Patch Matter: Dual Modality Patch Contrastive for Non-stationary Radio Signals

Jie Su, Yuting Jiang, Yuheng Ye, Zhenyu Wen†, Taotao Li, Shibo He, Xiaoqin Zhang, and Rajiv Ranjan *Fellow IEEE*

***Abstract*—The emergence of abundant non-stationary radio signal (NSRS) data presents significant opportunities for applications in wireless communications, radar systems, remote sensing, and healthcare. While deep learning models have shown promise in capturing sequence dependencies, deriving generic and fine-grained representations of NSRS data remains challenging due to its complex, dynamic nature and the scarcity of labeled data. The NSRS data are often frequency-sensitive and exhibit minuscule inter-class distances, posing significant challenges for precise classification. To address these issues, we propose a novel Dual Modality Patch Contrastive (DMPC) framework. This framework leverages a stochastic patching paradigm for diverse local pattern extraction and a time-frequency cross-view optimization for frequency-sensitive feature mining. Furthermore, an Attentive Patch Aggregation (APA) mechanism enhances fine-grained inference under few-shot conditions through patch-level feature voting. Extensive experiments demonstrate the effectiveness of our approach in addressing the unique challenges of NSRS data.**

***Index Terms*—Non-stationary Signal Processing, Contrastive Learning, Multi-Modality**

I. Introduction

Recent advancements in wireless communication technologies [1]–[5] have led to an exponential increase in the volume and complexity of radio signal data. Non-stationary radio signals (NSRS), characterized by their time-varying statistical properties, are particularly challenging to analyze and model. Deep learning has been widely adopted in applications such as automatic modulation recognition, signal decomposition, and spectrum monitoring due to its powerful ability in automatic pattern mining. However, a significant portion of NSRS data remains unlabeled or poorly annotated, posing challenges for deep learning-based methods. Inefficient handling of such vast and complex data can lead to inaccurate security identification and wasted spectrum resources.

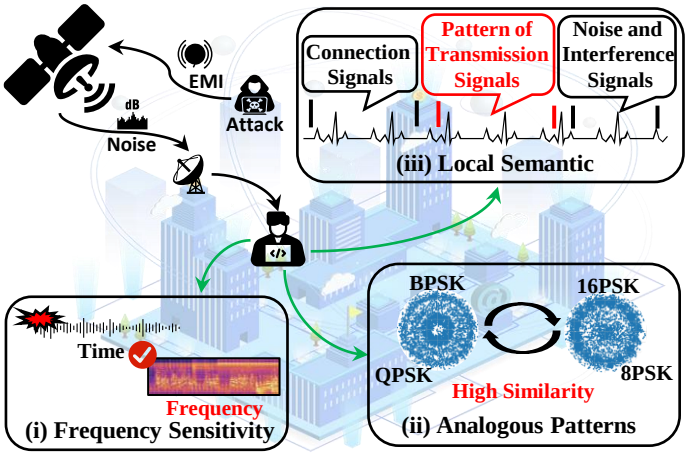


Fig. 1: The common complex characteristics in NSRS data.

Recently, numerous studies [6], [7] have explored pre-training approaches for NSRS data, adopting the "pre-training then fine-tuning" paradigm to address the challenge of limited annotated data. These approaches leverage pre-trained models that can easily adapt to new tasks. However, most existing methods apply pre-trained frameworks, e.g. contrastive learning (CL), from the vision or time-series domains to NSRS data, despite the unique characteristics that make direct application of standard CL methods challenging.

Challenges of applying CL approaches to NSRS data. As illustrated in Figure 1, NSRS data often exhibit complex characteristics that pose challenges to designing effective frameworks for learning robust representations. These challenges can be summarized as follows: **i) Frequency Sensitivity**–Transmitted radio signals often encode implicit patterns in the frequency domain, such as harmonic structures, periodic components, and spectral sparsity. These characteristics are essential for understanding signal behavior and ensuring accurate classification or recognition. However, conventional contrastive learning (CL) methods for time-series data, such as FOCAL [8] and COMET [9], predominantly focus on the time domain, relying on temporal contrastive mechanisms to mine sequential patterns. This narrow focus limits their ability to capture the periodic and spectral features inherent in the frequency domain. **ii) Analogous Patterns**–NSRS data from the same category often exhibit variations in amplitude levels despite sharing similar structural characteristics. For instance, signals

This work was supported by the National Nature Science Foundation of China under Grant 62472387 and U24A20242, National Key Research and Development Program Project of China under Grant 2024YFC3306902, Zhejiang Provincial Natural Science Foundation of Major Program (Youth Original Project) under Grant LDQ24F020001, Zhejiang Provincial Natural Science Foundation under Grant LDT23F02024F02, Key Research and Development Program of Zhejiang Province Grant 2024C01065.

Jie Su, Yuting Jiang, Yuheng Ye, Zhenyu Wen, Taotao Li are with the Institute of Cyberspace Security, and College of Information Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang, China (E-mail: jiesu@zjut.edu.cn, 221123030164@zjut.edu.cn, yeyuh@zjut.edu.cn, wenluke427@gmail.com, 2111903074@zjut.edu.cn)

Xiaoqin Zhang are with the school of computer science, Zhejiang University of Technology, Hangzhou, Zhejiang, China (E-mail: zhangxiaoqinnan@gmail.com)

Shibo He is with the School of Control Science and Engineering, Zhejiang University, Hangzhou, China. (e-mail: s18he@zju.edu.cn)

Rajiv Ranjan is with the School of Computing Science, Newcastle University, NE1 7RU, Newcastle upon Tyne, U.K. (e-mail: raj.ranjan@ncl.ac.uk)

Manuscript received April 19, 2021; revised August 16, 2021.

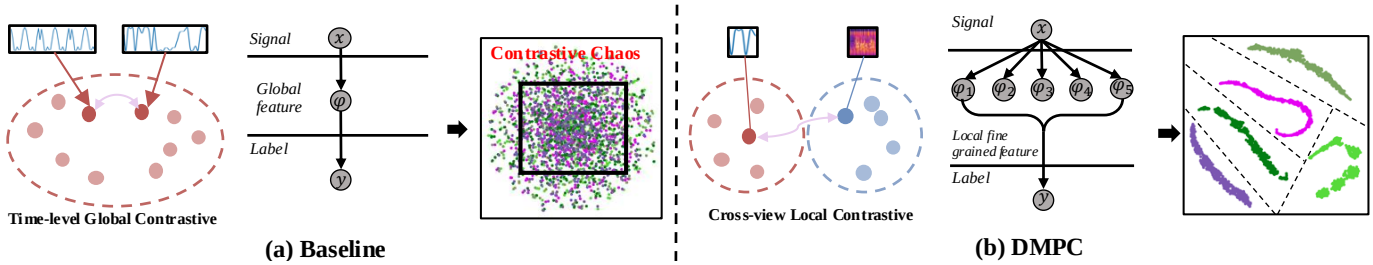


Fig. 2: Figure (a) shows that conventional CL approaches often extract global features from NSRS data, resulting in contrastive chaos for analogous classes. In contrast, the proposed DMPC framework (Figure (b)) extracts fine-grained, modality-invariant local features, creating a more discriminative representation space.

with identical modulation schemes may differ in amplitude due to factors such as transmission power, distance, or environmental interference, while maintaining their overall structure. Conventional augmentation-based contrastive learning (CL) approaches, such as SimCLR [10] and InfoTS [11], aim to identify meaningful augmentations of time-series data for contrastive learning tasks. However, these methods primarily enforce temporal consistency, which struggles to generalize to amplitude variations unless explicitly designed to account for such variability. **iii) Local Semantic**–Transmitted NSRS data typically span extended durations, exhibiting intricate temporal variations. However, most conventional CL approaches [10], [12], [13] leverage the entire sequential time-series data via temporal-consistent and augmentation-consistent mechanisms to construct contrastive tasks. This over-reliance on global features makes it difficult to capture meaningful patterns from long-sequential radio signals, resulting in coarse representation spaces (e.g., the "Contrastive Chaos" depicted in Figure 2(a)). Signals contain diverse local semantics and periodicities [14], which vary across different segments and must be extracted separately to build a more nuanced representation.

To tackle this issue, we propose a unified framework called Dual Modality Patch Contrastive (DMPC), designed to learn fine-grained representations from unlabeled NSRS data and support a wide range of downstream tasks. Specifically, we first establish the theoretical foundation showing that patch-level contrastive optimization serves as an effective alternative for uncovering co-occurrence statistics, thereby improving the ability to capture locally sensitive contextual information. Next, we address a key limitation of previous patch-based CL methods [15], [16], which overlook important frequency patterns in signals. Time-series signals often contain features like oscillations across different frequency bands that cannot be fully captured by focusing only on the time domain. To solve this, we introduce a patch-level contrastive learning approach in the time-frequency domain, allowing the model to better capture local dependencies between time and frequency.

The proposed framework consists of three main components: 1) The **Stochastic Patching Augmentation** initially generates fixed-scale, non-overlapping patches, and varied views/augmentations as the input for the subsequent contrastive learning process. This diverse signal exposes the model to complex temporal dynamics, resulting in more robust time-based embeddings. 2) The **Dual Modality Patch Contrastive** module would take the generated patches as input to mine implicit local fine-grained representations with maximum time-frequency consistency from a cross-modality perspective. After the contrastive pre-training, the 3) **Attentive Patch Aggregation** module is introduced to aggregate the "bag-of-patches" representations. It encourages local fine-grained patterns to contribute to the final classification of analogous categories. Empirical experiments demonstrate that patch-wise contrastive learning can achieve up to an 8% accuracy improvement in linear evaluation, making it competitive with supervised learning performance. Furthermore, patch-level training significantly accelerates training efficiency (e.g., up to $8\times$ faster) and constructs a more distinguishable representation space (e.g., as shown in Fig. 2(b)), thereby enhancing both the efficiency and representation quality for future CL approaches.

Our main contributions can be summarized as follows:

- In this paper, we propose a novel end-to-end DMPC method to explore patch-wise contrastive learning paradigms from a cross-modality view, enabling discriminative representation space construction.
- We provide a theoretical basis and in-depth analysis of how patch-wise contrastive learning functions as co-occurrence statistics modeling, thereby enhancing the extraction of locally sensitive context information.
- We further explore the potential knowledge transfer ability of the DMPC across datasets of different scales, marking the first attempt in the signal processing domain.
- Extensive experiments were conducted on three public datasets, and the promising results demonstrate its effectiveness. Furthermore, the empirical findings on training efficiency provide valuable insights and contribute to advancing future CL-based signal processing research.

II. Related Work

In the following, we briefly discuss the most relevant approaches related to our work.

A. Contrastive Learning

Contrastive learning was first studied in the self-supervised setting [11], [17]–[19], typically relying on a pretext task to learn embedding. The method in [20] used a probabilistic contrastive loss to capture mutual information between different views of data. Later, SimCLR [10] was introduced to facilitate large-scale representation learning through an effective augmentation contrastive task. It incorporates a nonlinear layer within a contrastive learning framework to bolster

the efficiency of representation learning. Moco [12], [21], on the other hand, addresses the computational challenges of SimCLR by employing a memory bank to store a larger dictionary of negative samples, thereby enabling more efficient contrastive learning. The negative samples enable the network to distinguish between different classes or concepts, and the quality of the sampled negative pair determines the quality of the learned latent space. Consequently, a series of works [22]–[24] investigate the hard negative sampling approach to enhance the acquisition of more discriminative representations. Notably, traditional contrastive learning learned from positive and negative samples can easily collapse and generate trivial constant solutions. Recent works remove negative samples to learn a robust model. [25] maximizes the consistency between variables and representations, reduces feature redundancy, and increases the information density of training samples without using negative samples. [13] is achieved by decorrelating the different dimensions of the instance embedding stacking matrix and eliminating the correlation information of negative samples.

