

Article

Prototypical Network with Residual Attention for Modulation Classification of Wireless Communication Signals

Bo Zang , Xiaopeng Gou , Zhigang Zhu , Lulan Long and Haotian Zhang

School of Electronic Engineering, Xidian University, Xi'an 710071, China; gxp@stu.xidian.edu.cn (X.G.); zgzh@xidian.edu.cn (Z.Z.); longlulan@xidian.edu.cn (L.L.); zhanghaotian@stu.xidian.edu.cn (H.Z.)

* Correspondence: bzang@mail.xidian.edu.cn

Abstract: Automatic modulation classification (AMC) based on data-driven deep learning (DL) can achieve excellent classification performance. However, in the field of electronic countermeasures, it is difficult to extract salient features from wireless communication signals under scarce samples. Aiming at the problem of modulation classification under scarce samples, this paper proposes a few-shot learning method using prototypical network (PN) with residual attention (RA), namely PNRA, to achieve the AMC. Firstly, the RA is utilized to extract the feature vector of wireless communication signals. Subsequently, the feature vector is mapped to a new feature space. Finally, the PN is utilized to measure the Euclidean distance between the feature vector of the query point and each prototype in this space, determining the type of the signals. In comparison to mainstream few-shot learning (FSL) methods, the proposed PNRA can achieve effective and robust AMC under the data-hungry condition.

Keywords: automatic modulation classification; few-shot learning; residual attention; prototypical network



check for updates

Citation: Zang, B.; Gou, X.; Zhu, Z.; Long, L.; Zhang, H. Prototypical Network with Residual Attention for Modulation Classification of Wireless Communication Signals. *Electronics* **2023**, *12*, 5005. <https://doi.org/10.3390/electronics12245005>

Academic Editor: Christos J. Bouras

Received: 10 November 2023

Revised: 8 December 2023

Accepted: 12 December 2023

Published: 14 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Automatic modulation classification (AMC) [1,2] is widely used for military and civilian fields, including cognitive radio, electronic warfare, and spectrum monitoring [3–5]. Acting as an intermediary between signal detection and demodulation, AMC significantly facilitates the efficient classification of the modulation types. Thus, it plays an important role in wireless communication systems.

AMC methods can be categorized into likelihood-based (LB) [6], feature-based (FB) [7], and deep learning (DL) [8,9]. LB methods rely on probability theory and Bayesian estimation theory, utilizing the probability density of the received signals to assess the potential likelihood hypotheses. FB methods need to extract statistical features from the received signals, such as time-frequency diagram [10], instantaneous phase [11], bispectrum [12], high-order cumulant [13], constellation diagram [14], cyclic spectrum [15], etc. DL methods [16–19] can automatically extract distinctive modulation features from the received signals in a data-driven manner. In recent years, DL methods have made remarkable achievements in signal processing, which has been with the mainstream pipeline for the AMC.

DL methods rely heavily on data-driven pattern recognition and feature extraction, requiring a substantial repository of well-labeled signal samples. However, in non-cooperative communication scenarios, the intricate and diverse nature of communication signals complicates the process of gathering and labeling a significant number of samples. Meanwhile, only a limited number of labeled samples are available and valuable. In such situations, the DL network suffers from difficult learning, leading to weak generalization performance and low classification accuracy for AMC.

Few-shot learning (FSL) methods can perform DL-based AMC tasks on small-scale datasets, which has been in the spotlight [20–25]. FSL, a learning paradigm inspired by

biological systems, aims to surpass the limitations of conventional DL networks regarding their capacity for generalization and adaptability across diverse scenarios. Its core idea involves enhancing learning algorithms or models by discerning relationships among interconnected tasks, mitigating challenges posed by limited data and subpar generalization in traditional deep learning. FSL encompasses a training phase where models accumulate experience from diverse tasks. In the subsequent testing phase, models rapidly adapt to new tasks with limited labeled samples. In [26], the authors proposed a spatial–temporal hybrid feature extraction network for few-shot AMC tasks, in which dual feature extraction branches are designed to map signals onto the spatial and temporal space, respectively, then a hybrid inference classifier is designed to fuse classification results from both branches. In [27], the authors proposed an attention relation network (ARN), which introduces channel and spatial attention, respectively, to learn a more effective feature representation of support samples. Experimental results show that the ARN method can achieve excellent performance for AMC even with only one support sample. In [28], the authors proposed an automatic modulation classification relation network (AMCRN) to distinguish different modulation types by comparing the feature similarity between test signals and prototypes of modulation types. Experimental results show that the architecture reached a maximum classification accuracy of 93%.

It can be concluded that the crucial aspect of FSL-based AMC lies in effectively representing signal modulation in the data-hungry scenario. However, the majority of the FSL-based AMC methods exhibit inadequate feature expression capabilities in such cases, constraining the improvement of related model performance. To address this issue, our contributions are summarized as follows:

- A novel few-shot learning network based on the prototypical network (PN) with residual attention (RA), named PNRA, is proposed for signal modulation classification.
- The RA is introduced to guide the learning of the PNRA, thereby enabling the extraction of salient features with strong intraclass similarity in data-hungry scenarios.
- Compared to mainstream FSL methods, the proposed PNRA can achieve effective and robust modulation classification performance.

The rest of this paper is organized as follows. Section 2 provides an overview of the signal model. Section 3 describes the PNRA model in detail. Section 4 presents some experimental results from various perspectives. Section 5 gives the conclusions.

2. An Overview of Signal Model

Assume the received signal $r(t)$ can be defined as:

$$r(t) = s(t) * h(t) + n(t) \quad (1)$$

where $s(t)$ represents the RF signal from the transmitter, $*$ denotes the convolution operation, $h(t)$ denotes the channel impulse response, and $n(t)$ denotes additive white Gaussian noise (AWGN).

