}@bit.edu.cn

Abstract

Physical-layer device identification serves as a critical security mechanism for mitigating cyberattacks and enhancing network resilience. However, its deployment in modern full-duplex Ethernet networks faces two major challenges: (1) the presence of Out-of-Distribution (OOD) data caused by temporal distribution shift and diverse local transmitters, and (2) the difficulty of environment-agnostic device identification, where the unique features of target devices are often obscured in mixed signals. To address these challenges, we propose GenID, which consists of two core components. First, GenID constructs prototypes to emulate the full communication environment through a two stage process: offline and online. Based on the prototype context, it then introduces an invariant feature extraction method to disentangle mixed signals and improve generalization under OOD conditions. Extensive experiments demonstrate that GenID significantly outperforms state-of-the-art baselines, achieving an increase of at least 7.94% in Macro-F1 score under OOD environments. Our code is available on https://github.com/zyw2004/GenID_IJCAI2026.

1 Introduction

The widespread adoption of the internet have significantly increased reliance on Ethernet networks. However, this increased connectivity also raises the risk of cyberattacks and security threats [Sommer *et al.*, 2010], such as MAC address spoofing, firmware tampering, and switch port hijacking. To mitigate these risks, physical-layer device identification (PLDI), often referred to as Radio frequency (RF) fingerprinting [Zhang *et al.*, 2016; Brik *et al.*, 2008; Polak *et al.*, 2011], has emerged as an effective security solution. PLDI aims to learn a device-unique fingerprint by utilizing the communication signal data that implies physically unclonable characteristics [Gerdes *et al.*, 2012; Carbino *et al.*, 2015; Liu *et al.*, 2022]. Early PLDI methods relied on handcrafted features [Hall *et al.*, 2004; Hippenstiel and Payal, 1996]. In contrast, recent advancements leverage deep neural networks (DNNs) [Peng *et al.*, 2019; Al-Shawabka *et al.*, 2025] to learn discriminative representations directly from raw signal data. These DNN-based approaches automatically extract device-specific features, enabling their use in more complex communication environments.

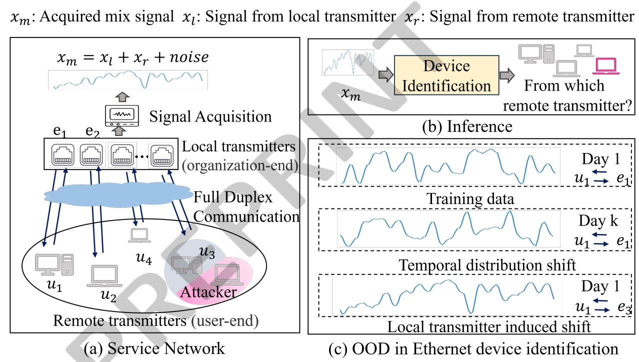


Figure 1: Device Identification in Full Duplex System

Most of existing PLDI methods are designed for half-duplex communication systems [Mick, 2000], where signals transmit in only one direction at a time. This design severely limits their applicability to full-duplex communication, which overwhelmingly adopted in modern Ethernet infrastructures [George *et al.*, 2022]. As Figure 1 (a) shows, full-duplex communication support simultaneous bidirectional transmission over a single physical medium. In such systems, the acquired signal x_m comprises a mix of the local transmitter's signal x_l, the remote peer's signal x_r, and environmental noise. We will hereafter refer to remote peer as the **device** for better distinguishing between these two signal sources. In this setting, the device's signal is entangled with other components in x_m (Figure 1 (b)), posing unique challenges for device identification.

Specifically, the challenges of identifying devices in full-duplex system are two-fold: **(1) Out-of-Distribution (OOD) environments.** Existing PLDI models are highly susceptible to dynamic environmental shifts [Shen *et al.*, 2021; Reising *et al.*, 2020; He and Wang, 2020] including *temporal distribution shift* and *local transmitter induced shift*, as shown in Figure 1 (c). Temporal shifts arise when varia-

*Lanting Fang is the corresponding author.

tions in ambient noise, temperature, or port configurations cause the same device-transmitter pair to produce different signals across days (e.g., Day 1 vs. Day k). Transmitter-induced shifts occur when the test-time transmitter differs from the one used during training (e.g., training with e_1, testing with e_3). While OOD generalization has been studied in domains like voice detection [Schroter *et al.*, 2022; Zhao *et al.*, 2022] and time series analysis [Lu *et al.*, 2022; Wang *et al.*, 2024], these methods do not transfer well to PLDI. This is primarily due to the nature of physical-layer signals: the device-specific features reside in extremely weak physical-layer characteristics that are easily overwhelmed by the dominant transmitter signal and ambient noise. Existing OOD methods may overlook these subtle features, leading to poor performances under distribution shifts. **(2) Environment-agnostic device identification.** In practice, the PLDI methods need to identify devices based on mixed signals, which introduces a major challenge: How to isolate environment-agnostic features from signals entangled by local transmitter and background noise. Existing environment-agnostic representation methods often adopt basic encoder architectures, such as CNNs [LeCun *et al.*, 2002; Shi *et al.*, 2024] or Transformers [Vaswani *et al.*, 2017], to extract invariant and relevant feature representations. They combine task-specific learning strategies in model design to encourage feature disentanglement and are limit in environment-agnostic device feature extraction. For example, CNNs lack explicit mechanisms for separating mixed signal, and Transformers often overemphasize shared features, which hinders the extraction of device-specific information.

To address the aforementioned challenges, we propose GenID (GENeralizable Identification of Devices), which is the first work for PLDI under full-duplex and OOD settings. To master the varying environments and address the OOD problem, GenID introduces a set of anchor devices and constructs an offline feature dictionary that captures the unique signatures of local transmitters. When a device connects, GenID generates device-specific prototypes by integrating this dictionary with the incoming mixed signals based on a novel data augmentation method. These prototypes simulate how the device would perform with all local transmitters under diverse environments. Moreover, to extract environment-agnostic features, GenID leverages augmented mixed signals as input to the Invariant Feature Extraction (IFE) network. The IFE network employs an attention mechanism to assess the relevance of each feature dimension between the input signals and both inter- and intra-class prototypes. This process enables the network to distill the most salient and invariant device features, thereby ensuring reliable identification under dynamic environmental conditions.

- We propose GenID to achieve accurate device identification in full-duplex communication and OOD settings. To the best of our knowledge, GenID is the first PLDI method designed for full-duplex devices with OOD generalization.
- We design a novel mixed-signal augmentation technique that generates device-specific prototypes, simulating a variety of temporal and local transmitter conditions. Furthermore, we propose an Invariant Feature Extraction (IFE) network that leverages these prototypes to learn robust and discriminative device representations.
- Extensive experiments are conducted to verify the effectiveness and efficiency of GenID. The results show that GenID outperforms the state-of-the-art baselines with over 7.94% improvements of Macro-F1.

