electronics

MDPI

Article

CrossTLNet: A Multitask-Learning-Empowered Neural Network with Temporal Convolutional Network–Long Short-Term Memory for Automatic Modulation Classification

Gujiuxiang Gao [1], Xin Hu [1,*], Boyan Li [1], Weidong Wang [1] and Fadhel M. Ghannouchi [2]

[1] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; ggjx@bupt.edu.cn (G.G.); liboyan_vincent@bupt.edu.cn (B.L.); wangweidong@bupt.edu.cn (W.W.)
[2] iRadio Lab, University of Calgary, Calgary, AB T2N 1N4, Canada; fadhel.ghannouchi@ucalgary.ca
* Correspondence: huxin2016@bupt.edu.cn

Abstract: Amidst the evolving landscape of non-cooperative communication, automatic modulation classification (AMC) stands as an essential pillar, enabling adaptive and reliable signal processing. Due to the advancement of deep learning (DL) technology, neural networks have found application in AMC. However, the previous DL models face the inter-class confusion problem in high-order modulations. To address this issue, we propose a multitask-learning-empowered hybrid neural network, named CrossTLNet. Specifically, after the signal enters the model, it is first transformed into two task components: in-phase/quadrature (I/Q) form and amplitude/phase (A/P) form. For each task, we design a method that combines a temporal convolutional network (TCN) with a long short-term memory (LSTM) network to effectively capture long-term dependency features in high-order modulations. To enable interaction between these two different dimensional features, we innovatively introduce a cross-attention method, thereby further enhancing the model's ability to distinguish signal features. Moreover, we also design a simple and efficient knowledge distillation method to reduce the size of CrossTLNet, making it easier to deploy in real-time or resource-limited scenarios. The experimental results indicate that the suggested method exhibits exceptional performance in AMC on public benchmarks, especially in high-order modulations.

Keywords: automatic modulation classification; temporal convolutional network; long short-term memory network; cross-attention; multitask learning; knowledge distillation

Citation: Gao, G.; Hu, X.; Li, B.; Wang, W.; Ghannouchi, F.M. CrossTLNet: A Multitask-Learning-Empowered Neural Network with Temporal Convolutional Network–Long Short-Term Memory Network for Automatic Modulation Classification. *Electronics* **2023**, *12*, 4668. https://doi.org/10.3390/electronics12224668

Academic Editor: Christos J. Bouras

Received: 20 October 2023
Revised: 9 November 2023
Accepted: 13 November 2023
Published: 16 November 2023

1. Introduction

With the swift advancement in modern wireless communication, automatic modulation classification (AMC) has become increasingly associated with tasks such as spectrum monitoring and adaptive modulation, establishing it as a pivotal technology in non-cooperative communication scenarios [1]. Additionally, it finds extensive applications in various civilian and military fields. Due to the effects of channel multipath fading and noise, improving the accuracy in classifying modulation types is a challenging problem. Compared to traditional likelihood-based and feature-based methods, deep learning (DL)-based techniques have demonstrated significant performance improvements in AMC tasks, garnering widespread attention and favor from researchers.

In prior work, researchers have achieved benchmark performance in AMC through the use of a convolutional neural network (CNN) [2], a residual network (ResNet) [3], a long short-term memory (LSTM) network [4], and a hybrid network [5], as shown in Figure 1. In order to enhance classification accuracy, Zhang et al. [6] combine data in both in-phase/quadrature (I/Q) form and amplitude/phase (A/P) form, proposing the DS-CLDNN model. However, this model still confuses QAM16 and QAM64. To address this issue, Wang et al. [7] suggest using dedicated classifiers to differentiate between QAM16 and QAM64, but this leads to a high degree of customization and a lack of generality.

Chang et al. [8] propose using four different classification heads to discriminate signals under different conditions, presenting the MLDNN model. Although this improves classification accuracy, it is necessary to simultaneously optimize loss functions for four different modules during training, making the model difficult to train and apply. Ying et al. [9] propose a model called CTDNN, which combines CNN and transformer structures. However, this architecture requires a larger training dataset to effectively train the transformer component. Table 1 presents a comparison of different deep-learning-based AMC models.