For M-ary phase shift keying (PSK) signals, $s(t)$ can be expressed as:

$$s(t) = \sum_{n=-\infty}^{\infty} g(t - nT_s) \cos(2\pi f_c t + \phi_0 + \phi_m) \quad (2)$$

$$g(t) = \begin{cases} 1, & 1 \leq t \leq T_s \\ 0, & \text{others} \end{cases} \quad (3)$$

where T_s represents the symbol period. f_c denotes the carrier frequency. ϕ_0 and ϕ_m denote the initial phase and the modulation phase.

Then, the received signal $r(t)$ is quadrature sampled at the receiver, and the in-phase/quadrature (I/Q) signal $r(n)$ can be expressed as:

$$r(n) = [(r_1^I, r_1^Q), (r_2^I, r_2^Q), \dots, (r_N^I, r_N^Q)] \quad (4)$$

where N denotes the sampling length.

3. PNRA Model

3.1. The PNRA Model

The PNRA mainly consists of two core components: a feature extraction module based on residual attention and a metrics module based on the prototypical network. RA is utilized to extract key features from the constellation diagram of wireless communication signals. PN is utilized to train an effective classifier by measuring the Euclidean distance between the class prototype and the query point. The detailed architecture of PNRA is illustrated in Figure 1, conforming to the procedure as follows:

1. Denote the training set D_{train} containing K modulation types. For each modulation type, u_1 signals are randomly sampled to construct the support set S , and u_2 signals are selected from each modulation to form the query set Q . The support set is $S = \{S_1, \dots, S_k, \dots, S_K\} = \{(x_i, y_i)\}_{i=1}^{N_1}$, where $N_1 = K * u_1$, x_i denotes the modulated signal, and y_i denotes the modulation type. Similarly, the query set is $Q = \{Q_1, \dots, Q_k, \dots, Q_K\} = \{(x_j, y_j)\}_{j=1}^{N_2}$, where $N_2 = K * u_2$,
2. x_i and x_j are encoded by the RA f with learnable parameters ϕ to compute feature vectors. The detailed architecture of RA is shown in Figure 2.

$$v_i = f(x_i; \phi), \quad i = 1, 2, \dots, N_1 \quad (5)$$

$$v_j = f(x_j; \phi), \quad j = 1, 2, \dots, N_2 \quad (6)$$

where v_i and v_j are the feature vectors of signals in the support set and query set, respectively.

3. The corresponding prototype c_k for each modulation type is computed, which is the mean vector of the signals in the support sets:

$$c_k = \frac{1}{u_1} \sum_{x_i, y_i \in S_k} v_i, \quad i = 1, 2, \dots, u_1 \quad (7)$$

4. The PN has learnable parameters ψ . The detailed architecture of PN is shown in Figure 3. A distance function d is constructed, and the distance x_j from the query point in the query set to the prototype c_k is given as

$$p(y = k | x_j; \psi) = \frac{e^{-d(v_j, c_k)}}{\sum_{k=1}^K e^{-d(v_j, c_k)}} \quad (8)$$

$$d(v_j, c_k) = \|v_j - c_k\|^2 \quad (9)$$

where $j = 1, 2, \dots, N_2$.

5. The network parameters are updated by minimizing the loss function $loss_J$ [29]:

$$loss_J = -\log p(y = k | x_j; \psi) \quad (10)$$

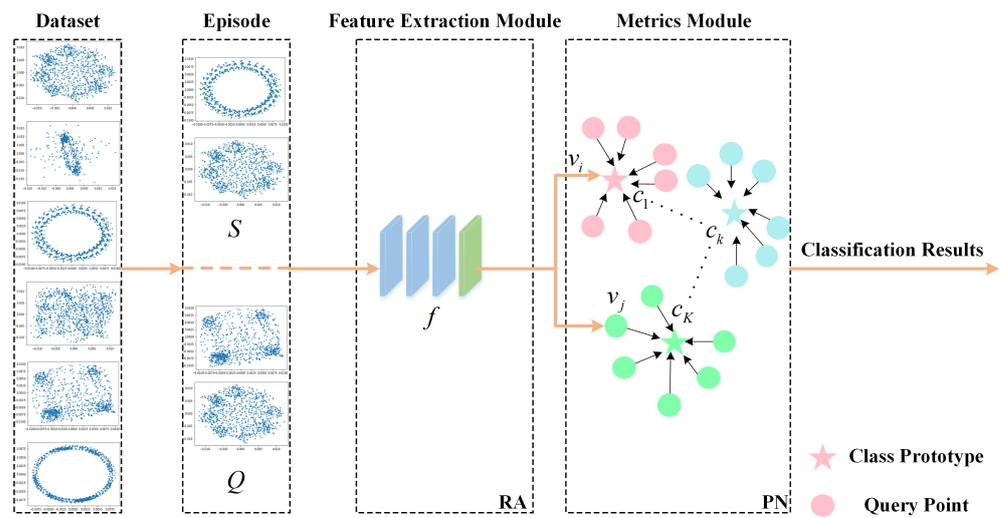


Figure 1. The architecture of PNRA.