2 Related Work

2.1 Physical-Layer Device Identification

Device authentication methods can be broadly categorized into manual feature-based and deep learning-based approaches. Feature-based methods extract hand-crafted features such as signal amplitude and phase, followed by classification techniques like Euclidean distance [Hall *et al.*, 2004; Hippenstiel and Payal, 1996] or traditional classifiers such as SVM [Reising *et al.*, 2020; Carbino *et al.*, 2015; Huang *et al.*, 2017]. These approaches are lightweight but heavily depend on domain expertise and are often sensitive to environmental noise. In contrast, most deep learning-based approaches identify devices by inputting raw or minimally processed signals into deep neural networks. Peng et al.[Peng *et al.*, 2021] transform I/Q samples into 2D representations and classify them using the InceptionV3 architecture. Shen et al. [Shen *et al.*, 2023] apply short-time Fourier transform to obtain time–frequency representations and then feed them into a Transformer. However, these methods are primarily designed for in-distribution settings and often fail to generalize under OOD conditions.

2.2 OOD Generalization

OOD Generalization in PLDI. In PLDI, OOD challenges often arise from dynamic environmental shifts that distort signal characteristics. However, few studies explicitly address this issue. Existing methods typically rely on limited target-domain data to adapt models trained on a source domain [Al-Shawabka *et al.*, 2025; Han *et al.*, 2023; Peng *et al.*, 2024]. For example, SignCRF [Al-Shawabka *et al.*, 2025] uses adversarial training with an environment translator to remove channel effects while preserving device-specific features.

In contrast, our approach seeks to achieve OOD generalization without any access to target-domain data.

General OOD Generalization Methods. OOD generalization [Wang *et al.*, 2022] approaches can be broadly categorized into three main strategies: data manipulation, learning strategies, and representation learning. Data manipulation techniques focus on increasing the diversity of the training data [Zhong *et al.*, 2025], thereby exposing the model to a wider range of variations that might be encountered during testing. Learning strategies aim to enhance generalization by leveraging general learning frameworks, including techniques like ensemble learning [Cha *et al.*, 2021] and meta-learning [Khoee *et al.*, 2024]. Among OOD generalization methods, representation learning is particularly prevalent. It primarily involves two strategies: domain invariant representation learning and feature disentanglement. Domain-invariant representation learning seeks features stable across domains using techniques like adversarial training [Fu *et al.*, 2023], feature alignment [Du *et al.*, 2021], and invariant risk minimization [Wang *et al.*, 2025]. Feature disentanglement aims to separate domain-shared from domain-specific components through methods such as causal inference [Li *et al.*, 2022] and generative modeling [Kim *et al.*, 2024].

Despite their success in other domains, general OOD generalization methods remain unexplored for physical-layer device identification. Physical-layer features are fine-grained, easily obscured by noise and transmitter interference, and often difficult to disentangle. As a result, these methods cannot be directly applied to this task.

3 Methodology

3.1 Problem Formulation

Definition 1 (Physical-Layer Device Identification)**.** *Given a mixed signal segment x, produced by the interaction between a device $u \in \mathcal{U}^u$ and a local transmitter $e \in \mathcal{E}$, physical-layer device identification aims to predict the device u responsible for generating x.*

Definition 2 (Temporal distribution shifts)**.** *The training signal segments $\mathcal{X}_{train}$ and the test signal segments $\mathcal{X}_{test}$ are collected under different communication environments, leading to temporal distribution shifts.*

Definition 3 (Local transmitter-induced shifts)**.** *The sets of local transmitters used for the training signal segments $\mathcal{X}_{train}$ and the test signal segments $\mathcal{X}_{test}$ differ, inducing distribution shifts due to transmitter variations.*

Problem Statement. The objective of this paper is to train a neural network capable of performing physical-layer device identification robustly under both temporal distribution shifts and local transmitter-induced shifts.

3.2 Overview of GenID

In this section, we introduce the proposed framework GenID and describe its key components. Collecting signals from devices under every possible communication condition is often impractical, and models trained solely on online data tend to perform poorly when the environment changes. To address this challenge, we adopt an offline–online training strategy.

During the offline phase, user devices are *inaccessible* or *uncontrollable*. Therefore, we use a set of anchor devices as proxies. These devices allow us to capture the communication characteristics of local transmitters when connected to diverse devices. At the end of this phase, we construct a dictionary of local transmitters, where each local transmitter is represented by multiple entries, and each entry corresponds to a feature extracted from a signal segment collected under a specific environment.

The online phase occurs during deployment, where signals are collected from only a single environment. The prototype generation module leverages these signals together with the offline dictionary to simulate mixed signals between the user device and different local transmitters under diverse environmental conditions. These simulated mixed signals are encoded and stored in a prototype memory bank. Subsequently, the network training module employs the Invariant Feature Extraction (IFE) network to learn robust and invariant representations from these prototypes, producing reliable embeddings for user devices. An overview of the proposed framework is shown in Figure 2 (a).

- **Dictionary Construction (Offline):** In this phase, each anchor device $u_i \in \mathcal{U}^o$ connects to every local transmitter $e_j \in \mathcal{E}$. For each device–transmitter pair (u_i, e_j), N signal samples are extracted, where N is a predefined hyperparameter. These samples are processed through a pre-processing pipeline to generate feature representations, which collectively form the dictionary $\mathcal{D}$.
- **Prototypes Generation (Online):** Each user device in $\mathcal{U}^u$ is connected to at least one available local transmitter from the set $\mathcal{E}$. From this connection, we collect signal samples $\mathcal{X}_{train}$ along with their corresponding labels $\mathcal{Y}_{train}$. We design a data augmentation method to construct a prototype memory bank $\mathcal{M}$ using $\mathcal{X}_{\text{train}}$, $\mathcal{Y}_{\text{train}}$, and the offline dictionary $\mathcal{D}$.
- **Network Training (Online):** To distill invariant device features, GenID employs an Invariant Feature Extraction (IFE) network to learn invariant representations of user devices by modeling both intra-class consistency and inter-class discrimination among their features.

3.3 Dictionary Construction (Offline)

The objective of this phase is to construct a dictionary $\mathcal{D}$ that captures local transmitter characteristics under varying environmental conditions. To this end, we leverage a set of anchor devices $\mathcal{U}^o = \{u_1, \dots, u_m\}$ as controlled signal sources and systematically connect each device to every local transmitter in $\mathcal{E} = \{e_1, \dots, e_n\}$ at different times. The process will be described in order below.