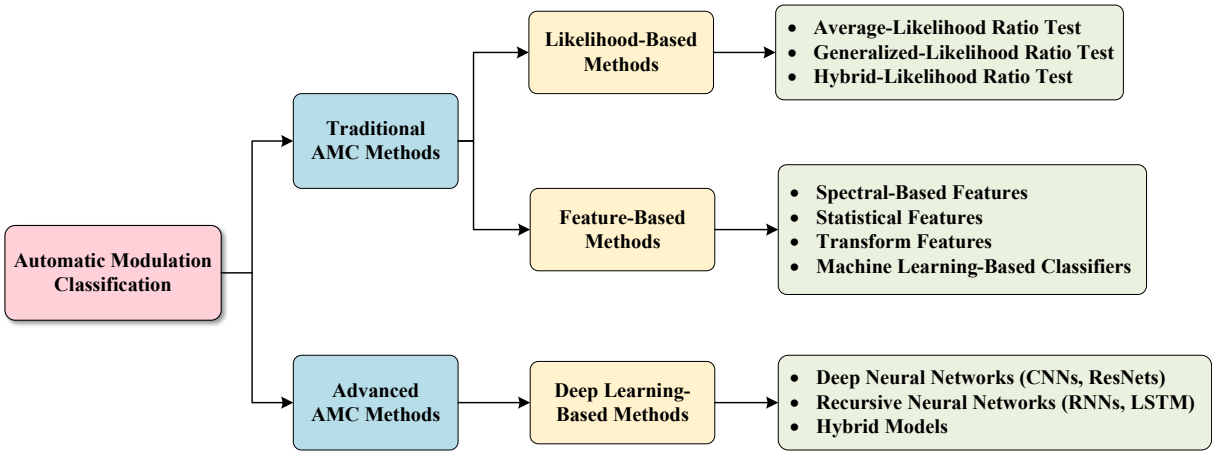

Figure 1. Summary of AMC methods.

Table 1. Comparison of different deep-learning-based AMC models.

Author	Year	Model	Dataset	Input Signal	Main Structure of the Model
O'Shea et al. [2]	2016	CNN	RML	I/Q	CNN
Liu et al. [3]	2017	ResNet	RML	I/Q	ResNet
Rajendran et al. [4]	2018	LSTM	RML	A/P	Two LSTM Layers
Zhang et al. [5]	2021	PET-CGDNN	RML	I/Q	CNN + GRU + DNN
Zhang et al. [6]	2020	DS-CLDNN	RML	I/Q + A/P	CNN + LSTM
Wang et al. [7]	2022	IQCLNet	RML	I/Q	CNN + LSTM + Expert Feature Method for QAMs
Chang et al. [8]	2021	MLDNN	RML	I/Q + A/P	CNN + BiGRU + SAFN
Ying et al. [9]	2023	CTDNN	RML	I/Q	CNN + Transformer

To tackle the aforementioned issues, a novel hybrid model called CrossTLNet is proposed in this paper, which combines temporal convolutional network (TCN) and LSTM. The signal is initially transformed into both I/Q and A/P forms by a pre-processing block. Then, within a multi-task learning framework, features of the signal are extracted separately from different dimensions through the TCN and LSTM modules. This architecture enables the effective capture of long-term dependency features in the signal, greatly improving the classification accuracy of high-order modulations. To enable the interaction of features from both the I/Q and A/P dimensions, a cross-attention method is innovatively introduced, thereby improving the feature discriminability of the signal. In addition, considering applications in real-time or resource-limited scenarios, a simple and efficient knowledge distillation method is also designed to ensure the lightweight nature of the proposed CrossTLNet. The experimental results on public benchmarks indicate that the suggested method attains accurate modulation classification, especially in high-order modulations.

The structure of this paper is as follows. Section 2 describes the signal model, presents the framework of the proposed CrossTLNet, and elaborates on the implementation details of each module. In addition, a knowledge distillation method is designed to reduce the size of CrossTLNet. Section 3 first describes the dataset and the parameter settings of the model, and then evaluates the performance of the proposed method through comparative experiments and discusses the advantages of the proposed method. Finally, Section 4 provides the conclusion of the paper.