B. Contrastive Learning on Signal Process

Benefiting from recent advances in the vision and language processing community, contrastive learning offers a promising solution for the signal processing community to learn representations from large amounts of unlabeled signal sequences. [26] proposed an unsupervised framework based on generative adversarial networks (InfoGANs) and radio frequency fingerprint embedding (RFFE) to enhance individual discriminability. Later, [27] explored the contrastive setting on the constellation modality of signals, effectively learning spatial patterns from signals. SA2SEI [28] was designed to leverage adversarial augmentation (Adv-Aug) to overcome the limitation of label dependence for auxiliary datasets. [29] integrated multi-domain information of radar signal intra-pulses to obtain information fusion samples, enabling the network to extract deep features of radar signal intra-pulses.

Although these approaches yield promising results across various signal processing tasks, many of them still depend on advanced Contrastive Learning (CL) approaches from other domains [27]. While some attempt to embed special signal characteristics [26], [30], most focus on contrastive learning using small-scale datasets, limiting their generalizability. Moreover, none of the prior studies examine the knowledge transfer capabilities of the trained network, which make it less reliable in real-world scenarios.

C. Patch-based Contrastive learning

Patch-wise strategies have become a strong paradigm in the field of computer vision for self-supervised learning for learning locally discriminative representations. The CPC [20] framework was first proposed to learn representations by predicting future latent patch embeddings via an autoregressive contrastive objective. DINO [31] used self-distillation on patch tokens to produce semantically meaningful local descriptors. MAE [32] was proposed to convert patches into a masked-reconstruction objective that forces the model to learn the contextual relationship between the masked patch and all visible patches. Later, SelfPatch [33] applies contrastive losses at the patch level to encourage local separability. Conventional ViT-style patching, followed by global contrastive or classification heads, likewise treats patches as atomic units. Despite their strengths, these approaches have common limitations for time–frequency signal processing: many use fixed-scale, rigid patching or a single view of locality, which under-represents temporal dynamics and scale variability; reconstruction-heavy solutions (MAE-style) can be computationally costly and are not always optimal for downstream discriminative tasks; single-modality or per-patch contrast ignores cross-modality time–frequency consistency that is crucial for signals; and final aggregation is often a blunt instrument (global pooling) that dilutes locally discriminative patterns.

III. Problem Formulation

Background Knowledge. In wireless communication systems, modulation aims to add information to a set of signals by varying one or more properties of periodic electromagnetic waves (carriers) which can be transmitted [34]. A transmitted time modulation signal $r(t)$ can be illustrated as:

$$r(t) = \mathcal{S}(t) * h(t) \exp[j2\pi\Delta f t + \psi_0] + noise(t), \quad (1)$$

where $*$ represents the convolution operation, the transmission channel response is symbolized as $\mathcal{S}(t)$, $h(t)$ stands for the baseband signal being transmitted, Δf indicates the carrier frequency offset, ψ_0 signifies the initial phase, and $noise(t)$ refers to the environmental noise.

To facilitate signal information extraction and signal recovery, in-phase signals and the quadrature-phase signal are used to jointly characterize the relevant modulation information, i.e. I-Q data [35]. So we define the received discrete complex signal as $x_{IQ} = \{x^I, x^Q\}$, which is sampled from $r(t)$:

$$\begin{aligned} \{x^I, x^Q\} &= sample\{r_I(t), r_Q(t)\} \\ \{r_I(t), r_Q(t)\} &= \{Re\{r(t)\}, Im\{r(t)\}\}, \end{aligned} \quad (2)$$

where x^I denotes the in-phase signal, x^Q represents the quadrature-phase signal, Re is the real part and Im indicates the imaginary part. For the automatic modulation classification task, the modulated signal segment x (i.e., sampled from x_{IQ}) and its corresponding label y are used for the training procedure (feature extraction).

Automatic Modulation Recognition Pre-training. The pre-training process operates in two phases: 1) During the training stage, the model is trained on a large-scale dataset using a self-supervised or contrastive learning objective without label information to learn a robust representation space. 2) After pre-training, a small labeled subset of the dataset is used to fine-tune the pre-trained model, enabling rapid adaptation to specific tasks.

We define the pre-training dataset as $D_{pre} = \{x, y\}$, where x represents the quadrature modulated signal segments, and the corresponding label $y \in \mathbb{R}^C$ indicates one of the C modulation classes. During the pre-training stage, label information is not available. A small subset of the pre-training dataset (labeled), denoted as $\hat{D}_{pre} = \{\hat{x}, \hat{y}\}$, consisting of typically 1% or 10%

TABLE I: Summary of notations.

Notations	Explanations
x, ch, L	Input signal sequence, Input signal channels, and its length
st, et	The start and end indices of the signal patch
$x_{pi}, \tilde{x}_{pi}, x_{pi}^{f}, l$	Signal patches, Augmented signal patches, Converted frequency patches, and patch length
x_p^Q, x_p^I	Real and imaginary part of signal
$z_i, \tilde{z}_i, z_i^f$	Extracted time-level patch features, Augmented time-level patch features, and frequency-level features
Θ, t, τ, j	Random rotation angles, Flipping time step, Temperature parameter in contrastive learning, and imaginary unit
$\{x, x^+\}, \{x_{pi}, \tilde{x}_{pi}\}$	Positive signal pairs
$\{x_{pi}, \tilde{x}_{pk}\}$	Negative signal pairs
$f_\theta, \mathcal{F}_\theta, sim, FFT$	Function of time encoder, Frequency encoder, Cosine similarity, and Frequency transform
W_Q, W_K, W_V	Learned weight matrices of attention
Q, K, V	Value of attention
A_{in}, A_w, Pw	Time-frequency joint features, Attentive category probabilities, Patches weight

of the pre-training data, is used for fine-tuning. As a result, the objective of the pre-training could be formulated as follows:

$$\min(\mathcal{L}_{pretext}) = \mathbb{E}_{x\sim D_{pre}}\left[g(f(x), f(x^+))\right], \tag{3}$$

where the x^+ denotes a positive view of x, created using data augmentation techniques. The $f(\cdot)$ represents the encoder function that maps input signals into a feature representation space. The $g(\cdot,\cdot)$ is a task-specific function that enforces alignment or consistency between the representations of x and x^+.

After the pre-training, the learned encoder function would be further tuned using $\hat{D}_{pre}$, which can be formulated as:

$$min(\epsilon_{error}) = \mathbb{E}_{(\hat{x},\hat{y})\sim\hat{D}}[f'(\hat{x}) \neq \hat{y}], \tag{4}$$

where ϵ_{error} denotes the target error on labeled subset, and the f' represents the trained.

IV. Methodology

In this section, we first provide an overview of the proposed framework. Then, we introduce the theoretical basis for patch-based contrastive learning and present the details of the key components. Moreover, notations utilized in this paper are summarized in Table I.

A. Framework Overview

Figure 3 presents the key components and the workflow of the proposed framework: 1) The **Stochastic Patching Augmentation** module aims to produce diverse augmented signal segmentation for subsequent contrastive optimizations. 2) The **Patch-Wise Cross-View Contrastive Learning** module aims to obtain local temporal-reliance and frequency-sensitive features from stochastic sampled patches by encouraging the information consistency of the time-to-time and time-to-frequency pairs; 3) The **Attentive Patch Aggregation** module aims to assign attention weights to each cross-view patch combination, thereby encouraging essential implicit features to contribute significantly to the final discrimination.

Pipeline. During the *Pre-training* stage, given the time-series signal input, the **Stochastic Patching Augmentation** module first randomly generates non-overlapping fixed-scale *time-series signal patches* and their corresponding augmentations. Additionally, the frequency conversion (i.e., FFT transform) is performed on the time-series patches to obtain *frequency signal patches*. Next, the **Patch-Wise Cross-View Contrastive Learning** component encodes the time and frequency patches using a time encoder and a frequency encoder to obtain features from the time and frequency views. The encoders are then optimized with the proposed time-time and time-frequency cross-view contrastive losses. During the *Fine-tuning and inference* stage, the **Attentive Patch Aggregation** module takes the extracted cross-view features as input and produces attentive class probabilities for each patch. Subsequently, these attentive class probabilities undergo a majority voting procedure to determine the final patch feature weights. These weights are then multiplied with the cross-view features for the final classification.

B. Why Patch Contrastive?

Conventional contrastive learning (CL) provides a promising solution for learning meaningful signal representations under conditions of label scarcity. Previous CL works in signal processing typically enforce a non-collapsing representation space by leveraging global-level signal information. This global approach may neglect local semantics (e.g., instantaneous properties of the signals), leading to coarse-grained representation space in distinguishing analogous categories with marginal distribution differences. For example, some long sequential signals like airplane broadcasts often contain rich local patterns that are crucial for capturing the underlying dynamics and nuances of the data. Recent studies in the vision domain [32] have demonstrated that patch-level contrastive learning (CL) is an effective method for exploring locally sensitive information, making it a potential solution for signal processing. However, the principle behind patch-level contrastive learning remains unclear. Thus, in the following, we provide the theoretical basis for the fact that patch-level contrastive optimization serves as an alternative method for mining co-occurrence statistics.

Link Patch Contrastive to Co-Occurrence Statistics Modeling. Co-occurrence statistics [36] capture how often certain features or patterns appear together within the data. Previous

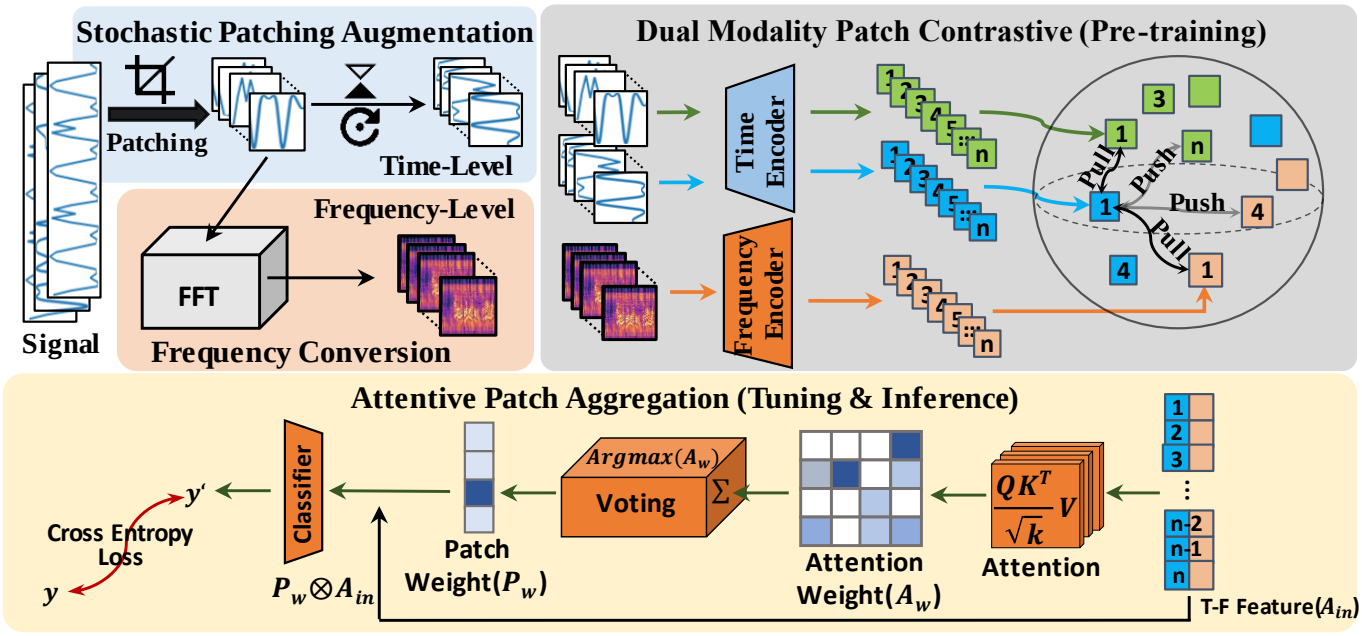


Fig. 3: The Framework of DMCP and APEI.