Signal Collection and Pre-processing. Each anchor device $u_i \in \mathcal{U}^o$ is systematically paired with every local transmitter $e_j \in \mathcal{E}$. For each device-transmitter pair (u_i, e_j), we acquired N mixed signal sample $\{x_{i,j}^1, \dots, x_{i,j}^N\}$, where $x_{i,j}^k$ denotes the k-th acquired signal sample for pair (u_i, e_j).

Each signal sample is passed through a pre-processing pipeline $Pre(\cdot)$. Following prior works [Levy *et al.*, 2023; Choi *et al.*, 2018; Gerdes *et al.*, 2012], the pipeline removes the direct current component and applies FFT-based spectral averaging to obtain a stable frequency-domain representation $Pre(x_{i,j}^k) \in \mathbb{R}^d$.

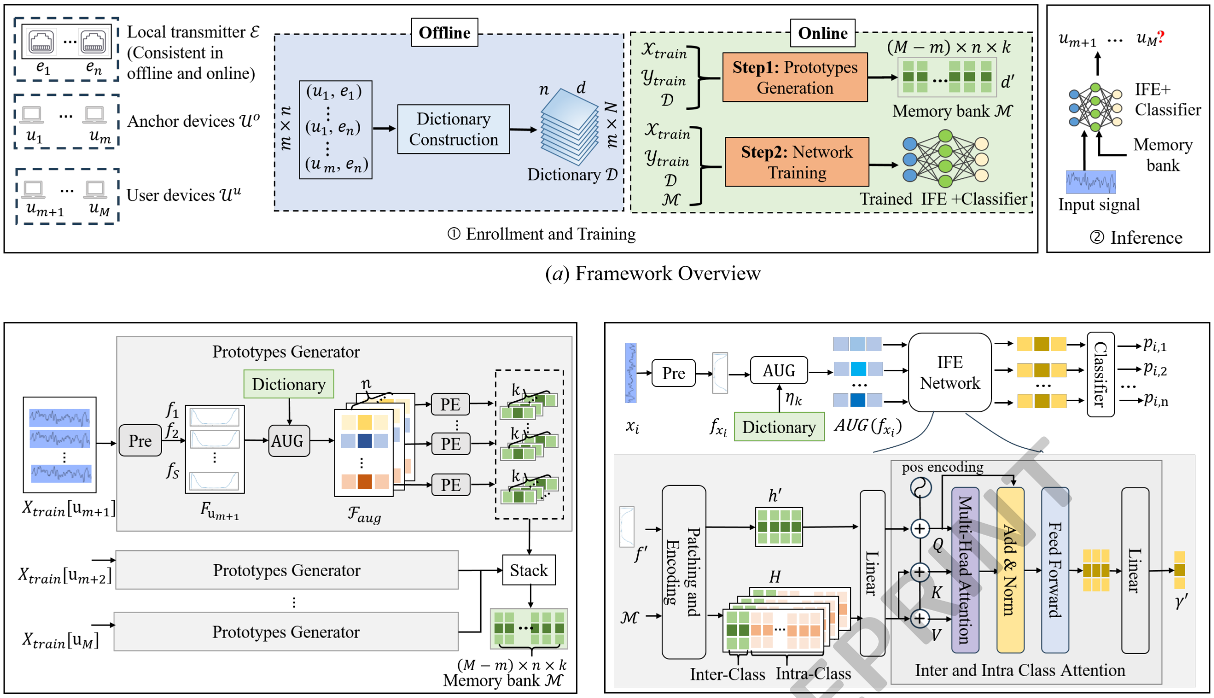
Dictionary Construction. After pre-processing, each device -transmitter pair (u_i, e_j) yields N feature vectors $\{Pre(x_{i,j}^1), \dots, Pre(x_{i,j}^N)\}$. We construct the dictionary $\mathcal{D}$ by collecting all feature units across the m anchor devices, each with N signal samples:

$$\mathcal{D} = \{\eta_{1,1}, \dots, \eta_{1,N}, \dots, \eta_{m,1}, \dots, \eta_{m,N}\}, \tag{1}$$

Each feature unit $\eta_{i,k}$ corresponds to one environment and is formed by concatenating the pre-processed signals from all local transmitters $\eta_{i,k} = [\text{Pre}(x_{i,1}^k); \dots; \text{Pre}(x_{i,n}^k)]$.

3.4 Prototypes Generation (Online)

At the online stage, each user device $u_i \in \mathcal{U}^u$ typically communicates with only a limited set of transmitters under re-



(a) Framework Overview

(b) Prototypes Generation

(c) Network Training

Figure 2: The framework of GenID. (a) provides an overview of GenID. The key building blocks, i.e., prototypes generation and network training are detailed in (b) and (c).

stricted temporal conditions. We leverage the offline dictionary $\mathcal{D}$ and a novel augmentation method to simulate the device's behavior across all transmitters across varying temporal conditions.

As shown in Figure 2 (b), we first partition the training data by device labels. Let $\mathcal{X}_{\text{train}}[u_i] = \{x'_1, \ldots, x'_S\}$ denote the training samples collected from a user device u_i. Each signal is processed by the pre-processing module $Pre(\cdot)$ to produce a feature set $F_{u_i} = \{Pre(x'_1), \ldots, Pre(x'_S)\}$. These features, along with the dictionary $\mathcal{D}$, are then feed into data augmentation.

Data Augmentation (AUG$(\cdot)$). For each preprocessed feature $f' \in F_{u_i}$ with label $y' = (u_i, e_k)$, we randomly sample a feature unit $\eta = [d_{e_1}, \ldots, d_{e_n}] \in \mathcal{D}$, which encodes the signatures of all local transmitters. We take $d_{\text{base}} = d_{e_k}$, representing the feature of the sampled anchor device when connected to transmitter e_k.

To simulate how the user device u_i would behave under other transmitters, we apply an augmentation function F_{aug} to pair f' with each $d_{e_j} \in \eta$, and compute the augmented features as:

$$F_{\text{aug}}(d_{e_j}, f') = f' - d_{\text{base}} + d_{e_j} \tag{2}$$

This function approximates the preprocessed feature that would be observed if device u_i were to communicate with transmitter e_j. Further proof is provided as follows.

Proof. The preprocessed feature f' can be expressed as:

$$f' = Pre(x_{u_i} + x_{e_k} + \text{noise}_1), \tag{3}$$

where x_{u_i} and x_{e_k} denotes the signal generated by u_i and e_k, respectively, and noise$_1$ represents noise. Since the preprocessing module Pre$(\cdot)$ removes the direct current component and applies FFT-based spectral averaging, both of which preserve linearity, we have:

$$f' = Pre(x_{u_i}) + Pre(x_{e_k}) + Pre(\text{noise}_1) \tag{4}$$

Similarly, $d_{e_j} = Pre(x_{u_i}) + Pre(x_{e_j}) + Pre(\text{noise}_2)$, and the base feature vector can be expressed as $d_{base} = Pre(s_{u_i}) + Pre(s_{e_k}) + Pre(\text{noise}_3)$.