2. Signal Model and Proposed Classification Method

2.1. Signal Model

The modulated signal $s(t)$ from the transmitter propagates through the channel $h(t)$, where it is subjected to additive white Gaussian noise $n(t)$, before reaching the receiver. The received signal $r(t)$ can be denoted as follows:

$$r(t) = s(t) * h(t) + n(t), \tag{1}$$

Subsequently, the analog received signal $r(t)$ is discretized into a digital form through the ADC. It is then decomposed into I/Q components by the DSP module, serving as the input to the subsequent neural network model for achieving AMC.

2.2. The Framework of CrossTLNet

Noticing that the signal, in addition to the commonly used I/Q form, also exhibits distinct features in the amplitude and phase dimensions, i.e., A/P form. Inspired by this, a multi-task learning framework model is proposed, as shown in Figure 2, named CrossTLNet, which achieves complementary features of I/Q and A/P form.

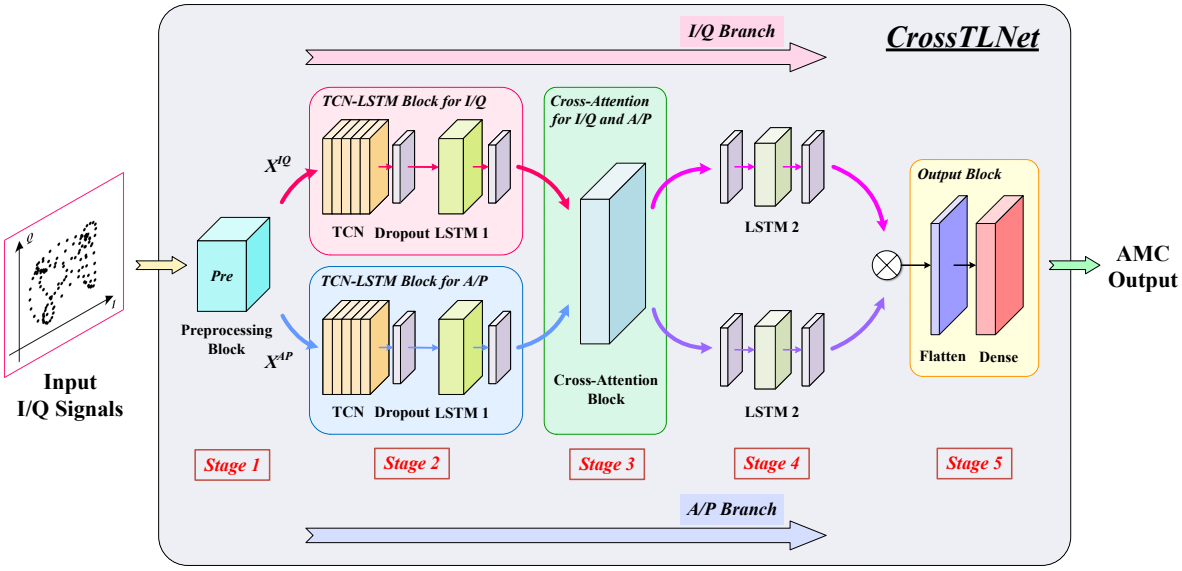

Figure 2. Model architecture of proposed CrossTLNet.

First, CrossTLNet takes the I/Q signal as its input. After passing through the pre-processing block, the signal is transformed into both I/Q form itself and A/P form, which are separately learned as two tasks. For each task, a feature extraction method with TCN–LSTM is designed to handle long-term dependency features among high-order modulations. Subsequently, a cross-attention method that enables feature interaction between the two tasks is introduced. The interacted signals are then further abstracted through a layer of LSTM, fused into one branch via an outer product operation, and finally classified through a dense layer for AMC.

Specifically, the proposed model consists of five stages.