works in natural language processing [37], [38] have demonstrated that the co-occurrence statistics modeling can capture the local sensitive context information of words in a sentence. This helps in understanding the underlying dependencies and relationships between different parts of the data, which is crucial for mining local representative information. The patch-level contrastive aims to align the representations of different views of the same instance. This process intuitively leads to the representations becoming invariant to the transformations that produce these views. This learning objective can be interpreted as utilizing the inner product to capture the co-occurrence statistics of local signal patches, as presented below.

Given the input time-series signal $x \in \mathbb{R}^{C\times L}$ with C channels and L length, the signal patches can be denoted as $x_p \in \mathbb{R}^{C\times(st-et+1)}$, where st and et are the start and end indices of the patch, respectively. Assume $\tilde{x}_{p1}$ and $\tilde{x}_{p2}$ are two randomly augmented signal patches sampled from the signal dataset. Let $q(\tilde{x}_{p1})$ and $q(\tilde{x}_{p2})$ represent their marginal distributions, accounting for variations due to different signals, patch locations within a signal, and random augmentations. The joint distribution $q(\tilde{x}_{p1}, \tilde{x}_{p2})$ signifies the probability of these patches co-occurring within the same signal. $\tilde{z}_1$ and $\tilde{z}_2$ are corresponding embedding vectors of $\tilde{x}_{p1}$ and $\tilde{x}_{p2}$ in the representation space, with the transformation parameterized by a neural network. We first restate the modeling goal: the patch-level spectral contrastive objective is intended to (i) pull together embeddings of co-occurring patches (samples drawn from the joint distribution $q(\tilde{x}_{p1}, \tilde{x}_{p2})$) and (ii) avoid representation collapse by penalizing trivial alignment under the product-of-marginals $q(\tilde{x}_{p1})q(\tilde{x}_{p2})$. Below, we make this connection precise by showing that the spectral contrastive loss L_{Spec} can be written as an instance of a co-occurrence statistics modeling loss L_{Co} up to multiplicative constants.

Proposition. The patch-level spectral contrastive loss function L_{Spec}:

$$L_{Spec} = \mathbb{E}_{q(\tilde{x}_{p1},\tilde{x}_{p2})}\left[-\tilde{z}_1^T\tilde{z}_2\right] + \lambda\,\mathbb{E}_{q(\tilde{x}_{p1})q(\tilde{x}_{p2})}\left(\tilde{z}_1^T\tilde{z}_2\right)^2, \tag{5}$$

is equivalent (up to constants that do not depend on the embeddings $\tilde{z}_1, \tilde{z}_2$) to the co-occurrence statistics modeling loss L_{Co}:

$$L_{Co} = \int q\left(\tilde{x}_{p1}\right) q\left(\tilde{x}_{p2}\right) \left[w\,\tilde{z}_1^T\tilde{z}_2 - \frac{q\left(\tilde{x}_{p1},\tilde{x}_{p2}\right)}{q\left(\tilde{x}_{p1}\right) q\left(\tilde{x}_{p2}\right)} \right]^2 d\tilde{x}_{p1}d\tilde{x}_{p2}. \tag{6}$$

Proof. We expand L_{Co} and isolate the terms that depend on the embeddings. Constants or terms independent of $\tilde{z}_1, \tilde{z}_2$ are dropped since they do not affect optimization over the embeddings.

$$L_{Co} = \int q(\tilde{x}_{p1})q(\tilde{x}_{p2})\left(w\,\tilde{z}_1^T\tilde{z}_2 - \tfrac{q(\tilde{x}_{p1},\tilde{x}_{p2})}{q(\tilde{x}_{p1})q(\tilde{x}_{p2})}\right)^2 d\tilde{x}_{p1}d\tilde{x}_{p2} \tag{7}$$

$$= \int q(\tilde{x}_{p1})q(\tilde{x}_{p2})\left[(w\,\tilde{z}_1^T\tilde{z}_2)^2 - 2w\,\tilde{z}_1^T\tilde{z}_2 \cdot \frac{q(\tilde{x}_{p1},\tilde{x}_{p2})}{q(\tilde{x}_{p1})q(\tilde{x}_{p2})}\right] d\tilde{x}_{p1}d\tilde{x}_{p2} \;+\; C, \tag{8}$$

where C collects terms that do not depend on $\tilde{z}_1, \tilde{z}_2$ (and hence can be ignored for embedding optimization).

We now recognize the two remaining integrals as expectations under the indicated distributions:

$$\int q(\tilde{x}_{p1})q(\tilde{x}_{p2})\left(w\,\tilde{z}_1^T\tilde{z}_2\right)^2 d\tilde{x}_{p1}d\tilde{x}_{p2} = w^2\mathbb{E}_{q(\tilde{x}_{p1})q(\tilde{x}_{p2})}\left[\left(\tilde{z}_1^T\tilde{z}_2\right)^2\right], \tag{9}$$

and

$$
\begin{aligned}
&-2w \int q(\tilde{x}_{p1}) q(\tilde{x}_{p2}) \tilde{z}_1^T \tilde{z}_2 \cdot \frac{q(\tilde{x}_{p1}, \tilde{x}_{p2})}{q(\tilde{x}_{p1}) q(\tilde{x}_{p2})} \, d\tilde{x}_{p1} d\tilde{x}_{p2} \\
&\quad = -2w \, \mathbb{E}_{q(\tilde{x}_{p1}, \tilde{x}_{p2})} \left[\tilde{z}_1^T \tilde{z}_2 \right].
\end{aligned} \tag{10}
$$

Combining these terms and ignoring the additive constant C, we obtain

$$
L_{Co} = -2w \, \mathbb{E}_{q(\tilde{x}_{p1}, \tilde{x}_{p2})} \left[\tilde{z}_1^T \tilde{z}_2 \right] + w^2 \, \mathbb{E}_{q(\tilde{x}_{p1}) q(\tilde{x}_{p2})} \left[\left(\tilde{z}_1^T \tilde{z}_2 \right)^2 \right]. \tag{11}
$$

Up to an overall positive scalar factor, this matches the form of L_{Spec}. If we choose the scaling such that the linear term coefficient equals -1, i.e. divide through by $2w$, then we obtain

$$
L_{Spec} = \mathbb{E}_{q(\tilde{x}_{p1}, \tilde{x}_{p2})} \left[-\tilde{z}_1^T \tilde{z}_2 \right] + \lambda \, \mathbb{E}_{q(\tilde{x}_{p1}) q(\tilde{x}_{p2})} \left(\tilde{z}_1^T \tilde{z}_2 \right)^2, \tag{12}
$$

with the identification $\lambda = \frac{w}{2}$. This completes the derivation showing that the spectral contrastive objective is an instance of co-occurrence statistics modeling, differing only by constant scaling factors that do not affect the learned embeddings.

Interpretation. The first (linear) term encourages embeddings of co-occurring patches to be close in inner-product similarity; the second (quadratic) term enforces dispersion under the product-of-marginals, which prevents collapse and promotes uniformity. Thus L_{Spec} simultaneously models co-occurrence strength and imposes a regularizer that preserves discriminability.

Link to NT-Xent. The NT-Xent loss [10] augments this pairwise formulation by replacing the simple inner-product similarity with a log-softmax normalized similarity:

$$
L_{\text{NT-Xent}} = \mathbb{E}_{q(\tilde{x}_{p1}, \tilde{x}_{p2})} \left[-\log \frac{\exp(\tilde{z}_1^T \tilde{z}_2 / \tau)}{\sum_{k=1}^{N} \exp(\tilde{z}_1^T \tilde{z}_k / \tau)} \right], \tag{13}
$$

where the denominator sums over (in-batch) negatives k. This normalization can be interpreted as a data-dependent way to enforce the dispersion/anti-collapse effect embodied by the quadratic term in L_{Spec}. Consequently, patch-level NT-Xent optimization inherits the co-occurrence modeling properties described above while additionally benefiting from softmax normalization across multiple negatives.

C. Dual Modality Patch Contrastive

The aforementioned theoretical basis demonstrates that modeling patch-level contrastive optimization enables the mining of patch-level co-occurrence statistics, thereby presenting a promising solution for fine-grained local pattern discovery. However, the complexity of time series signals, including varying semantic meaning and frequency components, poses significant challenges for learning paradigms that rely solely on time-level information. The lack of frequency domain information makes it difficult to capture periodic and oscillation patterns, resulting in coarse representation extraction. To address these limitations, we introduce patch-level cross-view contrastive optimization paradigms that aim to mine local fine-grained representations with maximum time-frequency consistency from a cross-modality perspective.

Stochastic Patching Augmentation aims to generate fixed-scale, non-overlapping patches and varied views (augmentations) from the time-series signal for subsequent contrastive learning. Unlike the regular patching strategy, we introduce a stochastic patching mechanism that randomly selects the starting time step of each signal patch. This mechanism exposes the model to complex temporal dynamics, resulting in more robust time-based embeddings.

Specifically, for a given signal x, the patching process will generate n non-overlapping sequences of patches $x_p \in \mathbb{R}^{ch \times n \times l}$, where l denotes the length of each patch and $n \times l = L$. Subsequently, we follow the previous study [39] to employ rotation and flipping augmentations to the selected patches, which can be represented as:

$$
\begin{aligned}
&Rotate : R(x_p) = [x_p^I + j \cdot x_p^Q] \, [cos(\Theta) + j \cdot sin(\Theta)] \\
&Flip : F(x_p) : x_p(t) = x_p(l - t), t \in [0, l],
\end{aligned} \tag{14}
$$

where x_p^I, x_p^Q denote the real and imaginary parts of IQ signal, Θ denotes the angle of rotate, j denotes the imaginary unit and t represents the flipping time step. We will perform random augmentation for a signal patch, including random rotation angles $\Theta \in [0, \pi]$ as well as random flip.

Time-based Contrastive aims to extract invariant temporal reliance features from the previously generated time-series signal patches and their corresponding augmentations. By maximizing the agreement between patches and their augmented counterparts while minimizing the similarity with other patches, the model is able to learn to capture consistent temporal features despite variations in the data.