According to the definition of mapping function $F_{\text{aug}}(d_{e_j}, f') = f' - d_{base} + d_{e_j}$, we have:

$$\begin{aligned} F_{\text{aug}}(d_{e_j}, f') &= (Pre(x_{u_i}) + Pre(x_{e_k}) + Pre(\text{noise}_1)) \\ &\quad - (Pre(x_{u_i}) + Pre(x_{e_k}) + Pre(\text{noise}_2)) \\ &\quad + (Pre(x_{u_i}) + Pre(x_{e_j}) + Pre(\text{noise}_3)) \\ &= Pre(x_{u_i}) + Pre(x_{e_j}) + \text{noise}_4 \\ &= Pre(x_{u_i} + x_{e_j} + \text{noise}_5), \end{aligned} \tag{5}$$

where noise$_4 = Pre(\text{noise}_1) - Pre(\text{noise}_2) + Pre(\text{noise}_3)$ and noise$_5 = Pre(\text{noise}_4)$. □

Memory Bank Construction. We construct the full augmented feature set as follows:

$$\mathcal{F}_{\text{aug}} = \begin{bmatrix} F_{\text{aug}}(d_{e_1}, Pre(x'_1)), & \ldots, & F_{\text{aug}}(d_{e_{|\mathcal{E}|}}, Pre(x'_1)) \\ F_{\text{aug}}(d_{e_1}, Pre(x'_2)), & \ldots, & F_{\text{aug}}(d_{e_{|\mathcal{E}|}}, Pre(x'_2)) \\ & \cdots & \\ F_{\text{aug}}(d_{e_1}, Pre(x'_S)), & \ldots, & F_{\text{aug}}(d_{e_{|\mathcal{E}|}}, Pre(x'_S)) \end{bmatrix} \tag{6}$$

where $\mathcal{F}_{\text{aug}}[s][j]$ represents the feature of the s-th simulated mixed signal from (u_i, e_j) pair. To extract prototypes for each (u_i, e_j) pair, we first aggregate the features corresponding to transmitter e_j across all S augmented samples:

$$\mathcal{F}(u_i, e_j) = \{\mathcal{F}_{\text{aug}}[1][j], \ldots, \mathcal{F}_{\text{aug}}[S][j]\} \tag{7}$$

We then apply a Prototype Extraction (PE) module to derive k prototypes, each capturing a representative environmental condition. Specifically, we perform K-Means clustering with k clusters on $\mathcal{F}(u_i, e_j)$, and use the resulting cluster centroids as the prototypes.

Prototypes generated from $M-m$ user devices and n local transmitters are aggregated into the memory bank $\mathcal{M}$, resulting in a total of $(M-m) \times n \times k$ prototypes.

3.5 Network Training (Online)

This step aims to optimize the Invariant Feature Extraction (IFE) network to produce robust features that are invariant to OOD environments. As illustrated in Figure 2 (c), each input signal x_i is first processed by the pre-processing module (detailed in Section 3.3) and the augmentation strategy (detailed in Section 3.4) to increase input diversity and simulate variations in temporal and local transmitter induced environments. The resulting augmented features $\text{AUG}(f_{x_i}) = [F_{\text{aug}}(d_{e_1}, f_{x_i}), \ldots, F_{\text{aug}}(d_{e_{|\mathcal{E}|}}, f_{x_i})]$ are then fed into the IFE network. The IFE network consists of two main modules: Patching and Encoding, and Inter- and Intra-Class Attention.

Patching and Encoding. For each feature vector $f' \in \text{AUG}(f_{x_i})$, the IFE network divides it into s overlapping patches with a fixed size and 50% overlap, yielding $\{P_1, P_2, \ldots, P_s\}$. Each patch, as well as the original feature, is independently projected by linear layers. The resulting embeddings are concatenated to form a joint representation $h' \in \mathbb{R}^{(s+1)\times l}$:

$$h' = \text{Stack}(\text{Linear}_1(f'), \text{Linear}_2(P_1), ..., \text{Linear}_2(P_s)). \tag{8}$$

The same encoding process is applied to all prototypes in the memory bank $\mathcal{M}$, producing a prototype representation tensor $H \in \mathbb{R}^{|\mathcal{M}|\times(s+1)\times l}$, where $|\mathcal{M}| = (M-m) \times n \times k$ is the number of prototypes.

Inter- and Intra- Class Attention. The attention mechanism enables the network to selectively focus on the most relevant parts of the input for prediction, assigning varying importance (attention weights) to different components and thus emphasizing salient features while suppressing less informative ones [Vaswani *et al.*, 2017]. To distill invariant device features, we design an attention mechanism that jointly models relationships within the same device class (intra-class) and across different device classes (inter-class).

The attention computation involves three key components: the query $Q = \text{Linear}(h_i^T) \in \mathbb{R}^{l\times d}$, and the key and value $K = V = \text{Linear}(H^T) \in \mathbb{R}^{|\mathcal{M}|\times l\times d}$. To incorporate spatial or sequential context, a learnable positional encoding h_{pos} is added:

$$Q \leftarrow Q + h_{pos}, \quad K \leftarrow K + h_{pos}, \quad V \leftarrow V + h_{pos}. \tag{9}$$

Multi-head attention is then applied, followed by a residual connection, layer normalization and a feedforward network:

$$h'_o = \text{FFN}\left(\text{Norm}\left(\text{MultiHeadAttention}(Q, K, V) + Q\right)\right). \tag{10}$$

where $h'_o \in \mathbb{R}^{l\times d}$, with l denoting the mapping dimension and d the dimensionality of the hidden representation.

Finally, the invariant feature vector is computed as:

$$\gamma' = h_o'^T \cdot W + b, \tag{11}$$

where $W \in \mathbb{R}^{d\times 1}$ and $b \in \mathbb{R}^l$. By jointly modeling intra- and inter-class interactions, this mechanism enhances the model's ability to distinguish subtle yet critical variations among devices.

Classifier. Each feature vector $f' \in \text{AUG}(f_{x_i})$ is processed by the IFE network to produce an invariant representation γ'. Applying the IFE to all augmented samples of x_i yields a set of representations $\gamma_{i,1}, \ldots, \gamma_{i,n}$, which are then passed into the classifier consisting of normalization and linear projection:

$$p_{i,j} = \text{Linear}(\text{Norm}(\gamma_{i,j})). \tag{12}$$

where $p_{i,j} \in \mathbb{R}^{M-m}$ represents the predicted probability distribution over the $M-m$ device categories for the j-th augmented sample of x_i.