Stage 1 : The pre-processing block takes 128×2 I/Q signal as model's input and transforms it into two parts: the original I/Q signal and the A/P signal for further processing. The transformation to obtain the A/P signal can be denoted as follows:

$$\boldsymbol{X}^{AP} = \begin{bmatrix} \boldsymbol{X}^{A} \\ \boldsymbol{X}^{P} \end{bmatrix} = \begin{bmatrix} \sqrt{\left(\mathrm{Re}(\boldsymbol{X}^{IQ})\right)^2 + \left(\mathrm{Im}(\boldsymbol{X}^{IQ})\right)^2} \\ \arctan \dfrac{\mathrm{Im}(\boldsymbol{X}^{IQ})}{\mathrm{Re}(\boldsymbol{X}^{IQ})} \end{bmatrix}, \tag{2}$$

Stage 2: A symmetric multi-task learning architecture with two TCN–LSTM blocks is designed for the initial extraction of signal features. Each TCN–LSTM block consists of a TCN module and an LSTM module, both with 256 units. The convolutional kernel size is 3, and a dropout layer (dropout rate is 0.5) is inserted after them to prevent overfitting. By combining the TCN module and LSTM module, CrossTLNet becomes more sensitive to features at different time scales, which is beneficial for distinguishing between highly similar high-order modulations. The signal processed by the TCN–LSTM block is then propagated backward in a size of 128×256.

Stage 3: The cross-attention block takes the extracted I/Q and A/P signal features from the TCN–LSTM blocks. Through a process of interaction, one task branch of the model could learn features in the other form of the signal, enhancing its focus on important features. The signal after the cross-attention block maintains the same dimensions and continues to propagate backward.

Stage 4: The signal after passing through the cross-attention block undergoes further feature extraction through a new LSTM layer with 128 units. To prevent overfitting, dropout layers (dropout rate is 0.5) are added before and after the LSTM layer. The output is a one-dimensional feature sequence of 128 bits.

Stage 5: The I/Q and A/P signals after *Stage 4* are fused through an outer product operation and finally fed into the output block. Mathematically, this can be represented as:

$$\boldsymbol{Y} = f(\boldsymbol{X}^{IQ}) \cdot \left[f(\boldsymbol{X}^{AP}) \right]^{T}, \tag{3}$$

where $f(\cdot)$ represents all operations from *Stage 1* to *Stage 4*, and $\boldsymbol{Y}$ represents the input to the flatten layer. The size of $\boldsymbol{Y}$ is 128×128. After the outer product operation, the data features from the two branches are adequately amplified and fused. Following the flatten layer, the final classification is performed by the dense layer, resulting in the output of the results.

The following will provide specific details about the method with TCN–LSTM for high-order modulations and the method of cross-attention.

2.3. The Method with TCN–LSTM for High-Order Modulations

Previous studies have encountered the issue of classification confusion when dealing with high-order modulations. Compared with conventional DL methods such as CNN, a method with TCN–LSTM is proposed for different high-order modulations, which achieves accurate classification by improving the model's capability to capture long-term dependency features. Without loss of generality, two different high-order modulations of the same scheme *HMOD* are denoted as *HMOD-A* and *HMOD-B*, respectively, where *HMOD-B* is of higher order than *HMOD-A*, i.e., $B > A$. In this way, *HMOD-A* and *HMOD-B* are used as examples to elucidate the motivation and the design details of the suggested method.

An analysis is conducted first to understand the reasons behind the classification confusion between *HMOD-A* and *HMOD-B*. In an ideal scenario, *HMOD-A* and *HMOD-B* utilize *A* and *B* distinct symbols, respectively, for information representation. They possess noticeable differences, and there are clear distinctions in their amplitude and phase representations as well. However, since both of them are based on the same *HMOD* constellation distribution, if a DL model only focuses on capturing and learning local features, it may find the two very similar and thus easily confuse them.