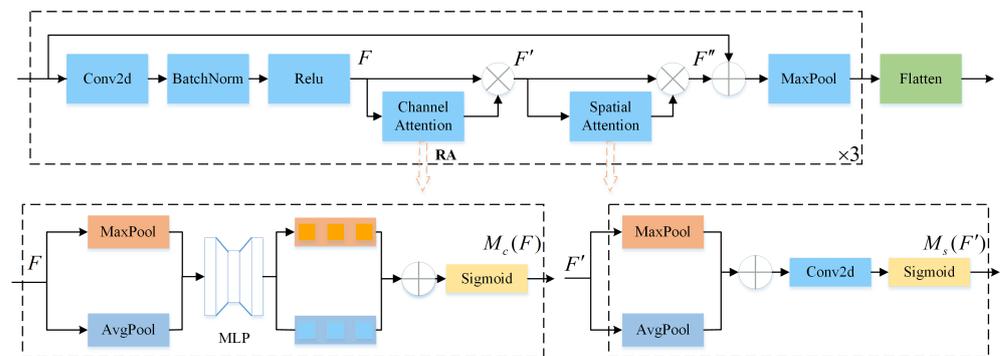


Figure 2. The architecture of RA.

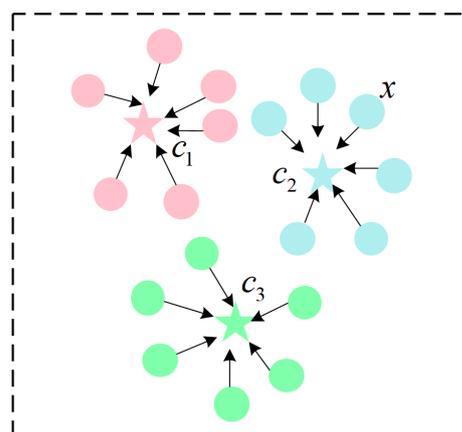


Figure 3. The architecture of PN.

3.2. RA Feature Extraction Module

To enhance the model’s feature extraction capability, we introduce a new residual attention (RA) module. This module includes channel attention to emphasize critical channel features and spatial attention to enhance useful spatial features. The attention mechanism is introduced to deep learning frameworks due to its effectiveness in guiding a model to pay more attention to critical information. In [30], the authors introduce a convolutional neural network called SCA-CNN that incorporates spatial and channel attentions in a CNN. By refining the feature maps, the SCA-CNN performs well by taking full advantage of the characteristics of CNN to yield attentive image features: spatial,

channel-wise, and multi-layer. The detailed architecture of RA is presented in Figure 2. RA is utilized to extract the salient feature vector of the constellation diagram of wireless communication signals under scarce samples.

The format of the constellation diagram for each signal is 32×32 . RA extracts the main feature vectors and reduces the dimension of the signal. Then the low-dimensional feature vectors can be easily mapped into the metric space. Structurally, the RA contains three ResBlocks and a Flatten layer. Each ResBlock includes a 2D convolution with the kernel size of 3×3 , a BatchNorm layer, an activation function Relu, a channel attention module, a spatial attention module, and a MaxPooling layer with the kernel size of 2×2 .

Specifically, the channel attention module focuses on the critical channel features using both average-pooling and max-pooling operations, generating two different sets of average-pooled features and max-pooled features, respectively. Both sets of features are then forwarded to a multi-layer perceptron (MLP). After the shared network is applied to each feature, the output feature vectors are merged using element-wise summation. The merged feature vectors are normalized using a sigmoid function to ensure that the weights for each channel fall within the range of 0 to 1. Ultimately, the normalized weights are applied to each channel of the input feature map, achieving channel-wise adaptive weighting. Given a feature map F as the input to the channel attention module, the outputs $M_c(F)$ and F' are given as:

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \quad (11)$$

$$F' = M_c(F) * F \quad (12)$$

where $*$ denotes element-wise multiplication, and σ denotes the sigmoid function.

Likewise, the spatial attention module focuses on useful spatial features using both average-pooling and max-pooling operations along the channel axis, generating two different sets of average-pooled features and max-pooled features, respectively. Both sets of features are merged using element-wise summation and then convolved by a standard convolution layer. The output feature vectors are normalized using a sigmoid function to ensure that the weights for spatial dimensions fall within the range of 0 to 1. Ultimately, the normalized weights are applied to the spatial dimensions of the input feature map, achieving spatial adaptive weighting. Given a feature map F' as the spatial attention input, the outputs $M_s(F')$ and F'' are given as:

$$M_s(F') = \sigma(f^{3 \times 3}([\text{AvgPool}(F'); \text{MaxPool}(F')])) \quad (13)$$

$$F'' = M_s(F') * F' \quad (14)$$

where $*$ denotes element-wise multiplication, σ denotes the sigmoid function, and $f^{3 \times 3}$ represents a convolution operation with the filter size of 3×3 .

By introducing the channel attention module and the spatial attention module, the network can adaptively learn the importance of each channel and its importance in different spatial locations. This enables RA to effectively leverage the relationships among channels and the diversity of features, thereby improving the performance of signal classification tasks.

3.3. PN Metrics Module

To improve the few-shot classification ability of the model, PN [29] is utilized to train an effective classifier, measuring the Euclidean distance between the class prototype and the query point. It learns a metric space in which points cluster around a single prototype representation for each signal class. Firstly, the network learns a non-linear mapping of the input signal into an embedding space and takes the mean of its support set in the embedding space to be each signal class prototype. Classification is then performed for an embedded query point by identifying the nearest class prototype. By minimizing the

loss function to update the network parameters, samples in the same signal category are brought closer, while samples from different signal categories are pushed farther apart. The detailed architecture of PN is presented in Figure 3.

According to Figure 3, x is the feature vector of the query signal point, and c_1 , c_2 , and c_3 represent different signal prototypes. By calculating the Euclidean distance between x and signal prototypes c_1 , c_2 , and c_3 , respectively, the distance between the query signal point x and signal prototype c_2 is shorter. Therefore, x is classified into the type represented by c_2 .