Specifically, given a batch of signal patch samples x_p, the aforementioned signal augmentation approaches are applied, resulting in augmented patches $\tilde{x}_p$. Each augmented patch is paired with its corresponding original patch to form positive sample pairs, i.e., $\{x_{pi}, \tilde{x}_{pi}\}$, where i denotes the patch index. Further, augmented patches from different signal patches serve as negative examples, i.e., $\{x_{pi}, \tilde{x}_{pk}\}$ for $i \neq k$. The constructed positive and negative sample pairs are then fed into the time encoder f_θ to obtain the corresponding time-level feature vectors as $\{z_i = f_\theta(x_{pi}), \tilde{z}_i = f_\theta(\tilde{x}_{pi}), \tilde{z}_k = f_\theta(\tilde{x}_{pk})\}$. Finally, the contrastive task in the time domain is achieved by optimizing the time-based NT-Xent contrastive loss $\mathcal{L}_{tcl}$, which can be calculated by:

$$
\mathcal{L}_{tcl} = -\frac{1}{N} \sum_{i=1}^{N} (\log \frac{\exp(\text{sim}(z_i, \tilde{z}_i)\tau)}{\sum_{k \in N} \mathbb{1}_{[k \neq i]} \exp(sim(z_i, \tilde{z}_k)\tau)}), \tag{15}
$$

where the $sim(z, z) = z^T \cdot z / \|z\| \, \|z\|$ is a cosine similarity, N is the batch sample numbers, and $\mathbb{1}$ is an indicator function evaluating to 1 if $k \neq i$, and τ denotes a temperature parameter.

Time-Frequency Contrastive. To encode the frequency pattern, we first use Fast Fourier Transform (FFT) [40] to convert the signal into the frequency domain. The Fast Fourier Transform (FFT) recursively breaks down the signal, employs butterfly operations, and utilizes pre-computed twiddle factors to compute the frequency spectrum with a complexity of $O(N \log N)$. Specifically, with the aforementioned time-series signal patches x_p, the frequency modulation conversion can

be represented by $x_p^f = \text{FFT}(x_p)$. Then, the positive pair can be constructed by pairing the time-series signal patches with their corresponding frequency patches, i.e., $\{x_{pi}, x_{pi}^f\}$. The negative pair can be constructed by pairing the time-series signal patches with other frequency patches, i.e., $\{x_{pi}, x_{pk}^f\}$ for $i \neq k$. Since the time-level feature is already extracted from the time-based contrastive learning, only the frequency patch input is fed into the frequency encoder $\mathcal{F}_\theta$ to obtain the corresponding frequency-level feature vectors as $\{z_i^f = \mathcal{F}_\theta(x_{pi}^f), z_k^f = \mathcal{F}_\theta(x_{pk}^f)\}$. Finally, by maximizing the temporal-frequency similarity of the same sample and minimizing the temporal-frequency similarity of different samples, we can obtain frequency patterns that possess a high dependence on the time domain. The time-frequency NT-Xent contrastive loss $\mathcal{L}_{tfcl}$ can be calculated by:

$$\mathcal{L}_{tfcl} = -\frac{1}{N}\sum_{i=1}^{N}(\log \frac{\exp(sim(z_i, z_i^f)/\tau)}{\sum_{k\in N} \mathbb{1}_{[k\neq i]} \exp(sim(z_i, z_k^f)\tau)}), \tag{16}$$

where the $sim(z, z) = z^T \cdot z/\|z\| \, \|z\|$ is a cosine similarity, N is the batch sample numbers, and $\mathbb{1}$ is an indicator function evaluating to 1 if $j \neq i$, and τ denotes a temperature parameter.

Put them together. To this end, the input time-series signals are first patched and augmented via stochastic patching augmentation as described in Eq. 14. Then, the Time-based and Time-Frequency Contrastive (i.e., Eq. 15 and Eq. 16) will take the generated patches and their corresponding augmentations to optimize the neural network simultaneously. The final optimization objective can be written as:

$$\mathcal{L}_{Pre} = \mathcal{L}_{tfcl} + \mathcal{L}_{tcl}. \tag{17}$$

Optimization. During the pre-training stage, for a batch of signal samples, each sample is first partitioned into fixed-length patches $(Batch \times Channel \times PatchLength)$. From these patches, we construct two corresponding batches — an augmentation batch $(Batch \times Channel \times PatchLength)$ and a frequency batch $(Batch \times Channel \times PatchLength)$ — for subsequent optimization. For each patch in the original batch, we form a positive pair consisting of the original patch together with its corresponding augmentation patch and frequency patch, and optimize this pair according to Eq. 15 and Eq. 16 to learn time-level and time–frequency level invariant representations. Conversely, when a patch from the original batch is paired with augmentation or frequency patches from other samples in the batch, we treat these as negative pairs and push them apart according to Eq. 15 and Eq. 16 to encourage discriminative representations.

D. Attentive Patch Aggregation

After learning a representation of fixed-scale signal patches, we aggregate all the signal patches into a fixed size embedding space for the final downstream fine-tuning and inference tasks. Conventional strategies for aggregating these patches, such as hard voting, neglect the differences in the information contributed by different patches and the potential noise they may introduce, leading to biased final results. To address these challenges, we introduce the attentive patch aggregation module, which aggregates the "bag-of-patches" representations and encourages local fine-grained patterns to contribute to the final classification of analogous categories.

Algorithm 1: Voting Algorithm

Input: Attention weight: A_w
Output: Patch weight: P_w
1 **Step 1**: Determine the category of each patch: C
2 $C = \underset{c}{\text{Argmax}}(A_w[bs, n, c])$
3 **Step 2**: Calculate the P_w
4 $P_w[bs, n] = 0.2$; `// Initialize weights of patches`
5 **for** *b in bs* **do**
6 $C\_c = \sum_{i=1}^{class} \sum_{j=n}^{c} \mathbb{1}(i = C[b, j])$
7 $C_{max\_index} = \text{Argmax}(C\_c)$
8 `// Set the weight of` P_w
9 **if** C_{max_index} = *patch_index* **then**
10 $P_w[bs, patch\_index] = 1$, $patch\_index \in (1, n)$
11 **end**
12 **end**
13 **return** P_w

Specifically, we first concatenate the extracted cross-view (i.e., time and frequency level) features, $z \in \mathbb{R}^{bs\times n\times f}$ and $z^f \in \mathbb{R}^{bs\times n\times f}$, to produce the fusion feature $A_{in} \in \mathbb{R}^{bs\times n\times 2f}$ as illustrated in Eq 18.

$$A_{in} = Concate(z, z^f). \tag{18}$$

Then, the self-attention network would take the fusion feature as the input and apply linear transformations to obtain the query, key, and value matrices:

$$Q = W_Q A_{\text{in}} \tag{19}$$

$$K = W_K A_{\text{in}} \tag{20}$$

$$V = W_V A_{\text{in}}, \tag{21}$$

where W_Q, W_K, and W_V are the learned weight matrices.

Finally, the attentive category probabilities A_w are computed by taking the dot product of the query and key matrices, scaling by the square root of the dimension k, and applying the softmax function and an MLP layer:

$$A_w = MLP(D(\text{softmax}\left(\frac{QK^T}{\sqrt{k}}\right)V)), \tag{22}$$

where $\sqrt{k}$ is the dimension of the key vectors, and D stands for dropout function with dropout ratio 0.1.

With the given attentive category probabilities, we perform the majority voting and assign specific weights.

$$P_w = Voting(A_w), \tag{23}$$

where $P_w \in \mathbb{R}^{bs\times n}$ and $Voting$ function is explained in Algorithm 1.

Firstly, the category predictions for different patches are calculated (line 2). Then, each patch's category is counted to find the one with the most votes (lines 6-7). Finally,

TABLE II: Partitions of three benchmark datasets.

Dataset Setting	Pre-training	Fine-tuning	Test
Dimension of $\mathcal{D}_{ADS-B}$ and $\mathcal{D}_{WiFi}$	2×4800	2×4800	2×4800
Dimension of $\mathcal{D}_{Sig53}$	2×4096	2×4096	2×4096
Number of categories in $\mathcal{D}_{ADS-B}$	90	10	10
Number of categories in $\mathcal{D}_{WiFi}$	10	6	6
Number of categories in $\mathcal{D}_{Sig53}$	53	53	53
Number of samples per category in $\mathcal{D}_{ADS-B}$	100	{1, 10}	100
Number of samples per category in $\mathcal{D}_{WiFi}$	1000	{1, 10}	1000
Number of samples per category in $\mathcal{D}_{Sig53}$	2000	{20, 200}	2000

the category with the highest count in the previous step is mapped back to its original index, and the weights are assigned accordingly (lines 8-10). In the classification stage, we assign the weights obtained earlier to the time-frequency features of different patches to obtain the final weighted features.

$$y = S(Classifer(\sum_{k=1}^{n} A_{in} \otimes P_w)), \tag{24}$$

where $A_{in} \otimes P_w$ denotes applying the calculated weights P_w to A_{in} and then summing over the dimensions of the patch. The Classifier$(\cdot)$ function represents an MLP comprising two Linear layers and LeakyReLU activations, while S stands for the Softmax function.

V. Experimental Setup

A. Datasets

Our experiments are conducted on three datasets denoted as $\mathcal{D}_{\text{ADS-B}}$, $\mathcal{D}_{\text{WiFi}}$, and $\mathcal{D}_{\text{Sig53}}$. The dataset partitions are presented in Table II, and the details of these datasets are listed as follows.

$\mathcal{D}_{ADS-B}$ [41] is an Automatic Dependent Surveillance-Broadcast (ADS-B) dataset that includes the raw real and imaginary parts (IQ) sample records collected from the on-board transponders. $\mathcal{D}_{WiFi}$ [42] is a WiFi dataset, collected from over-the-air transmissions from 16 USRP X310 transmitting radios with a different transmitter-receiver distance. We use the 62 feet dataset for the main wifi dataset, which is characterized by $\mathcal{D}_{WiFi}$ when not explicitly stated. Specifically, we also used the 2 feet and 20 feet datasets in our transfer learning experiments to validate the generalization performance of our model between the different distances. $\mathcal{D}_{Sig53}$ [43] is a modulation recognition dataset that consists of 5 million synthetic samples from 53 different signal classes and expert-selected impairments. To verify the performance on complex signals, we partitioned the dataset by taking out the clean part of the dataset of 2000 samples per category.

B. Implementation Details

During pre-training, the network was optimized using AdamW [44] with a learning rate of 1×10^{-3} for 300 epochs. During fine-tuning, we employed Adam [45] with a learning rate of 1×10^{-4} for 100 epochs. The temperature parameter was set to $\tau = 0.05$. The MLP architectures for the APA and the frequency-conversion module are listed in Appendix. For the time encoder, we follow the previous study [43] and apply XCiT-T12 [46] for the temporal representation encoding.

C. Experiment Setup

Experiments were conducted on a workstation equipped with three NVIDIA RTX A5000 GPUs. We used a batch size of 128 and a contrastive feature dimensionality of 1,920. Evaluation metrics comprise Top-1 and Top-5 accuracy. To assess the effectiveness of the proposed self-supervised learning approach, we report results under three evaluation protocols: linear evaluation (LE), semi-supervised learning (SSL), and transfer learning (TL).