We train the IFE network and classifier using a categorical cross-entropy loss with label smoothing, which mitigates overconfidence and improves generalization:

$$\mathcal{L} = -\sum_{x_i \in \mathcal{X}_{train}} \sum_{j=1}^{n} \sum_{k=1}^{M-m} y_i^{\text{smooth}}[k] \cdot \log(p_{i,j}[k]). \tag{13}$$

The smoothed label y_{smooth} is calculated as:

$$y_i^{smooth}[k] = (1-\epsilon) \cdot y_i[k] + \frac{\epsilon}{M-m}, \tag{14}$$

where y_i is the one-hot vector of the device label, ϵ is the label smoothing coefficient.

3.6 Inference

This section explains how the trained GenID model classifies new, unseen samples. As illustrated in Figure 2 (a), the Inference module processes each test sample $x_i \in \mathcal{X}_{\text{test}}$ as follows: First, each input x_i is processed by the pre-processing module. The resulting feature f_i is then passed into the IFE network along with the memory bank $\mathcal{M}$ to generate an invariant representation $\hat{\gamma}_i$. Finally, the trained classifier outputs the predicted class probabilities:

$$\hat{p}_i = \text{Classifier}(\text{IFE}(\text{Pre}(x_i), \mathcal{M})) \tag{15}$$

The predicted label $\hat{y}_i$ is determined by selecting the class with the highest probability.

Inference Time Complexity. During inference, GenID consists of: (1) FFT-based preprocessing with complexity $\mathcal{O}(d \log d)$; (2) patch-based linear encoding with complexity $\mathcal{O}(sd^2)$, where s is the number of patches and d is both the input and embedding dimension; (3) multi-head attention over P prototypes, yielding complexity $\mathcal{O}(sdP)$; and (4) final classification with complexity $\mathcal{O}(dC)$ over C classes. The overall inference time complexity per sample is: $\mathcal{O}(d \log d + sd^2 + sdP + dC)$. Neglecting lower-order terms, we further simplify to: $\mathcal{O}(sd(d+P))$.

4 Experiments

4.1 Experimental Settings

Datasets. We collected data over many days from all possible combinations of 4 local transmitters, 4 user devices, and 6 anchor devices. For each target device and local transmitter combination, we gathered some data points each day. This resulted in a grand total of 2.52×10^{10} data points across the entire collection period.

Evaluation Scenarios. We pre-process the acquired mixed signal data by segmenting it into fixed lengths of 7×10^4 for online data and 7×10^5 for offline data, yielding 1,000 online samples and 100 offline samples per day for each device–local transmitter pair, respectively. A Fast Fourier Transform (FFT) with length 128 is applied to each segment. The online dataset is then randomly split into training, validation, and testing sets in a 7:1:2 ratio. To assess GenID's robustness, we evaluated it under four distinct settings:

- **Exp.(a) Temporal + Local transmitter induced OOD:** 15,000 samples where both acquisition days and local transmitters differ between training and testing sets.
- **Exp.(b) Temporal OOD:** 7,500 samples where local transmitters remain fixed but acquisition days differ.
- **Exp.(c) Local transmitter induced OOD:** 7,500 samples with consistent acquisition days but varying local transmitters.
- **Exp.(d) i.i.d.:** 20,000 samples where both acquisition days and local transmitters are shared across training and testing.

Evaluation Metrics. We assessed the performance of the GenID model using three key metrics: micro F1 score, macro F1 score, and the Kappa coefficient (KC) [Vieira *et al.*, 2010].

Baselines. We evaluate GenID against four categories of baseline methods on the same preprocessed signal representations produced by our preprocessing pipeline for fair comparison: (1) Physical-layer device identification methods, including LoRa [Shen *et al.*, 2021], RivIs [Reising *et al.*, 2020], and Csei [He and Wang, 2020]. Due to protocol differences that prevent direct application to the 1000Base-T Ethernet setting, we adapt their classification backbones to the preprocessed fingerprints. (2) OOD generalization methods for classification, including RBS [Song *et al.*, 2020], RandConv [Xu *et al.*, 2020], and MixStyle [Zhou *et al.*, 2023]. These methods are combined with our classifier to assess robustness under distribution shifts. (3) OOD-aware tabular and time-series classifiers,including TabPFN [Hollmann *et al.*, 2022] and ITSR [Shi *et al.*, 2024]. (4) State-of-the-art time series classification methods, such as CAFO [Kim *et al.*, 2024] and TSLANet [Eldele *et al.*, 2024]. These two are evaluated using their open-source implementations on preprocessed data.

Hyper-Parameters. We set the PE module with $k = 4$ clusters. In the IFE network, we set the patch size to 16. The output dimensions of the patching-encoding module and the linear layer are 128 and 16, respectively. Multi-head attention is configured with a single layer and 4 heads. The feedforward network consists of two linear layers with intermediate dimensions of 64. All Dropout ratios were set to 0.1.

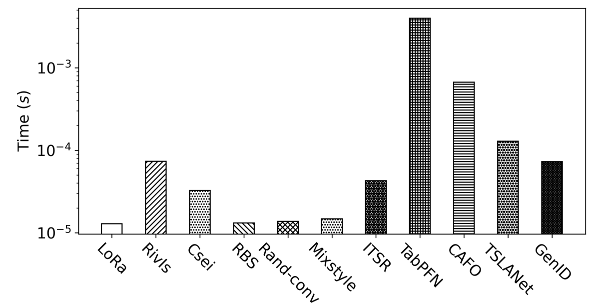


Figure 3: Efficiency

For optimization, we used AdamW with a learning rate of $1 \times e^{-3}$ and a weight decay of $1 \times e^{-4}$. Training runs for up to 100 iterations, with early stopping applied if the validation loss does not improve for 30 epochs.

4.2 Comparison With SOTA Methods

(1) Does GenID outperform SOTA methods in terms of device classification performance? Table 1 presents the comparative results of GenID against a comprehensive set of SOTA baselines across four experimental settings Exps.(a–d), evaluated using micro-F1, macro-F1, and KC. GenID consistently outperforms all competing methods in almost all metrics and experimental settings,

While existing physical-layer identification methods perform comparably to GenID in the i.i.d. setting Exp.(d), their performance drops sharply in the OOD scenarios Exps.(a–c), indicating limited robustness to temporal and transmitter-induced distribution shifts. General OOD classification methods perform even worse, as the fine-grained physical-layer features are easily masked by noise and interference, making them difficult to disentangle. Tabular OOD method TabPFN and Time-series models like CAFO and TSLANet show good performance under specific OOD conditions but fail to generalize across all scenarios

Across all methods, performance in Exp.(d) is consistently higher than in Exp.(a–c), underscoring the challenge of distribution shifts. Notably, GenID demonstrates significantly smaller performance degradation, showcasing its robustness to both temporal and transmitter variations.