4. Experimental Results and Discussion

4.1. Datasets

To evaluate the performance of the proposed method, we conduct experiments on the RadioML2016.10a dataset [31]. The dataset comprises eight digital modulated signals widely used in wireless communications, including 8PSK, BPSK, CPFSK, GFSK, PAM4, 16QAM, 64QAM, and QPSK. Each sample includes in-phase and quadrature (IQ) channels. The SNR ranges from -20 dB to 18 dB with an interval of 2 dB. Signals are modulated at a rate of eight samples per symbol. In addition, random walk drifting of the carrier frequency oscillator, additive white Gaussian noise (AWGN), and Rician fading of the channel impulse response are taken into account in the process of generating signals. Furthermore, translation, dilation, and unknown scale are introduced when the signal is transmitted through harsh channels. Specifically, four out of eight modulation types are selected to constitute the training set. Then, the remaining four modulation types are utilized to test the models. The details of the experimental dataset are shown in Table 1.

Table 1. The details of the experimental dataset

Parameter	Value
Train set	BPSK, 8PSK, 16QAM, GFSK
Test set	QPSK, PAM4, 64QAM, CPFSK
The number of samples in the support set for each signal	15
The number of samples in the query set for each signal	20

All experiments are conducted on the Nvidia GeForce RTX 2080Ti GPU. In the training process, the deep learning framework is tensorflow. The training parameters are shown in Table 2. In the testing stage, to avoid the contingency caused by a single test, we use the average accuracy of 1000 test experiments as the final evaluation indicator. The calculation formula acc of a single test is defined as:

$$acc = \left(\frac{N_{true}}{N_{all}} \right) * 100\% \quad (15)$$

where N_{true} is the number of samples correctly classified, and N_{all} is the number of all samples.

Table 2. The training parameters.

Parameters	Value
Learning rate	0.001
Optimizer	Adam
Episode	500
Dropout	0.2

4.2. Performance Comparisons with Different Values of u_1

To evaluate the classification performance of the PNRA under different u_1 values, we consider four different values: 1, 5, 10, and 15, respectively. The value of u_2 is set to 20. For each modulation type, u_1 signals are randomly sampled to construct the support set S , and u_2 signals are selected from each modulation to form the query set Q . A detailed split of the datasets in the training and test stages is listed in Table 1. The experimental results are shown in Figure 4.

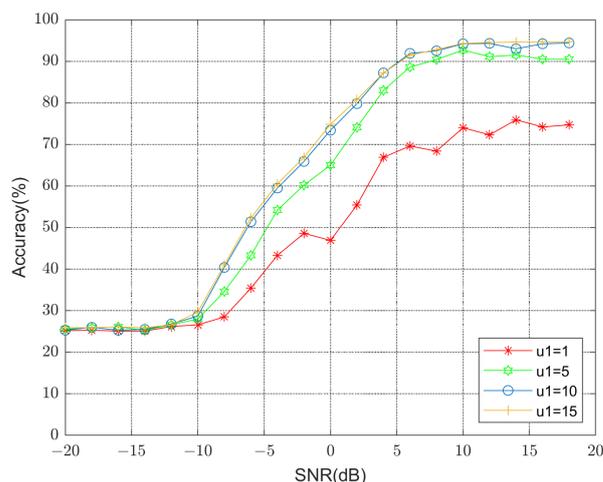


Figure 4. The number of samples required for signal prototype generation.

From Figure 4, the classification accuracy of PNRA at $u_1 = 15$ significantly outperforms that at $u_1 = 1$ and $u_1 = 5$. When $u_1 = 15$, the PNRA achieves a classification accuracy of 94.5% at 14 dB, which is an improvement of 1.5% compared to the case when $u_1 = 10$. By increasing the value of u_1 , the PNRA learns the differences between different classes from a larger sample pool, thereby enhancing its performance on new tasks. However, the rate of improvement in classification accuracy slows down, indicating diminishing returns from increasing the number of samples in the support set. In the following part, we set $u_1 = 15$ to construct the prototype for each modulation signal.

4.3. Performance Comparisons of Different Feature Extraction Methods

To evaluate the performance of the RA in this paper, we compared five different feature extraction networks. Different feature extraction networks are convolutional neural network (CNN), deep residual network (ResNet), long short-term memory (LSTM), convolutional long short-term memory fully connected deep neural networks (CLDNNs), and residual attention (RA). A detailed split of the datasets in the training and test stages is listed in Table 1. The number of signals in support set u_1 is set to 15, and the number of signals in query set u_2 is set to 20. The experimental results under different feature extraction modules are shown in Figure 5. Table 3 presents several different feature extraction module structures proposed in this paper.

Table 3. Several different feature extraction module structures.

LSTM	RA	CNN	CLDNN
			Conv2d + BN + ReLU
		Conv2d + BN + ReLU	MaxPool2D
LSTM layer	ResBlock	MaxPool2D	Conv2d + BN + ReLU
LSTM layer	ResBlock	Conv2d + BN + ReLU	MaxPool2D
Flatten	Flatten	MaxPool2D	LSTM layer
		Flatten	Flatten

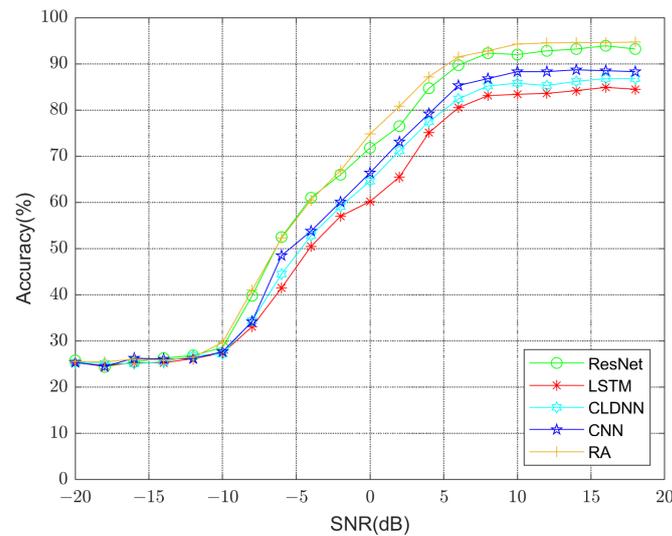


Figure 5. Performance comparisons of different feature extraction methods.