D. Comparison Methods

To validate the effectiveness of the proposed approach, we conducted a comparative analysis with 12 distinct methods. For all baseline methods, we used the released code when available, and re-implemented those without official code in [49]. We retained the parameter settings from each baseline's original implementation and ensured identical input signal patches to guarantee a fair comparison. Detailed descriptions of baseline methods are listed below:

1) *SL*: Supervised learning (SL) simply utilizes the labeled set to train the model. In the LE evaluation method, we use 10 and 1 sample per category in $\mathcal{D}_{ADS-B}$ and $\mathcal{D}_{WiFi}$ and the pre-training setting in $\mathcal{D}_{Sig53}$ to train the model. In the SSL evaluation method, we use a setup of fine-tuning in Table II to do a supervised learning comparison experiment.
2) *SimCLR*: SimCLR [10] is a self-supervised learning framework that leverages data augmentation and contrastive loss to train models by maximizing the agreement between differently augmented views of the same image.
3) *MoCo variants*: MoCo v1 [12] is a contrastive learning framework using a dynamic dictionary and a moving-averaged encoder for consistent key building in unsupervised representation learning. MoCo v2 [21] enhances this by adding stronger data augmentations, a robust MLP projection head, and a larger queue size for better representation learning.
4) *BYOL*: BYOL [47] is a self-supervised image learning framework that requires no negative samples. Through collaborative learning between dual networks (an online network and a target network), the online network predicts the representation output of the target network for

TABLE III: The result of semi-supervised learning.

Setting	Label fraction											
	1%	10%	1%	10%	1%	10%	1%	10%	1%	10%	1%	10%
Method \ Dataset	$\mathcal{D}_{ADS-B}$		$\mathcal{D}_{WiFi}$		$\mathcal{D}_{Sig53}$		$\mathcal{D}_{ADS-B}$		$\mathcal{D}_{WiFi}$		$\mathcal{D}_{Sig53}$	
ACC	**Top 1 (%)**						**Top 5 (%)**					
SL	23.40	54.20	92.31	**99.96**	27.48	56.67	62.50	**100.00**	100.00	99.96	67.51	77.20
SimCLR(PMLR,2020) [10]	14.80	24.40	50.00	90.31	14.87	18.11	55.00	62.60	83.33	100.00	49.92	54.24
MoCo v1(CVPR,2020) [12]	12.30	14.30	33.97	41.53	20.40	27.93	54.10	51.90	83.33	98.05	61.86	67.04
MoCo v2(CVPR,2020) [21]	11.00	12.10	16.67	16.67	19.43	42.88	52.50	54.20	97.55	83.33	59.19	71.78
BYOL(NeurIPS,2020) [47]	14.20	16.30	17.67	18.37	2.15	3.01	55.40	65.00	83.43	83.45	9.72	11.08
SwAv(NeurIPS,2020) [48]	14.80	20.09	83.10	89.40	36.80	39.90	54.30	64.50	100.00	100.00	70.10	70.80
ZeroCL(ICLR,2021) [13]	37.15	67.02	48.62	97.02	51.98	52.24	81.88	96.21	95.65	100.00	71.79	73.19
DINO(ICCV,2021) [31]	15.90	18.50	87.50	92.03	28.58	36.80	56.30	64.60	100.00	100.00	66.79	70.01
ARB(CVPR,2022) [25]	36.80	66.30	68.90	89.10	53.73	55.12	81.23	93.79	97.60	100.00	72.54	76.03
MAE(CVPR,2022) [32]	15.00	20.20	28.35	31.32	14.85	23.48	58.80	60.40	99.68	99.78	47.84	60.64
SelfPatch(CVPR,2022) [33]	11.80	16.90	20.40	23.30	11.30	20.88	49.50	50.40	84.28	85.53	41.02	62.41
SA2SEI(TIFS,2023) [28]	24.80	67.30	83.85	87.53	20.02	26.02	63.60	95.30	100.00	100.00	54.72	58.38
FOCAL((NeurIPS,2023) [8]	21.80	46.50	93.73	98.18	43.68	51.13	65.60	98.10	100.00	100.00	68.11	71.81
DMCP (Ours)	**40.00**	**73.90**	**94.63**	99.79	**54.89**	**58.70**	**83.89**	97.67	**100.00**	**100.00**	**73.89**	**80.67**

Note: **bold** denotes the best, underline denotes the second best

TABLE IV: The result of linear evaluation for the three datasets.

Method \ Dataset	$\mathcal{D}_{ADS-B}$	$\mathcal{D}_{WiFi}$	$\mathcal{D}_{Sig53}$
ACC	**Top 1 (%)**		
SL	**96.50**	92.31	**73.60**
SimCLR(PMLR,2020) [10]	29.70	31.30	23.82
MoCo v1(CVPR,2020) [12]	48.30	46.67	29.44
MoCo v2(CVPR,2020) [21]	49.26	48.33	21.65
BYOL(NeurIPS,2020) [47]	21.20	19.13	2.23
SwAv(NeurIPS,2020) [48]	32.70	**94.65**	41.98
ZeroCL(ICLR,2021) [13]	90.30	51.40	42.17
DINO(ICCV,2021) [31]	31.60	92.25	34.62
ARB(CVPR,2022) [25]	89.20	57.70	41.81
MAE(CVPR,2022) [32]	11.20	29.10	25.45
SelfPatch(CVPR,2022) [33]	15.00	31.83	23.93
SA2SEI(TIFS,2023) [28]	77.60	84.30	34.56
FOCAL((NeurIPS,2023) [8]	47.50	93.41	42.06
DMCP (Ours)	93.37	94.33	51.97
ACC	**Top 5 (%)**		
SL	**100.00**	100.00	**83.22**
SimCLR(PMLR,2020) [10]	72.10	83.26	65.65
MoCo v1(CVPR,2020) [12]	78.70	88.23	68.16
MoCo v2(CVPR,2020) [21]	80.60	83.33	64.78
BYOL(NeurIPS,2020) [47]	70.50	83.33	9.71
SwAv(NeurIPS,2020) [48]	80.80	100.00	69.87
ZeroCL(ICLR,2021) [13]	99.90	88.38	65.74
DINO(ICCV,2021) [31]	76.80	100.00	69.58
ARB(CVPR,2022) [25]	99.70	99.08	63.74
MAE(CVPR,2022) [32]	52.60	96.48	64.87
SelfPatch(CVPR,2022) [33]	52.10	93.87	68.08
SA2SEI(TIFS,2023) [28]	99.00	100.00	70.45
FOCAL((NeurIPS,2023) [8]	99.60	100.00	74.10
DMCP (Ours)	100.00	**100.00**	78.80

Note: **bold** denotes the best, underline denotes the second best

the same image under different augmented views, thereby learning effective image representations.

5) *SwAv*: SwAv [48] is a self-supervised learning framework based on clustering that achieves representation learning by exchanging the clustering assignments of different augmented views.
6) *ZeroCL*: ZeroCL [13] is proposed to prevent trivial solutions in contrastive learning by using instance-wise and feature-wise whitening.
7) *DINO*: DINO [31] is a self-supervised learning framework based on a self-distillation mechanism, enabling unsupervised representation learning through a teacher-student network architecture.
8) *ARB*: Align Representations with Base (ARB) [25] is a technique designed to improve model performance by aligning learned representations with a base set of reference representations.
9) *MAE*: MAE [32] is a self-supervised learning framework based on masked reconstruction. Its core idea is to learn visual representations by randomly masking image patches and reconstructing the missing content.
10) *SelfPatch*: SelfPatch [33] is a self-supervised learning framework based on block-level self-distillation, achieving representation learning by mining semantic similarities among blocks within images.
11) *SA2SEI*: SA2SEI [28] is a state-of-the-art few-shot Specific Emitter Identification (FS-SEI) method utilizing self-supervised learning and innovative adversarial augmentation (Adv-Aug) to reduce label dependence on auxiliary datasets.
12) *FOCAL*: FOCAL [8] is a contrastive learning framework that excels in extracting both shared and modality-exclusive features from multimodal time-series signals. It creates an orthogonal latent space, separating shared and private features, and imposes a temporal structural constraint to ensure the consistency of temporal information.

VI. Empirical Result

To validate the effectiveness and efficiency of our DMCP, we conduct comprehensive experiments on three datasets to answer the following seven research questions:

- RQ1: Does our DMCP demonstrate superior performance compared to baselines across diverse datasets?

TABLE V: Analysis of the transferability across $\mathcal{D}_{ADS-B}, \mathcal{D}_{WiFi}, \mathcal{D}_{Sig53}$ datasets.

Origin Dataset	$\mathcal{D}_{Wifi-62}$				$\mathcal{D}_{ADS-B}$			
Target Dataset	$\mathcal{D}_{Wifi-2}$	$\mathcal{D}_{Wifi-20}$	$\mathcal{D}_{ADS-B}$	$\mathcal{D}_{Sig53}$	$\mathcal{D}_{Wifi-2}$	$\mathcal{D}_{Wifi-20}$	$\mathcal{D}_{Wifi-62}$	$\mathcal{D}_{Sig53}$
Accuracy / **Method**	**TOP 1 (%)**							
SimCLR(PMLR,2020) [10]	39.78	40.37	27.10	11.90	27.48	22.67	33.52	8.25
MoCo v1(CVPR,2020) [12]	19.58	17.43	13.50	7.84	23.13	48.18	23.60	12.98
MoCo v2(CVPR,2020) [21]	34.12	35.98	13.10	9.34	16.67	16.67	32.42	6.15
BYOL((NeurIPS,2020) [47]	49.55	49.82	13.00	12.07	36.45	33.37	86.69	20.27
SwAv((NeurIPS,2020) [48]	**75.13**	53.35	41.90	19.55	18.45	41.07	73.90	12.07
ZeroCL(ICLR,2021) [13]	26.55	33.58	20.90	21.56	51.20	64.00	95.12	21.78
DINO((ICCV,2021) [31]	47.85	57.18	28.10	24.63	50.93	53.60	89.63	26.77
ARB(CVPR,2022) [25]	35.05	37.25	22.60	20.98	50.78	61.87	95.63	21.93
MAE((CVPR,2022) [32]	17.92	17.60	14.40	17.34	18.50	17.95	24.62	18.54
SelfPatch((CVPR,2022) [33]	18.43	18.20	14.70	18.23	18.07	19.32	23.48	19.77
SA2SEI(TIFS,2023) [28]	47.83	58.92	77.60	22.35	38.45	44.69	62.05	21.66
FOCAL((NeurIPS,2023) [8]	54.48	67.36	22.70	21.65	50.57	40.75	94.38	22.24
DMCP (Ours)	62.85	**71.70**	**84.40**	**22.75**	**71.12**	**83.63**	**99.80**	**23.18**
Accuracy / **Method**	**TOP 5 (%)**							
SimCLR(PMLR,2020) [10]	87.08	85.32	75.60	47.68	83.40	78.75	91.78	30.85
MoCo v1(CVPR,2020) [12]	83.33	83.33	52.30	40.18	93.18	100.00	83.42	49.26
MoCo v2(CVPR,2020) [21]	98.28	98.73	54.50	39.48	83.33	84.88	100.00	26.89
BYOL((NeurIPS,2020) [47]	100.00	100.00	56.70	44.98	99.82	99.18	100.00	64.67
SwAv((NeurIPS,2020) [48]	100.00	100.00	91.10	61.84	94.30	90.88	100.00	55.46
ZeroCL(ICLR,2021) [13]	95.55	87.00	60.30	38.77	100.00	100.00	100.00	40.56
DINO((ICCV,2021) [31]	100.00	99.60	74.70	68.36	100.00	100.00	100.00	**67.69**
ARB(CVPR,2022) [25]	93.63	93.12	62.40	37.06	100.00	100.00	100.00	65.44
MAE((CVPR,2022) [32]	85.18	86.32	57.10	54.44	84.98	84.97	91.62	57.13
SelfPatch((CVPR,2022) [33]	84.17	85.47	52.00	58.96	85.50	85.23	89.43	60.09
SA2SEI(TIFS,2023) [28]	100.00	100.00	99.10	64.31	95.01	94.55	99.76	63.77
FOCAL((NeurIPS,2023) [8]	100.00	100.00	74.90	**69.49**	100.00	100.00	100.00	67.44
DMCP (Ours)	**100.00**	**100.00**	**99.40**	48.00	**100.00**	**100.00**	**100.00**	42.51

- RQ2: How effectively does the DMCP framework transfer learned features to downstream tasks across diverse domains compared to conventional pre-training schemes?
- RQ3: How does the optimization function within DMCP, e.g., time-based contrastive loss, time-frequency contrastive loss, impact the performance of recognition?
- RQ4: How effective are the attentive patch aggregation Module strategies designed in DMCP?
- RQ5: Does DMCP pre-trained latent space present reasonable discriminative ability compared to other contrastive learning frameworks?
- RQ6: How does adjusting various hyperparameters, particularly the patch length, and projection embedding dimension, impact the performance and training stability of DMCP?
- RQ7: How efficient is DMCP compared to other contrastive learning frameworks?