(2) How does GenID compare to SOTA methods in terms of generalization under OOD scenarios? We further evaluate GenID under varying device partitions using Exp.(a), the most challenging OOD setting. Specifically, we generate three alternative random partitions of online and anchor devices while keeping all other factors unchanged. As shown in Table 2, performance varies across partitions due to inherent differences in device distinguishability. Despite this variability, GGenID consistently outperforms all other configurations, achieving an average Macro-F1 improvement of 7.94% over the strongest competing baseline, thereby demonstrating robustness to device partitioning.

4.3 Efficiency

Figure 3 presents the per-sample inference time for all comparison methods. As expected, LoRa, using a traditional

	Exp.(a)			Exp.(b)			Exp.(c)			Exp.(d)		
Method	micro-F1 %	macro-F1 %	KC %	micro-F1 %	macro-F1 %	KC %	micro-F1 %	macro-F1 %	KC %	micro-F1 %	macro-F1 %	KC %
LoRa	64.72 ± 0.87	63.58 ± 1.19	52.97 ± 1.15	72.88 ± 0.89	71.39 ± 1.22	63.82 ± 1.18	49.63 ± 1.88	44.88 ± 2.59	33.62 ± 2.40	86.79 ± 0.20	86.78 ± 0.20	82.39 ± 0.27
RivIs	71.73 ± 0.00	70.09 ± 0.00	62.31 ± 0.00	78.13 ± 0.00	77.36 ± 0.00	70.83 ± 0.00	58.00 ± 0.00	55.07 ± 0.00	44.72 ± 0.00	90.90 ± 0.00	90.89 ± 0.00	87.86 ± 0.00
Csei	53.45 ± 1.28	47.28 ± 1.36	37.97 ± 1.70	66.16 ± 2.23	63.92 ± 2.78	54.85 ± 2.98	57.31 ± 4.34	55.35 ± 3.90	43.46 ± 5.65	86.71 ± 0.37	86.69 ± 0.37	82.28 ± 0.49
RBS	49.88 ± 0.67	48.73 ± 0.82	33.22 ± 0.90	57.15 ± 1.89	56.71 ± 2.21	42.86 ± 2.53	58.72 ± 1.59	58.44 ± 1.45	44.95 ± 2.12	50.24 ± 3.33	48.91 ± 4.14	34.24 ± 4.12
Rand-conv	64.67 ± 0.94	64.03 ± 1.13	52.91 ± 1.25	71.52 ± 1.70	69.87 ± 2.32	62.00 ± 2.27	48.59 ± 0.89	43.45 ± 1.32	32.27 ± 1.12	85.52 ± 0.56	85.50 ± 0.55	80.69 ± 0.74
Mixstyle	64.91 ± 0.80	63.92 ± 1.01	53.22 ± 1.07	73.52 ± 2.09	72.20 ± 2.83	64.68 ± 2.78	50.77 ± 1.26	46.33 ± 1.76	35.07 ± 1.61	86.73 ± 0.40	86.72 ± 0.39	82.30 ± 0.52
ITSR	46.04 ± 2.97	44.86 ± 3.73	28.06 ± 3.93	41.25 ± 12.58	33.09 ± 19.23	21.63 ± 16.77	46.21 ± 4.42	42.33 ± 5.37	28.41 ± 5.61	75.50 ± 4.11	75.41 ± 4.17	67.37 ± 5.44
TabPFN	61.63 ± 1.54	56.44 ± 1.89	48.81 ± 2.04	75.89 ± 0.71	75.02 ± 0.89	67.85 ± 0.95	$\underline{66.32 \pm 12.41}$	$\underline{64.78 \pm 13.30}$	$\underline{55.55 \pm 16.29}$	$\underline{91.54 \pm 0.27}$	$\underline{91.53 \pm 0.27}$	$\underline{88.71 \pm 0.36}$
CAFO	60.36 ± 2.50	57.53 ± 4.10	47.22 ± 3.29	$\underline{81.44 \pm 1.46}$	$\underline{81.27 \pm 1.53}$	$\underline{75.25 \pm 1.94}$	54.11 ± 3.62	50.29 ± 4.08	39.44 ± 4.59	85.51 ± 3.42	85.27 ± 3.71	80.66 ± 4.55
TSLANet	$\underline{75.71 \pm 1.63}$	$\underline{75.58 \pm 1.70}$	$\underline{67.63 \pm 2.17}$	80.88 ± 1.13	80.59 ± 1.27	74.50 ± 1.50	57.44 ± 2.34	54.02 ± 3.48	43.99 ± 2.96	$\mathbf{91.59 \pm 0.41}$	$\mathbf{91.58 \pm 0.41}$	$\mathbf{88.78 \pm 0.54}$
GenID	$\mathbf{84.83 \pm 0.66}$	$\mathbf{84.79 \pm 0.69}$	$\mathbf{79.76 \pm 0.88}$	$\mathbf{85.32 \pm 0.27}$	$\mathbf{85.22 \pm 0.23}$	$\mathbf{80.42 \pm 0.36}$	$\mathbf{69.36 \pm 1.13}$	$\mathbf{66.06 \pm 1.82}$	$\mathbf{59.65 \pm 1.44}$	91.09 ± 0.56	91.08 ± 0.56	88.12 ± 0.74

Table 1: Comparison with SOTA methods. The highest scores are highlighted in bold, and the second highest scores are highlighted in underline.

Method	Exp.(a)	Exp.(a2)	Exp.(a3)	Exp.(a4)
LoRa	63.58 ± 1.19	65.33 ± 0.67	70.30 ± 0.77	51.47 ± 0.99
RivIs	70.09 ± 0.00	67.11 ± 0.00	$\underline{74.37 \pm 0.00}$	60.74 ± 0.00
Csei	47.28 ± 1.36	66.40 ± 1.43	54.50 ± 7.15	$\underline{64.02 \pm 3.94}$
RBS	48.73 ± 0.82	59.43 ± 0.92	66.24 ± 1.33	45.96 ± 0.91
Rand-conv	64.03 ± 1.13	65.21 ± 0.65	72.10 ± 0.90	52.32 ± 1.61
Mixstyle	63.92 ± 1.01	65.70 ± 0.57	70.32 ± 0.32	51.87 ± 0.79
ITSR	44.86 ± 3.73	56.54 ± 5.91	62.04 ± 3.57	42.87 ± 1.73
TabPFN	56.44 ± 1.89	66.51 ± 0.47	68.45 ± 1.79	57.47 ± 0.81
CAFO	57.53 ± 4.10	67.29 ± 1.91	59.55 ± 10.03	54.59 ± 2.69
TSLANet	$\underline{75.58 \pm 1.70}$	$\underline{70.05 \pm 2.92}$	72.54 ± 0.50	55.66 ± 0.94
GenID	$\mathbf{84.79 \pm 0.69}$	$\mathbf{81.11 \pm 1.07}$	$\mathbf{81.83 \pm 0.61}$	$\mathbf{70.52 \pm 1.43}$

Table 2: Robustness evaluation with different device partition.