As shown in Table 3, the CNN comprises two convolutional layers and a Flatten layer. Each convolutional layer includes a 2D convolution (Conv2d) with the kernel size 3×3 , a BatchNorm (BN) is used for normalization after each convolutional layer to accelerate the network convergence speed, a rectified linear unit (ReLU) serves as the activation function, and a MaxPooling layer with the kernel size 2×2 (MaxPool2D) is employed to reduce data dimension, simplifying the network complexity and reducing the computation amount. The LSTM consists of two LSTM layers and a Flatten layer. Each LSTM layer is composed of 64 LSTM cells. Generally, the LSTM cell mainly controls the flow and loss of feature information extracted from the signal through the three gate mechanisms of forget gate, memory gate, and output gate. The CLDNN contains two convolutional layers, a LSTM layer, and a Flatten layer. The ResNet contains three ResBlocks and a Flatten layer. Each ResBlock includes a 2D convolution with the kernel size 3×3 , a BatchNorm layer, an activation function ReLU, and a MaxPooling layer with the kernel size 2×2 . The RA is illustrated in Figure 2.

According to Figure 5, the RA feature extraction module achieves superior classification accuracy due to the introduction of the attention mechanism. This allows the PNRA network to focus more on the parts that are beneficial to classification when extracting signal features.

4.4. Performance Comparisons of PNRA with Mainstream FSL Methods

To evaluate the classification performance of the PNRA under the data-hungry scenario, we compared the classification accuracy of PNRA with three FSL methods, including model-agnostic meta-learning (MAML) [32], matching network (MN) [33], and relation network (RN) [34]. Specifically, 15 samples and 20 samples are randomly selected from each modulation type to form the support set and the query set, respectively. The experimental results of various FSL methods are shown in Figure 6.

Figure 6 shows that the proposed PNRA achieves superior classification accuracy. The classification accuracy of the PNRA exceeded 80% at 2 dB SNR. In addition, when the SNR is equal to 10 dB, the PNRA outperforms MN and RN by a margin of 6.5% and 3.1%, respectively, and when the SNR is equal to -6 dB, the PNRA outperforms MN and RN by a margin of 3.7% and 4.2%, respectively. PNRA measures the similarity between samples through Euclidean distance. Euclidean distance belongs to Bregman divergence; the difference between different types of modulation signals in the metric space can be maximized, leading to better classification results. MAML seeks an optimal initialization parameter through learning multiple tasks so that the network can quickly adapt to new

types of signals. However, the MAML needs to be fine-tuned when facing new types of signals. Due to the data-hungry nature of the signal, it is difficult to apply to networks with large parameters, limiting the further improvement of MAML performance. These results indicate the effectiveness of the PNRA.

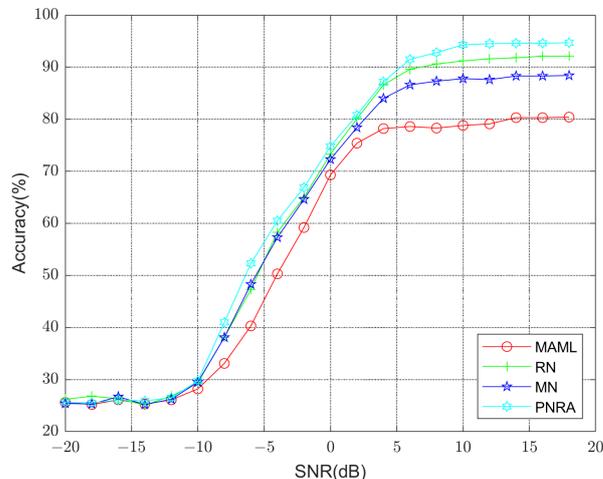


Figure 6. Experimental results of various FSL methods.

4.5. Performance Comparisons of PNRA under Different Dataset Splits

To evaluate the classification performance of the PNRA under different signal categories in the training stage and test stage, we randomly consider two dataset splits. Specifically, four out of eight modulation types are randomly selected to constitute the training set, while the remaining four modulation types are utilized for testing the models. The signal split scheme is presented in Table 4. The number of signals in support set u_1 is set to 15, and the number of signals in query set u_2 is set to 20. The experimental results under different dataset splits are shown in Figure 7. Figure 8 displays the confusion matrix for the split1 test set of the PNRA when SNR is equal to 18 dB, while Figure 9 displays the confusion matrix for the split2 test set of the PNRA under the same SNR condition.

Table 4. The split scheme of all signals

Splits	Train Set	Test Set
split1	BPSK, 8PSK, 16QAM, GFSK	QPSK, PAM4, 64QAM, CPFSK
split2	BPSK, PAM4, 16QAM, GFSK	QPSK, 8PSK, 64QAM, CPFSK

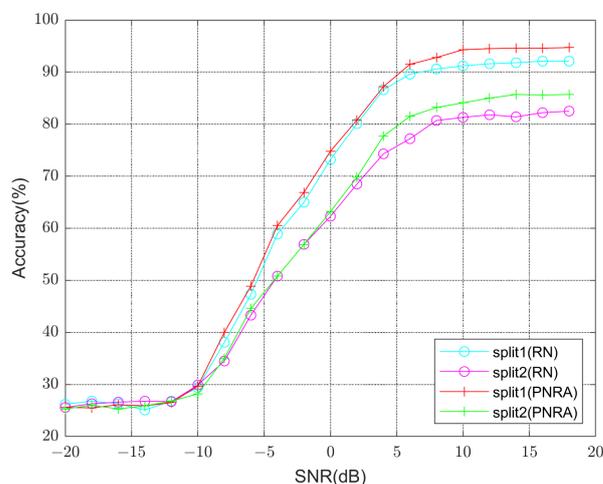


Figure 7. Experimental results under different dataset splits.