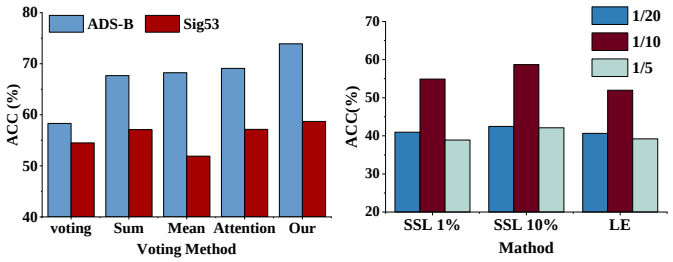
A. Performance Comparison (RQ1)

Linear evaluation. We follow the widely used linear evaluation protocol [50], [51] to fix the feature extractor and add one layer of trainable MLP for class projection. The linear evaluation can effectively reflect the distinguish-ability of the learned representation space. Table IV presents the linear evaluation performance of our framework compared to the baseline models. The results demonstrate that the robust representation space trained by patch-level cross-view optimization captures more fine-grained signal patterns, allowing for fast and accurate tuning. Specifically, the proposed framework exhibits superior performance compared with existing methods and shows the potential to be competitive with purely supervised learning. Particularly noteworthy is the outstanding performance on the $\mathcal{D}_{ADS-B}$ and $\mathcal{D}_{WiFi}$ datasets. It achieves an impressive 94.33% Top-1 accuracy on $\mathcal{D}_{WiFi}$, representing a 2.02% improvement over the performance of supervised learning. For the $\mathcal{D}_{ADS-B}$ and $\mathcal{D}_{Sig53}$ datasets, the proposed framework may lag behind supervised learning. This could be attributed to the large number of samples provided in LE's experiments, allowing supervised learning to obtain a more precise discriminative representation space.

Semi-supervised learning. We follow the [52] and sample 1% or 10% of the labelled training datasets in a class-balanced way (as demonstrated in Table II). We simply fine-tune the whole base network on the labelled data without regularization. Table III shows the comparisons of our results against other CL methods across three benchmark datasets. Again, our approach significantly improve over state-of-the-art methods with both 1% and 10% of the labels. Specifically, we can see that our approach can gain around 1% $\sim$ 8% top-1 and top-5 accuracy improvement across 1% and 10% fine-tune settings. Interestingly, *fine-tuning our pre-trained framework on 1% and 10% labels is also significantly better than training from scratch.*

TABLE VI: Ablation study of optimization function.

Loss / Dataset		Semi		
$\mathcal{L}_{tcl}$	$\mathcal{L}_{tfcl}$	$\mathcal{D}_{ADS-B}$	$\mathcal{D}_{Wifi}$	$\mathcal{D}_{Sig53}$
✓	✗	23.20	60.38	34.78
✗	✓	38.65	67.66	40.12
✓	✓	**40.00**	**94.63**	**54.89**
Loss / Dataset		**LE**		
✓	✗	73.50	88.23	38.35
✗	✓	88.09	90.23	45.66
✓	✓	**93.37**	**94.33**	**51.97**



(a) Strategy Analysis (b) Patch length l Analysis

Fig. 4: Ablation study results on patch aggregation strategy and patch length parameter.

B. Analysis of the Transferability (RQ2)

Transferability indicates the ability to learn invariant features across domains. We follow the [10] to evaluate the transfer learning performance across 5 datasets (three benchmark datasets and two additional datasets for WiFi signal under different transmission distances) in linear evaluation (fixed feature extractor). The linear evaluation across different datasets effectively reflects the generalization of the learned representations to new, unseen data from different domains. Table V shows the cross-dataset transfer learning results of the proposed framework. Specifically, our framework surpasses all other methods in Top-1 accuracy, demonstrating the generalizable representation space learned by the proposed learning paradigm. Furthermore, *it shows that leveraging temporal-frequency contrastive learning alongside the attentive patch ensemble could construct a latent space with better discriminability as well as transferability.*

C. Analysis of Optimization Function (RQ3)

To verify the contribution of time-based and time-frequency contrastive optimization, we performed ablation experiments in both linear evaluation (fixed feature extractor) and fine-tuning settings (semi-supervised with 1% labels)in the larges signal dataset $\mathcal{D}_{Sig53}$. Table VI shows that the time-frequency contrastive optimization achieves a 2% to 15% improvement compared to time-based contrastive optimization. Such results demonstrate that explicitly considering the frequency domain can provide an understanding of time series behaviour that cannot be directly captured solely in the time domain. Furthermore, the simultaneous optimization of time-based and time-frequency contrastive methods provides an additional 2% to 6% performance gain compared to time-frequency contrastive alone. This demonstrates that both invariant temporal-reliance and frequency-reliance information contribute to constructing a more discriminative representation space.

D. Analysis of Attentive Patch Aggregation Module (RQ4)

To verify the contribution of our proposed attentive patch aggregation module during the fine-tuning and inference stage, we conduct experiments on $\mathcal{D}_{ADS-B}$ and $\mathcal{D}_{Sig53}$ dataset under 10% labels semi-supervised setting. We compare the attentive patch aggregation module with 1) Hard voting: performing the voting to the class with the most patch predicted. 2) Sum: performing sum operation to the class prediction probability and normalized to assign class label with maximum probability. 3) Mean: performing average operations to the class prediction probability and normalizing them to assign a class label with maximum probability. 4) Attention: Apply the attention mechanism to directly generate attentive class probability and sign a class label with maximum probability.

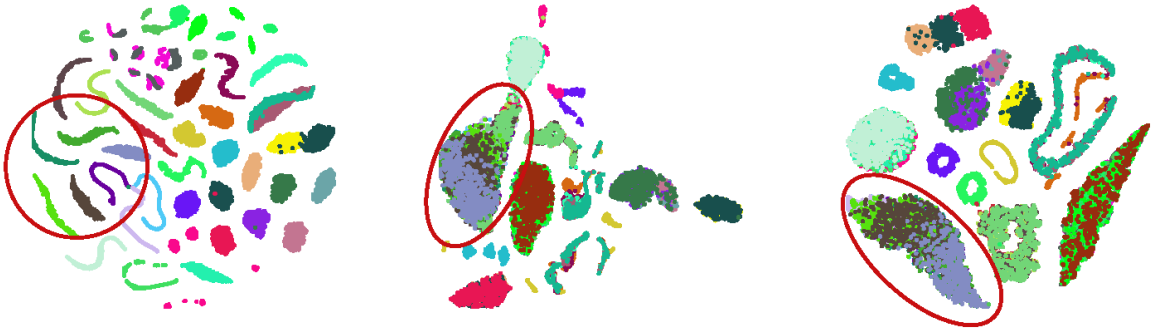
Figure 4a presents the results of five aggregation strategies. The voting baseline does not account for correlations between patches, resulting in poor classification accuracy. Adding the attention mechanism yields marginal improvements over the Sum and Mean aggregation methods, indicating that Sum and Mean aggregation also perform a similar attention process to a certain extent. The proposed attentive patch aggregation module combines the noise robustness of voting (ensemble) with the nuanced weighting of attention, achieving nearly 4.82% and 1.55% improvements in $\mathcal{D}_{ADS-B}$ and $\mathcal{D}_{Sig53}$ respectively.

E. Analysis of Patch Length l Parameter (RQ5)

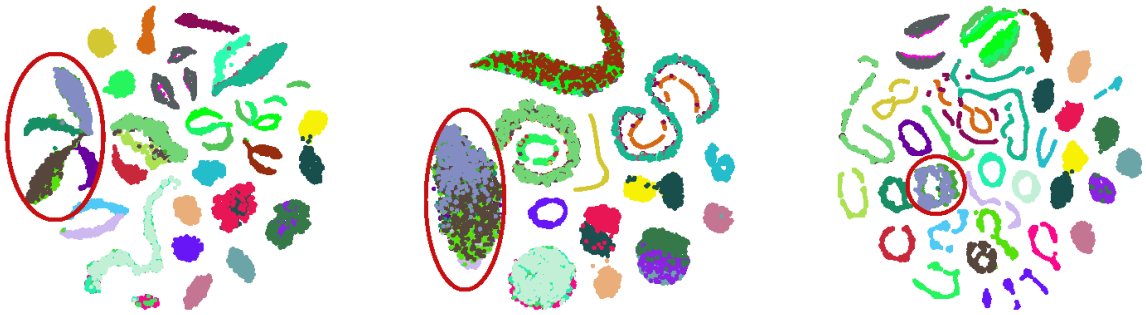
The patch length determines the receptive field that the network can capture during the training process. To analyze the effect of patch length, we conduct experiments on different data lengths (i.e., 1/5, 1/10, and 1/20 of the original length) on the $\mathcal{D}_{\text{Sig53}}$ dataset. Figure 4b presents the results of the patch length effect, and we find that a patch length of one-tenth yields the best results across all evaluation methods compared to the other patch lengths. Interestingly, we observe that smaller patches do not always lead to better performance. Specifically, the 1/20 patch length results in nearly a 20% decrease in accuracy compared to the 1/10 patch length. One possible explanation is that patches that are too small can lead to insufficient contextual information, resulting in poor representation construction.