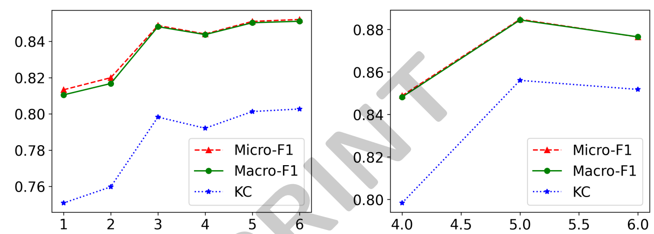
Methods	Exp.(a)	Exp.(b)	Exp.(c)	Exp.(d)
w/o AUG(·) in training	62.83	79.33	54.21	92.24
w/o AUG(·) in prototypes	63.84	80.77	54.24	91.65
w/o dictionary	62.91	79.68	54.96	91.80
w/o patching	83.27	80.00	65.76	89.21
w/ self-attention	60.41	46.80	40.03	24.67
GenID	84.83	85.32	69.36	91.09

Table 3: Ablation Study

architecture, achieves the fastest processing speed. GOOD classification methods incur only a slight efficiency overhead, maintaining high efficiency. In contrast, more complex deep learning methods like TabPFN, CAFO, and TSLANet exhibit a significant drop in efficiency, with processing times increasing by one or two orders of magnitude. GenID, however, strikes a favorable balance, maintaining competitive speed while delivering strong performance, staying on par with traditional approaches in terms of inference time.

4.4 Ablation Study

Table 3 presents the ablation study evaluating the contribution of key components. We remove each component individually to assess its contribution except for the IFE network, we replace it with standard self-attention. The full model consistently performs well across all settings, especially in the IID setting Exp.(d), where it remains comparable to most ablated variants. In OOD scenarios, removing AUG(·) during either training or prototype generation leads to a significant performance drop, emphasizing the importance of augmenting training diversity. Removing the dictionary, which disables AUG(·) in both training and prototype generation, results in a performance degradation similar to removing only



(a) Number of anchor devices (b) Number of user devices

Figure 4: Sensitivity analysis on the number of devices

one of them. This suggests that the two augmentation stages provide cooperative benefits rather than independent contributions. The patch embedding module's removal causes a slight decline, suggesting its contribution to generalization. Lastly, replacing the IFE network with standard self-attention leads to a sharp performance drop, underscoring its superior ability to extract environment-agnostic representations.

4.5 Sensitivity Analysis on the Number of Devices

Figure 4 illustrates the model performance under varying numbers of anchor and user devices. For anchor-device evaluation, we vary the number of anchor devices in the dictionary while keeping all other settings fixed. For user-device evaluation, we use a fixed dictionary with three anchor devices and examine performance as the number of user devices increases. All experiments are conducted under Exp. (a). We observe that GenID achieves consistent performance when the number of anchor devices exceeds three. Furthermore, the number of user devices does not significantly affect performance.

5 Conclusions

We propose GenID, a deep learning-based physical-layer device identification framework. GenID tackles two key challenges: out-of-distribution shifts and environment-agnostic device identification. It employs a communication environment augmentation strategy using a transmitter dictionary and a prototype memory bank to simulate diverse signal conditions. Additionally, the IFE network extracts robust and invariant features by modeling class-level relationships. Extensive experiments demonstrate that GenID outperforms existing methods and generalizes well under OOD scenarios while maintaining computational efficiency.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62472091, 62394333, 62572361)

References

[Al-Shawabka *et al.*, 2025] Amani Al-Shawabka, Philip Pietraski, Sudhir B Pattar, Pedram Johari, and Tommaso Melodia. Signcrf: Scalable channel-agnostic data-driven radio authentication system. *IEEE Transactions on Mobile Computing*, 2025.

[Brik *et al.*, 2008] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 116–127, 2008.

[Carbino *et al.*, 2015] Timothy J Carbino, Michael A Temple, and Trevor J Bihl. Ethernet card discrimination using unintentional cable emissions and constellation-based fingerprinting. In *2015 International Conference on Computing, Networking and Communications*, pages 369–373. IEEE, 2015.

[Cha *et al.*, 2021] Junbum Cha, Hancheol Cho, Kyungjae Lee, Seunghyun Park, Yunsung Lee, and Sungrae Park. Domain generalization needs stochastic weight averaging for robustness on domain shifts. *arXiv preprint arXiv:2102.08604*, 3(3):13, 2021.

[Choi *et al.*, 2018] Wonsuk Choi, Kyungho Joo, Hyo Jin Jo, Moon Chan Park, and Dong Hoon Lee. Voltageids: Low-level communication characteristics for automotive intrusion detection system. *IEEE Transactions on Information Forensics and Security*, 13(8):2114–2129, 2018.

[Du *et al.*, 2021] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 402–411, 2021.

[Eldele *et al.*, 2024] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. Tslanet: Rethinking transformers for time series representation learning. *arXiv preprint arXiv:2404.08472*, 2024.

[Fu *et al.*, 2023] Yuqian Fu, Yu Xie, Yanwei Fu, and Yu-Gang Jiang. Styleadv: Meta style adversarial training for cross-domain few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24575–24584, 2023.

[George *et al.*, 2022] A Shaji George, AS Hovan George, T Baskar, and Digvijay Pandey. The transformation of the workspace using multigigabit ethernet. *Partners Universal International Research Journal*, 1(3):34–43, 2022.

[Gerdes *et al.*, 2012] Ryan M Gerdes, Mani Mina, Steve F Russell, and Thomas E Daniels. Physical-layer identification of wired ethernet devices. *IEEE Transactions on Information Forensics and Security*, 7(4):1339–1353, 2012.

[Hall *et al.*, 2004] Jeyanthi Hall, Michel Barbeau, Evangelos Kranakis, et al. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. *Communications, internet, and information technology*, 1, 2004.

[Han *et al.*, 2023] Guangjie Han, Zhengwei Xu, Hongbo Zhu, Yunlu Ge, and Jinlin Peng. A two-stage model based on a complex-valued separate residual network for cross-domain iiot devices identification. *IEEE Transactions on Industrial Informatics*, 20(2):2589–2599, 2023.

[He and Wang, 2020] Boxiang He and Fanggang Wang. Cooperative specific emitter identification via multiple distorted receivers. *IEEE Transactions on Information Forensics and Security*, 15:3791–3806, 2020.

[Hippenstiel and Payal, 1996] Ralph D Hippenstiel and Yalcin Payal. Wavelet based transmitter identification. In *Fourth International Symposium on Signal Processing and Its Applications*, volume 2, pages 740–742. IEEE, 1996.