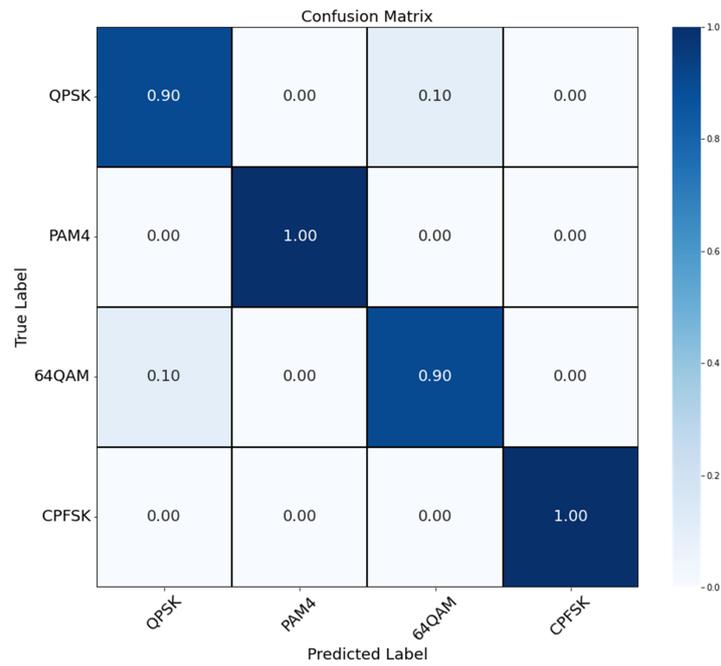


Figure 8. The confusion matrix for the split1 test set.

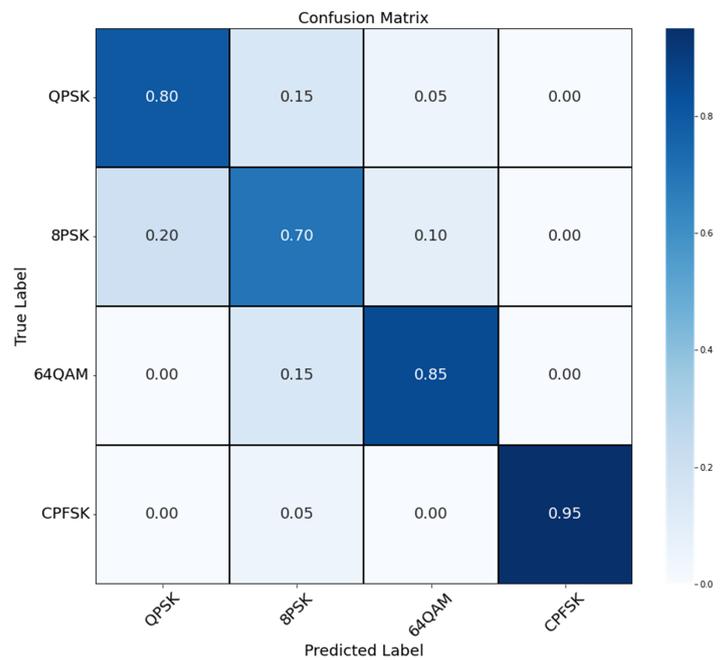


Figure 9. The confusion matrix for the split2 test set.

To intuitively observe the classification situation of query signal points, Principal Component Analysis (PCA) is employed to visualize the classification results. Figure 10 displays the classification result for the split1 test set of the PNRA when SNR is equal to 18 dB, while Figure 11 displays the classification result for the split2 test set of the PNRA under the same SNR condition.

According to Table 4, the test set of split1 is composed of four different digital modulation signals, while the test set of split2 contains two similar phase shift keying (PSK) modulation signals. As seen in Figures 8 and 10, PNRA achieves superior classification

accuracy. However, as seen in Figures 9 and 11, the classification accuracy for split2 test set drops noticeably compared to split1 due to confusion between QPSK and 8PSK signals. Experimental results show that the PNRA can adapt to the scenarios where the signal categories in the train set and test set differ. In addition, varying dataset splits result in differences in classification accuracy under the same method. Since the labels assigned to the training and testing sets were entirely separate of the PNRA and the RN methods, different signal categories in the training stage and test stage have a certain impact on the classification accuracy.

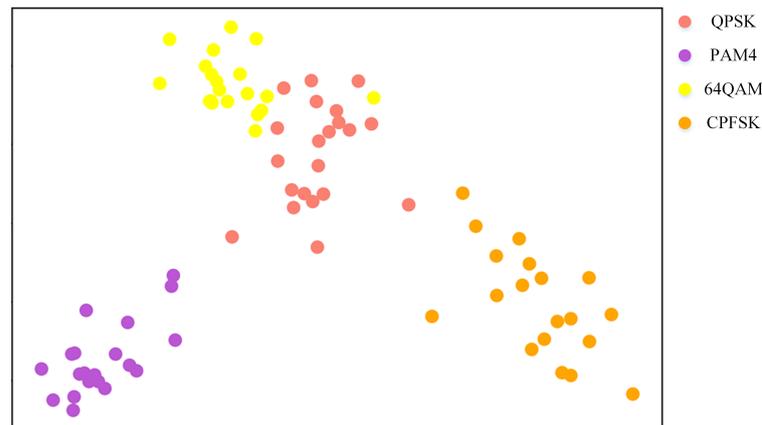


Figure 10. The PCA figure of split1.