F. Latent Visualization (RQ5)

To further verify the effectiveness of our DMCP framework–(whether a better intra-class concentration is provided), we applied t-SNE to generate visualizations for latent features on the Sig53 dataset. Figure 5 illustrates the t-SNE plot for six state-of-the-art contrastive learning approaches. We can witness that the clusters of latent features under DMCP pre-training (Figure 5a) are more distinct than others. Furthermore, we observe that

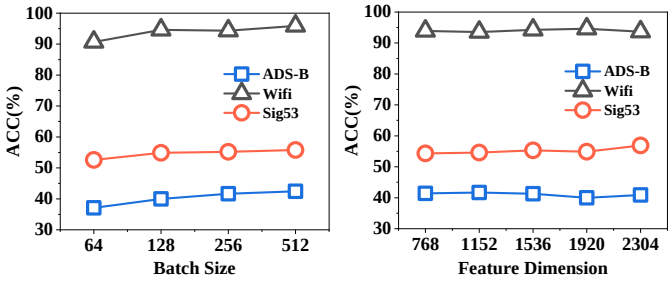


(a) DMCP (b) SimCLR (c) SA2SEI



(d) ZeroCL (e) ARB (f) FOCAL

Fig. 5: Experimental results of the Sig53 dataset in linear evaluation (t-SNE).



(a) Batch Size Analysis (b) Feature Dimension Analysis

Fig. 6: Ablation study results on training batch size and contrastive feature dimension.

the DMCP pre-training latent features present more organized results (with intra-class and inter-class concentration) than the other contrastive pre-trained latent features which demonstrate the effectiveness of our DMCP framework.

G. Analysis of Training Stability (RQ6)

To verify the training stability of our model under varying training parameters, we conduct experiments using four different batch sizes (i.e., 64, 128, 256, 512) and five different projection feature dimensions (i.e., 768, 1152, 1536, 1920, 2304) on three benchmark datasets. Figure 6a and 6b present that our model remains relatively stable across different batch sizes and feature dimensions. Specifically, larger batch sizes would provide more diverse negative samples in each batch, which improves the robustness and quality of the learned representations [10].

H. Insight on the efficiency of patch contrastive (RQ7)

From the empirical experiments, we observe that the proposed patch contrastive learning provides faster convergence and a better representation space. To verify this phenomenon, we conduct experiments on $\mathcal{D}_{ADS-B}$ to compare the training efficiency of patch contrastive learning. The experiment is configured to train for the same number of epochs (i.e., 300) and obtain classification results under linear evaluation. To quantify the training and tuning efficiency of baseline methods and the proposed DMCP, we define three complementary metrics. Let A denote the measured accuracy, T_{train} the pretraining time, and T_{tune} the fine-tuning time.

1) **Training Efficiency (TRE)**: $\text{TRE} = A/T_{\text{train}}$, which measures accuracy obtained per unit pretraining time.
2) **Tuning Efficiency (TUE)**: $\text{TUE} = A/T_{\text{tune}}$, which measures accuracy obtained per unit fine-tuning time.
3) **H-index (balanced efficiency)**: the harmonic mean of TRE and TUE,

$$H = \frac{2 * \text{TRE} * \text{TUE}}{\text{TRE} + \text{TUE}},$$

TABLE VII: Model Efficiency Analysis on D_{ADS-B} dataset.

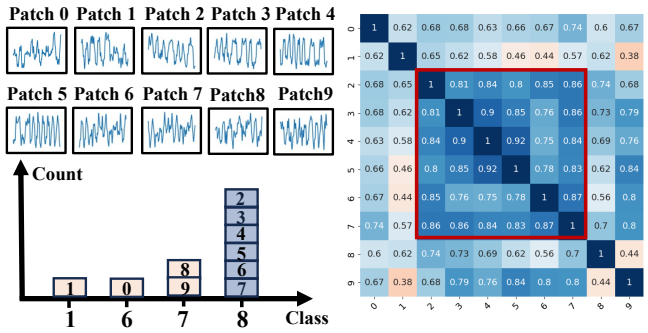
Model	Pretraining Time/s	Tuning Time/s	Accuracy	TUE (Tunining Efficiency)	TRE (Training Efficiency)	H-Index
SimCLR(PMLR,2020) [10]	20273.39	260.83	29.70	0.1139	0.0015	0.0029
MoCo v1(CVPR,2020) [12]	18347.97	743.50	48.30	0.0650	0.0026	0.0051
MoCo v2(CVPR,2020) [21]	**9089.06**	446.62	49.26	0.1103	0.0054	0.0103
BYOL((NeurIPS,2020) [47]	14361.01	**179.95**	21.20	0.1178	0.0015	0.0029
SwAv((NeurIPS,2020) [48]	10947.81	181.42	32.70	0.1802	0.0030	0.0059
ZeroCL(ICLR,2021) [13]	56959.92	260.61	90.30	**0.3465**	0.0016	0.0032
DINO((ICCV,2021) [31]	15012.14	261.47	31.60	0.1209	0.0021	0.0041
ARB(CVPR,2022) [25]	79510.28	276.59	89.20	0.3225	0.0011	0.0022
MAE((CVPR,2022) [32]	9795.68	1306.53	11.20	0.0086	0.0011	0.0020
SelfPatch((CVPR,2022) [33]	14668.90	1028.14	15.00	0.0146	0.0010	0.0019
SA2SEI(TIFS,2023) [28]	29706.94	1291.35	77.60	0.0601	0.0026	0.0050
FOCAL((NeurIPS,2023) [8]	13185.16	306.17	47.50	0.1551	0.0036	0.0070
DMCP (Ours)	10124.46	572.33	**93.37**	0.1631	**0.0092**	**0.0175**

which provides a balanced summary of training and tuning efficiency and prevents one component from dominating the combined score.

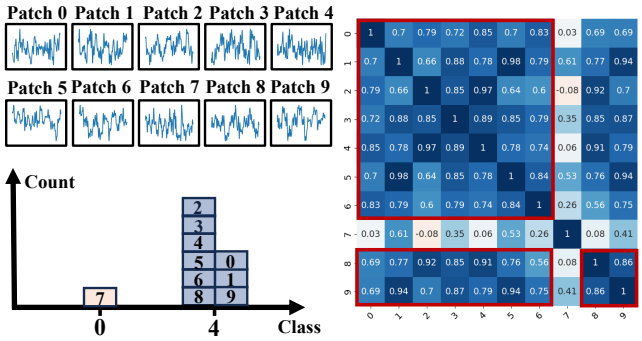
Table VII reports detailed evaluation results for all methods on the ADS-B dataset. The proposed DMCP framework achieves the highest values in the TRE and H-index columns, indicating that patch-level contrastive learning substantially improves training efficiency when processing long sequential signals. DMCP ranks fourth in tuning time, which is attributable to the voting aggregation step that currently lacks GPU acceleration; nevertheless, the observed slowdown is modest, and the overall trade-off remains competitive in the comprehensive evaluation. These findings suggest that the patch contrastive design delivers strong gains in representation quality and training efficiency, and point to implementation optimizations (e.g., GPU-accelerated voting) as a straightforward path to further improve tuning latency in future work.

I. Ablation Study on APA Module Interpretation.

To further demonstrate how the Attentive Patch Aggregation (APA) module contributes to robust inference, we performed a focused interpretative experiment and discuss the results in two complementary parts. In Fig. 7 the patch-feature heatmap reveals a clearly coherent group—patches 2–7—with pairwise correlations consistently above 0.75; these mutually consistent patches jointly drive the model toward the correct class (class 8), and the APA voting mechanism assigns them higher influence while down-weighting isolated, less-consistent patches, which prevents a single biased patch from flipping the prediction. Furthermore, Fig. 7b shows a case where most patches again form a coherent agreement on the same label index but one patch (patch 7) exhibits weak correlation with the others; APA's attention-driven aggregation treats that low-correlation patch as an outlier and effectively reduces its impact on the final decision, so the ensemble vote reflects the consensus rather than the biased signal. Taken together, these visualizations illustrate APA's core behavior: it elevates coherent, mutually supportive patch evidence and suppresses inconsistent, potentially misleading patch contributions, thereby improving prediction stability on long sequential inputs; this also suggests practical follow-ups such as implementing a GPU-accelerated aggregation kernel and exploring adaptive attention thresholds to further reduce tuning overhead while preserving robustness.



(a) Interpretation of signal patches correlation with APA voting mechanism on the sample belongs to Class 8



(b) Interpretation of signal patches correlation with APA voting mechanism on the sample belongs to Class 4

Fig. 7: Ablation study results on the APA module. Highly correlated samples are consistently voted into the same correct classes, thereby preventing biased patches from producing skewed predictions.

J. Ablation Study on Frequency Influence

Testing sensitivity to signal frequency is important because real-world signal captures vary across receivers and collection setups (different front-end filters, down-sampling, or truncated captures), so a robust method should maintain performance under these practical variations. We therefore evaluated DMCP on a range of frequency rate (see Table VIII); the results

TABLE VIII: Ablation study of Signal Frequency Change Effects on ADS-B Dataset.

Dataset	Signal Length	Frequency	ACC(%)
$\mathcal{D}_{ADS-B}$	4800	8 Mhz	40.00
	2400	4 Mhz	39.90
	1600	2.7 Mhz	37.80
	1200	2 Mhz	39.10
	800	1.3 Mhz	34.10
	600	1 Mhz	34.30

show that accuracy remains largely stable for typical settings (e.g. 4800@8 MHz: 40.00%, 2400@4 MHz: 39.90%, 1600@2.7 MHz: 37.80%, 1200@2 MHz: 39.10%), indicating that the patch-level contrastive design is tolerant to moderate changes in frequency and temporal resolution. As expected, extreme down-sampling with very short captures (e.g. length = 600 at 1 MHz, and the 800@1.3 MHz case) leads to noticeable performance degradation (accuracies drop to $\sim$34%), which we attribute to the loss or distortion of critical signal components and reduced temporal context under these conditions. Overall, DMCP demonstrates good robustness to realistic sampling variations while showing the expected sensitivity when essential information is removed by aggressive down-sampling or truncation.

VII. Limitation and Future Work

Although the proposed DMCP contrastive learner demonstrates effective representation learning capabilities for signals, it *still struggles to handle different types of signals simultaneously.* Learning a unified representation (one-for-all) is a major direction for most foundation or contrastive learning models, which aim to learn generalizable world models from large-scale, diverse datasets. However, the proposed DMCP cannot effectively address this scenario due to the significant variations across signals, such as differences in length and modality. Therefore, designing a unified encoder remains an important direction for future work.

Furthermore, *the feature extraction ability of the proposed approach is not sufficient.* For instance, on datasets like Sig53, there remains a performance gap of about 20% compared to supervised learning. Mining meaningful time and frequency patterns is still a non-trivial task due to their non-interpretable nature. In future work, we plan to explore improved network architectures to extract more fine-grained time and frequency features.

VIII. Conclusion

In this paper, we propose a unified framework named Dual Modality Patch Contrastive (DMCP) to learn fine-grained representations from unlabeled lengthy signals and serve various downstream tasks. Our method achieves optimal performance by extracting local fine-grained temporal-frequency information from diverse signal patches. However, there are still significant challenges for signal contrastive learning in complex data sets like Sig-53. In the future, we will utilize additional signal attributes to assist in contrastive learning.

References

[1] T. Li, Z. Hong, Q. Cai, L. Yu, Z. Wen, and R. Yang, "Bissiam: Bispectrum siamese network based contrastive learning for uav anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[2] C. Shahriar, S. Sodagari, and T. C. Clancy, "Performance of pilot jamming on mimo channels with imperfect synchronization," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 898–902.

[3] J. Su, P. Sun, Y. Jiang, Z. Wen, F. Guo, Y. Wu, Z. Hong, H. Duan, Y. Huang, R. Ranjan, and Y. Zheng, "A semantic-consistent few-shot modulation recognition framework for iot applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 5, pp. 8100–8111, 2025.