[Hollmann *et al.*, 2022] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2022.

[Huang *et al.*, 2017] Guangquan Huang, Yingjun Yuan, Xiang Wang, and Zhitao Huang. Specific emitter identification for communications transmitter using multi-measurements. *Wireless Personal Communications*, 94:1523–1542, 2017.

[Khoee *et al.*, 2024] Arsham Gholamzadeh Khoee, Yinan Yu, and Robert Feldt. Domain generalization through meta-learning: a survey. *Artificial Intelligence Review*, 57(10):285, 2024.

[Kim *et al.*, 2024] Jaeho Kim, Seok-Ju Hahn, Yoontae Hwang, Junghye Lee, and Seulki Lee. Cafo: Feature-centric explanation on time series classification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1372–1382, 2024.

[LeCun *et al.*, 2002] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[Levy *et al.*, 2023] Efrat Levy, Asaf Shabtai, Bogdan Groza, Pal-Stefan Murvay, and Yuval Elovici. Can-loc: Spoofing detection and physical intrusion localization on an in-vehicle can bus based on deep features of voltage signals. *IEEE Transactions on Information Forensics and Security*, 2023.

[Li *et al.*, 2022] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:11828–11841, 2022.

[Liu *et al.*, 2022] Jiaqi Liu, Aiqun Hu, and Sheng Li. Spectrum-based fingerprint extraction and identification method of 100m ethernet card. In *2022 6th International Conference on Cryptography, Security and Privacy*, pages 102–107. IEEE, 2022.

[Lu *et al.*, 2022] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. Out-of-distribution representation learning for time series classification. *arXiv preprint arXiv:2209.07027*, 2022.

[Mick, 2000] Colin Mick. Introduction to 100base-t: Fast (and faster) ethernet. In *Network Design*, pages 69–80. Auerbach Publications, 2000.

[Peng *et al.*, 2019] Linning Peng, Junqing Zhang, Ming Liu, and Aiqun Hu. Deep learning based rf fingerprint identification using differential constellation trace figure. *IEEE Transactions on Vehicular Technology*, 69(1):1091–1095, 2019.

[Peng *et al.*, 2021] Yang Peng, Pengfei Liu, Yu Wang, Guan Gui, Bamidele Adebisi, and Haris Gacanin. Radio frequency fingerprint identification based on slice integration cooperation and heat constellation trace figure. *IEEE Wireless Communications Letters*, 11(3):543–547, 2021.

[Peng *et al.*, 2024] Yang Peng, Pengfei Liu, Qianyun Zhang, Lantu Guo, Yuchao Liu, Yu Wang, Yun Lin, and Guan Gui. Dynamic adaptation rff identification method leveraging cognitive representation learning. *IEEE Transactions on Information Forensics and Security*, 2024.

[Polak *et al.*, 2011] Adam C Polak, Sepideh Dolatshahi, and Dennis L Goeckel. Identifying wireless users via transmitter imperfections. *IEEE Journal on selected areas in communications*, 29(7):1469–1479, 2011.

[Reising *et al.*, 2020] Donald Reising, Joseph Cancelleri, T Daniel Loveless, Farah Kandah, and Anthony Skjellum. Radio identity verification-based iot security using rf-dna fingerprints and svm. *IEEE Internet of Things Journal*, 8(10):8356–8371, 2020.

[Schroter *et al.*, 2022] Hendrik Schroter, Alberto N Escalante-B, Tobias Rosenkranz, and Andreas Maier. Deepfilternet: A low complexity speech enhancement framework for full-band audio based on deep filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7407–7411. IEEE, 2022.

[Shen *et al.*, 2021] Guanxiong Shen, Junqing Zhang, Alan Marshall, Linning Peng, and Xianbin Wang. Radio frequency fingerprint identification for lora using deep learning. *IEEE Journal on Selected Areas in Communications*, 39(8):2604–2616, 2021.

[Shen *et al.*, 2023] Guanxiong Shen, Junqing Zhang, Alan Marshall, Mikko Valkama, and Joseph Cavallaro. Towards length-versatile and noise-robust radio frequency fingerprint identification. *IEEE Transactions on Information Forensics and Security*, 2023.

[Shi *et al.*, 2024] Ruize Shi, Hong Huang, Kehan Yin, Wei Zhou, and Hai Jin. Orthogonality matters: Invariant time series representation for out-of-distribution classification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2674–2685, 2024.

[Sommer *et al.*, 2010] Jorg Sommer, Sebastian Gunreben, Frank Feller, Martin Kohn, Ahlem Mifdaoui, Detlef Saß, and Joachim Scharf. Ethernet–a survey on its fields of application. *IEEE Communications Surveys & Tutorials*, 12(2):263–284, 2010.

[Song *et al.*, 2020] Chuanbiao Song, Kun He, Jiadong Lin, Liwei Wang, and John E. Hopcroft. Robust local features for improving the generalization of adversarial training. In *International Conference on Learning Representations*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Vieira *et al.*, 2010] Susana M Vieira, Uzay Kaymak, and João MC Sousa. Cohen's kappa coefficient as a performance measure for feature selection. In *International conference on fuzzy systems*, pages 1–8. IEEE, 2010.

[Wang *et al.*, 2022] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Wang *et al.*, 2024] Junxiang Wang, Guangji Bai, Wei Cheng, Zhengzhang Chen, Liang Zhao, and Haifeng Chen. Pond: Multi-source time series domain adaptation with information-aware prompt tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3140–3151, 2024.

[Wang *et al.*, 2025] Yuanchao Wang, Zhao-Rong Lai, and Tianqi Zhong. Out-of-distribution generalization for total variation based invariant risk minimization. *arXiv preprint arXiv:2502.19665*, 2025.

[Xu *et al.*, 2020] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*, 2020.

[Zhang *et al.*, 2016] Jingwen Zhang, Fanggang Wang, Octavia A Dobre, and Zhangdui Zhong. Specific emitter identification via hilbert–huang transform in single-hop and relaying scenarios. *IEEE Transactions on Information Forensics and Security*, 11(6):1192–1205, 2016.

[Zhao *et al.*, 2022] Shengkui Zhao, Bin Ma, Karn N Watcharasupat, and Woon-Seng Gan. Frcrn: Boosting feature representation using frequency recurrence for monaural speech enhancement. In *IEEE international conference on acoustics, speech and signal processing*, pages 9281–9285. IEEE, 2022.

[Zhong *et al.*, 2025] Yingyi Zhong, Wen'an Zhou, and Zhixian Wang. A survey of data augmentation in domain generalization. *Neural Processing Letters*, 57(2):34, 2025.

[Zhou *et al.*, 2023] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Mixstyle neural networks for domain generalization and adaptation. *International Journal of Computer Vision*, pages 1–15, 2023.