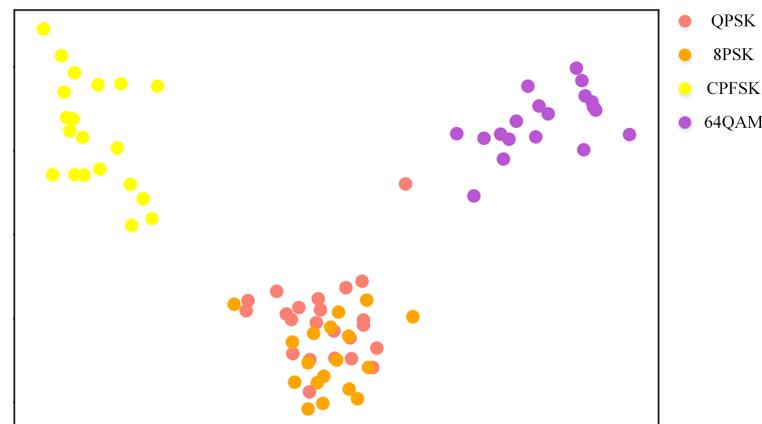


Figure 11. The PCA figure of split2.

According to Figure 7, when SNR is greater than -10 dB, the classification accuracy of split2 test set drops obviously compared to split1 for both PNRA and RN methods. Therefore, the classification accuracy is influenced by the similarity of modulation signal types in the test set. When the SNR is equal to 10 dB, PNRA outperforms RN by a margin of 3.1% for split1 and outperforms RN by a margin of 2.8% for split2. When the SNR is equal to -6 dB, PNRA outperforms RN by a margin of 1.6% for split1 and outperforms RN by a margin of 1.3% for split2. These results indicate the robustness of the PNRA.

5. Conclusions

In this paper, we propose a novel method named PNRA to achieve effective and robust AMC under the data-hungry condition. The RA component is utilized to extract the salient features between signals with different modulations, while the PN component effectively addresses the challenge of limited labeled samples in the data-hungry scenario. Experimental results demonstrate that the proposed PNRA achieves superior performance compared to mainstream FSL methods and facilitates few-shot modulation classification of new target signals. Future works will focus on improving the classification accuracy when

the test set contains multiple similar modulation signal types and utilizing a large number of unlabeled signal samples to enhance the FSL-based AMC task.

Author Contributions: Conceptualization, B.Z. and Z.Z.; methodology, B.Z., Z.Z. and X.G.; software, B.Z. and X.G.; validation, Z.Z., B.Z. and X.G.; formal analysis, L.L. and X.G.; investigation, B.Z., Z.Z. and X.G.; resources, L.L., H.Z. and X.G.; data curation, H.Z.; writing—original draft preparation, X.G. and Z.Z.; writing—review and editing, X.G. and B.Z.; visualization, X.G. and B.Z.; supervision, B.Z., X.G. and H.Z.; project administration, B.Z. and Z.Z.; funding acquisition, B.Z. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants # 62071349, # 62203343 and # U21A20455, and the Key Research and Development Program of Shaanxi (Program No. 2023-YBGY-223).

Data Availability Statement: In this paper, the RadioML2016.10a dataset is employed for experimental verification. The RadioML2016.10a dataset is a representative dataset for testing and evaluation of current AMC methods. Readers can obtain the dataset from the author by email (gxp@stu.xidian.edu.cn).