[4] T. Li, Z. Wen, Y. Long, Z. Hong, S. Zheng, L. Yu, B. Chen, X. Yang, and L. Shao, "The importance of expert knowledge for automatic modulation open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 730–13 748, 2023.

[5] Y. Zhu, Z. Wen, X. Li, X. Shi, X. Wu, H. Dong, and J. Chen, "Chatnav: Leveraging llm to zero-shot semantic reasoning in object navigation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 3, pp. 2369–2381, 2025.

[6] K. Davaslioglu, S. Boztaş, M. C. Ertem, Y. E. Sagduyu, and E. Ayanoglu, "Self-supervised rf signal representation learning for nextg signal classification with deep learning," *IEEE Wireless Communications Letters*, vol. 12, no. 1, pp. 65–69, 2023.

[7] D. Liu, P. Wang, T. Wang, and T. Abdelzaher, "Self-contrastive learning based semi-supervised radio modulation classification," in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 2021, pp. 777–782.

[8] S. Liu, T. Kimura, D. Liu, R. Wang, J. Li, S. Diggavi, M. Srivastava, and T. Abdelzaher, "Focal: Contrastive learning for multimodal time-series sensing signals in factorized orthogonal latent space," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[9] Y. Wang, Y. Han, H. Wang, and X. Zhang, "Contrast everything: A hierarchical contrastive framework for medical time-series," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *in Proceedings of International Conference on Machine Learning*, 2020, pp. 1597–1607.

[11] D. Luo, W. Cheng, Y. Wang, D. Xu, J. Ni, W. Yu, X. Zhang, Y. Liu, Y. Chen, H. Chen *et al.*, "Time series contrastive learning with information-aware augmentations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4534–4542.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[13] S. Zhang, F. Zhu, J. Yan, R. Zhao, and X. Yang, "Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning," in *International Conference on Learning Representations*, 2021.

[14] P. Liu, B. Wu, N. Li, T. Dai, F. Lei, J. Bao, Y. Jiang, and S.-T. Xia, "Wftnet: Exploiting global and local periodicity in long-term time series forecasting," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

[15] J. Kim, S. Cho, S. Hwang, W. Lee, and Y. Choi, "Enhancing lpi radar signal classification through patch-based noise reduction," *IEEE Signal Processing Letters*, vol. 31, pp. 716–720, 2024.

[16] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459–469.

[17] J. Yin, H. Wu, and S. Sun, "Effective sample pairs based contrastive learning for clustering," *Information Fusion*, vol. 99, p. 101899, 2023.

[18] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[19] C. Hao, X. Wan, D. Feng, Z. Feng, and X.-G. Xia, "Satellite-based radio spectrum monitoring: Architecture, applications, and challenges," *IEEE Network*, vol. 35, no. 4, pp. 20–27, 2021.

[20] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[21] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[22] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," *arXiv preprint arXiv:2010.04592*, 2020.
[23] R. Zhu, B. Zhao, J. Liu, Z. Sun, and C. W. Chen, "Improving contrastive learning by visualizing feature transformation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 306–10 315.
[24] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 798–21 809, 2020.
[25] S. Zhang, L. Qiu, F. Zhu, J. Yan, H. Zhang, R. Zhao, H. Li, and X. Yang, "Align representations with base: A new approach to self-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 600–16 609.
[26] J. Gong, X. Xu, and Y. Lei, "Unsupervised specific emitter identification method using radio-frequency fingerprint embedded infogan," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2898–2913, 2020.
[27] B. Liu, H. Yu, J. Du, Y. Wu, Y. Li, Z. Zhu, and Z. Wang, "Specific emitter identification based on self-supervised contrast learning," *Electronics*, vol. 11, no. 18, p. 2907, 2022.
[28] C. Liu, X. Fu, Y. Wang, L. Guo, Y. Liu, Y. Lin, H. Zhao, and G. Gui, "Overcoming data limitations: a few-shot specific emitter identification method using self-supervised learning and adversarial augmentation," *IEEE Transactions on Information Forensics and Security*, 2023.
[29] Z. Wu, W. Cao, D. Bi, and J. Pan, "Clipc: Contrastive learning-based radar signal intra-pulse clustering," *IEEE Internet of Things Journal*, 2023.
[30] Z. Wen, Y. Ye, J. Su, T. Li, J. Wan, S. Zheng, Z. Hong, S. He, H. Duan, Y. Li *et al.*, "Unraveling complexity: An exploration into the large-scale multi-modal signal processing," *IEEE Intelligent Systems*, 2024.
[31] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.
[32] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.
[33] S. Yun, H. Lee, J. Kim, and J. Shin, "Patch-level representation learning for self-supervised vision transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8354–8363.
[34] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based rf fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, 2019.
[35] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, 2019.
[36] S. M. Weiss, N. Indurkhya, T. Zhang, and F. Damerau, *Text mining: predictive methods for analyzing unstructured information*. Springer Science & Business Media, 2010.
[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[38] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
[39] P. Sun, J. Su, Z. Wen, Y. Zhou, Z. Hong, S. Yu, and H. Zhou, "Boosting signal modulation few-shot learning with pre-transformation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
[40] P. Heckbert, "Fourier transforms and the fast fourier transform (fft) algorithm," *Computer Graphics*, vol. 2, no. 1995, pp. 15–463, 1995.
[41] T. Ya, L. Yun, Z. Haoran, J. Zhang, W. Yu, G. Guan, and M. Shiwen, "Large-scale real-world radio signal recognition with deep learning," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 35–48, 2022.
[42] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, "No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 165–178, 2019.
[43] L. Boegner, M. Gulati, G. Vanhoy, P. Vallance, B. Comar, S. Kokalj-Filipovic, C. Lennon, and R. D. Miller, "Large scale radio frequency signal classification," *arXiv preprint arXiv:2207.09918*, 2022.
[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[46] A. Ali, H. Touvron, M. Caron, P. Bojanowski, M. Douze, A. Joulin, I. Laptev, N. Neverova, G. Synnaeve, J. Verbeek *et al.*, "Xcit: Cross-covariance image transformers," *Advances in neural information processing systems*, vol. 34, pp. 20 014–20 027, 2021.
[47] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.
[48] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.
[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.
[50] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.
[51] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer, 2016, pp. 649–666.
[52] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1476–1485.

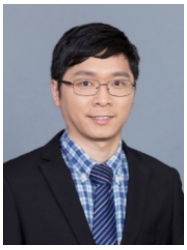
Jie Su received the B.S. degree in computer science and technology from China Jiliang University. Hangzhou, China, in 2017, the M.S. degree (Hons.) in data analytics from the University of Southampton, Southampton, U.K., in 2018, and the Ph.D. degree from Newcastle University, Newcastle upon Tyne, U.K., in 2023. He is currently an Assistant Professor with the Institute of Cyberspace Security and the College of Information Engineering, Zhejiang University of Technology, Hangzhou. His research interests include deep learning, signal processing, and IoT security.

Yuting Jiang is currently pursuing the MSc. degree in College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. Her current research interests focus on deep learning applications , signal processing and generation.

Yuheng Ye received a master's degree in computer science and technology from the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His research interests include large-scale model training and fine-tuning, radio signal recognition. His current work focuses on large language models (LLMs) and intelligent agent systems.

Zhenyu Wen (Senior Member, IEEE) is currently a Tenure-tracked professor with the Institute of Cyberspace Security and College of Information Engineering, Zhejiang University of Technology, and is a postdoctoral researcher with the University of Science and Technology of China. His current research interests include IoT, crowd sources, AI systems, and cloud computing. For his contributions to the area of scalable data management for the Internet of Things. He was awarded the IEEE TCSC Award for Excellence in Scalable Computing (Early Career Researchers) in 2020.

Taotao Li received the Ph.D. degree in control science and engineering from Zhejiang University of Technology, Hangzhou, China, in 2025. Now he is a postdoctoral fellow at the School of Computer Science and Technology, Zhejiang University of Technology. His current research interests include data mining, deep learning, and RF fingerprint recognition.



Shibo He (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2012. He is currently a Professor with Zhejiang University.He was an Associate Research Scientist from March 2014 to May 2014, and a Postdoctoral Scholar from May 2012 to February 2014, with Arizona State University, Tempe, AZ, USA. From November 2010 to November 2011, he was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada. His research interests include Internet of Things, crowdsensing, big data analysis, etc. Prof. He serves on the editorial board for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Springer Peer-to-Peer Networking and Application and KSII Transactions on Internet and Information Systems, and is a Guest Editor for Elsevier Computer Communications and Hindawi International Journal of Distributed Sensor Networks. He was a Symposium Co-Chair for the IEEE GlobeCom 2020 and the IEEE ICC 2017, TPC Co-Chair for i-Span 2018, a Finance and Registration chair for ACM MobiHoc 2015, a TPC Co-Chair for the IEEE ScalCom 2014, a TPC Vice Co-Chair for ANT 2013 'IC2014, a Track CoChair for the Pervasive Algorithms, Protocols, and Networks of EUSPN 2013, a Web Co-Chair for the IEEE MASS 2013, and a Publicity Co-Chair of IEEE WiSARN 2010, and FCN 2014.

Xiaoqin Zhang (Senior Member, IEEE) received the PhD degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a professor with the Zhejiang University of Technology, China. His research interests include pattern recognition, computer vision, and machine learning. He has published more than 100 papers in international and national journals and international conferences, including IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, IEEE Transactions on Image Processing, ICCV, CVPR, NIPS, IJCAI, AAAI, ACM MM, and others.

Rajiv Ranjan is a Professor of Computing Science in the School of Computing at Newcastle University, United Kingdom. Before moving to Newcastle University, he was Julius Fellow (2013-2015),Senior Research Scientist and Project Leader in the Digital Productivity and Services Flagship of Commonwealth Scientific and Industrial Research Organization (CSIRO C Australian Government's Premier Research Agency). Prior to that he was a Senior Research Associate (Lecturer level B) in the School of Computer Science and Engineering, University of New South Wales (UNSW). Professor Ranjan has a PhD (2009) from the Department of Computer Science and Software Engineering at the University of Melbourne.

APPENDIX

The network structure of the frequency encoder and MLP layer in APA module are listed below:

nn.Conv1d(2, 64, 1, 1)
nn.BatchNorm1d(64)
nn.LeakyReLU(0.2, inplace=True)
nn.Conv1d(64, 1, 1, 1)
nn.BatchNorm1d(1)
nn.LeakyReLU(0.2, inplace=True)
nn.Flatten()
nn.Linear(192, 128)
nn.BatchNorm1d(128)
nn.LeakyReLU(0.2)
nn.Linear(128, 480)
nn.BatchNorm1d(480)
nn.LeakyReLU((0.2, inplace=True)

TABLE IX: The structure of the frequency encoder.

nn.Linear(1920*2, 1920*2)
nn.LeakyReLU(0.2, inplace=True)
nn.Linear(1920*2, num_classes=10)
nn.LeakyReLU(0.2, inplace=True)

TABLE X: MLP layer structure in APA module.

TABLE XI: Frequency Band

Dataset	**Frequency**
$\mathcal{D}_{Wifi}$	2.4 GHz
$\mathcal{D}_{ADS-B}$	1090 MHz
$\mathcal{D}_{Sig53}$	None