Acknowledgments: I would like to acknowledge my colleagues for their wonderful collaboration and patient support. I also thank all the reviewers and editors for their great help and useful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Huynh-The, T.; Pham, Q.V.; Nguyen, T.V.; Nguyen, T.T.; Ruby, R.; Zeng, M.; Kim, D.S. Automatic modulation classification: A deep architecture survey. *IEEE Access* **2021**, *9*, 142950–142971. [[CrossRef](#)]
2. Dobre, O.A.; Abdi, A.; Bar-Ness, Y.; Su, W. Survey of automatic modulation classification techniques: Classical approaches and new trends. *IET Commun.* **2007**, *1*, 137–156. [[CrossRef](#)]
3. Mendis, G.J.; Wei, J.; Madanayake, A. Deep learning-based automated modulation classification for cognitive radio. In Proceedings of the 2016 IEEE International Conference on Communication Systems (ICCS), Shenzhen, China, 14–16 December 2016; pp. 1–6.
4. Grajal, J.; Yeste-Ojeda, O.; Sanchez, M.A.; Garrido, M.; López-Vallejo, M. Real time FPGA implementation of an automatic modulation classifier for electronic warfare applications. In Proceedings of the 2011 19th European Signal Processing Conference, Barcelona, Spain, 29 August–2 September 2011; pp. 1514–1518.
5. Liao, K.; Zhao, Y.; Gu, J.; Zhang, Y.; Zhong, Y. Sequential convolutional recurrent neural networks for fast automatic modulation classification. *IEEE Access* **2021**, *9*, 27182–27188. [[CrossRef](#)]
6. Panagiotou, P.; Anastasopoulos, A.; Polydoros, A. Likelihood ratio tests for modulation classification. In Proceedings of the MILCOM 2000 Proceedings, 21st Century Military Communications, Architectures and Technologies for Information Superiority (Cat. No. 00CH37155), Los Angeles, CA, USA, 22–25 October 2000; pp. 670–674.
7. Jiang, X.R.; Chen, H.; Zhao, Y.D.; Wang, W.Q. Automatic modulation recognition based on mixed-type features. *Int. J. Electron.* **2021**, *108*, 105–114. [[CrossRef](#)]
8. Chang, S.; Huang, S.; Zhang, R.; Feng, Z.; Liu, L. Multitask-learning-based deep neural network for automatic modulation classification. *IEEE Internet Things J.* **2021**, *9*, 2192–2206. [[CrossRef](#)]
9. Zhang, D.; Lu, Y.; Li, Y.; Ding, W.; Zhang, B.; Xiao, J. Frequency Learning Attention Networks based on Deep Learning for Automatic Modulation Classification in Wireless Communication. *Pattern Recognit.* **2023**, *137*, 109345. [[CrossRef](#)]
10. Zeng, Y.; Zhang, M.; Han, F.; Gong, Y.; Zhang, J. Spectrum analysis and convolutional neural network for automatic modulation recognition. *IEEE Wireless Commun. Lett.* **2019**, *8*, 929–932. [[CrossRef](#)]
11. Al-Sa’ d, M.; Boashash, B.; Gabbouj, M. Design of an optimal piece-wise spline wigner-ville distribution for TFD performance evaluation and comparison. *IEEE Trans. Signal Process.* **2021**, *69*, 3963–3976. [[CrossRef](#)]
12. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. In Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN), Warsaw, Poland, 11–15 September 2005; pp. 799–804.
13. Wang, A.; Li, R. Research on digital signal recognition based on higher order cumulants. In Proceedings of the International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 12–13 January 2019; pp. 586–588.
14. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y.D. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Trans. Neural. Netw. Learn. Syst.* **2019**, *30*, 718–727. [[CrossRef](#)]
15. Li, R.; Li, L.; Yang, S.; Li, S. Robust automated VHF modulation recognition based on deep convolutional neural networks. *IEEE Commun. Lett.* **2018**, *22*, 946–949. [[CrossRef](#)]
16. Liang, Z.; Tao, M.; Wang, L.; Su, J.; Yang, X. Automatic modulation recognition based on adaptive attention mechanism and ResNeXt WSL model. *IEEE Commun. Lett.* **2021**, *25*, 2953–2957. [[CrossRef](#)]

17. Zhang, X.; Zhao, H.; Zhu, H.; Adebisi, B.; Gui, G.; Gacanin, H.; Adachi, F. NAS-AMC: Neural Architecture Search-Based Automatic Modulation Recognition for Integrated Sensing and Communication Systems. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1374–1386. [[CrossRef](#)]
18. Han, H.; Yi, Z.; Zhu, Z.; Li, L.; Gong, S.; Li, B.; Wang, M. Automatic Modulation Recognition Based on Deep-Learning Features Fusion of Signal and Constellation Diagram. *Electronics* **2023**, *12*, 552. [[CrossRef](#)]
19. Zhang, Z.; Wang, C.; Gan, C.; Sun, S.; Wang, M. Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD. *IEEE Trans. Signal Inf. Process.* **2019**, *5*, 469–478. [[CrossRef](#)]
20. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **2020**, *53*, 1–34. [[CrossRef](#)]
21. Wang, H.; Wang, B.; Li, Y. IAFNet: Few-shot learning for modulation recognition in underwater impulsive noise. *IEEE Commun. Lett.* **2022**, *26*, 1047–1051. [[CrossRef](#)]
22. Zhang, Z.; Li, Y.; Zhai, Q.; Li, Y.; Gao, M. Few-shot learning for fine-grained signal modulation recognition based on foreground segmentation. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2281–2292. [[CrossRef](#)]
23. Li, L.; Huang, J.; Cheng, Q.; Meng, H.; Han, Z. Automatic modulation recognition: A few-shot learning method based on the capsule network. *IEEE Wireless Commun. Lett.* **2021**, *10*, 474–477. [[CrossRef](#)]
24. Zhai, Q.; Li, Y.; Zhang, Z.; Li, Y.; Wang, S. Adaptive feature extraction and fine-grained modulation recognition of multi-function radar under small sample conditions. *IET Radar Sonar Navig.* **2022**, *16*, 1460–1469. [[CrossRef](#)]
25. Liu, M.; Liu, Z.; Lu, W.; Chen, Y.; Gao, X.; Zhao, N. Distributed few-shot learning for intelligent recognition of communication jamming. *IEEE J. Sel. Top. Signal. Process.* **2021**, *16*, 395–405. [[CrossRef](#)]
26. Che, J.; Wang, L.; Bai, X.; Liu, C.; Zhou, F. Spatial-Temporal Hybrid Feature Extraction Network for Few-shot Automatic Modulation Classification. *IEEE Trans. Veh. Technol.* **2022**, *71*, 13387–13392. [[CrossRef](#)]
27. Zhang, Z.; Li, Y.; Gao, M. Few-shot learning of signal modulation recognition based on attention relation network. In Proceedings of the 28th European Signal Processing Conference (EUSIPCO), Electr Network, Virtual , 18–22 January 2021; pp. 1372–1376.
28. Zhou, Q.; Zhang, R.; Mu, J.; Zhang, H.; Zhang, F.; Jing, X. Amcrn: Few-shot learning for automatic modulation classification. *IEEE Commun. Lett.* **2021**, *26*, 542–546. [[CrossRef](#)]
29. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4080–4090.
30. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.S. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5659–5667.
31. O’shea, T.J.; West, N. Radio machine learning dataset generation with GNU radio. In Proceedings of the 6th GNU Radio Conference, Boulder, CO, USA, 12–16 September 2016; pp. 1–6.
32. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 1126–1135.
33. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3637–3645.
34. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1199–1208.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.