Drawing from the above analysis, it becomes evident that for a DL model to accurately differentiate between *HMOD-A* and *HMOD-B*, it must possess the capability to focus on capturing global features. Compared to CNN, TCN excels at capturing global features, enabling a more comprehensive distinction between the features of *HMOD-A* and *HMOD-B*. Inspired by [10], the TCN module in TCN–LSTM block is designed as illustrated in Figure 3. The TCN module consists of five residual blocks. Drawing inspiration from DenseNet, skip connections are also employed between each residual block to enhance the gradient flow. Within each residual block, one-dimensional causal convolutions are utilized. Unlike

traditional convolutions, causal convolutions cannot see future data. In other words, the output at time t is contingent solely upon the input at or before time t in the preceding layers, imposing a strict temporal constraint. This enables TCN to effectively handle time series data. Moreover, dilated convolutions with dilation factors of 1, 2, 4, 8, 16, and 32 are also applied in the one-dimensional causal convolution. This allows TCN to exponentially increase its receptive field without incurring pooling-related information loss, enabling each convolutional output to encompass a larger range of information. This equips TCN with robust long-term features processing capabilities and the capacity to capture global features. Following the one-dimensional convolution, layer normalization is employed to enhance the stability of model training. Furthermore, for other high-order modulations, effective classification can be achieved by adjusting the number of residual blocks in the TCN module and units in the one-dimensional convolution.

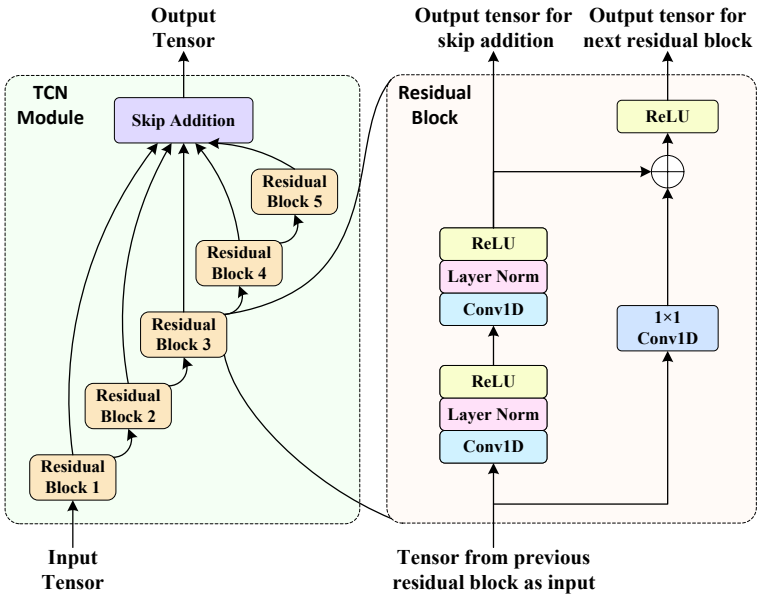

Figure 3. The detailed structure of TCN module.

While LSTM can also address the issue of long-term dependencies in time series data, TCN has an advantage over LSTM in processing sequences due to its parallel computation capability. TCN can process the entire sequence simultaneously, whereas LSTM needs to handle each time step sequentially. This means that TCN can more effectively capture spectral features in the signal without being limited by the sequence length. In comparison, *HMOD-B* signals are more complex and have more spectral features compared to *HMOD-A*. Therefore, better parallelism is needed to capture these features. Additionally, considering that LSTM uses gating mechanisms to control information flow for handling long-term dependencies, improper settings of these gating mechanisms can lead to issues like information flow obstruction or gradient vanishing, potentially causing the forgetting of important information. TCN, on the other hand, does not have gating mechanisms, allowing it to more stably and effectively capture long-term features between *HMOD-A* and *HMOD-B*.

In CrossTLNet, inspired by the hybrid structure of CNN–LSTM, we combine the designed TCN module with the LSTM module to form TCN–LSTM block. This combination not only mitigates the gradient vanishing problem in LSTM but also enhances the model's ability to extract and represent signal features. In fact, this structure achieves better performance than using TCN alone. In the ablation studies of Section 3, the impact brought by the TCN module will be further validated.

2.4. The Method of Cross-Attention

I/Q and A/P signals possess distinct dimensional features. Inspired by multi-modal data fusion, a cross-attention method is innovatively introduced, as illustrated in Figure 4, to enable interaction and emphasize the unique features from both I/Q and A/P.

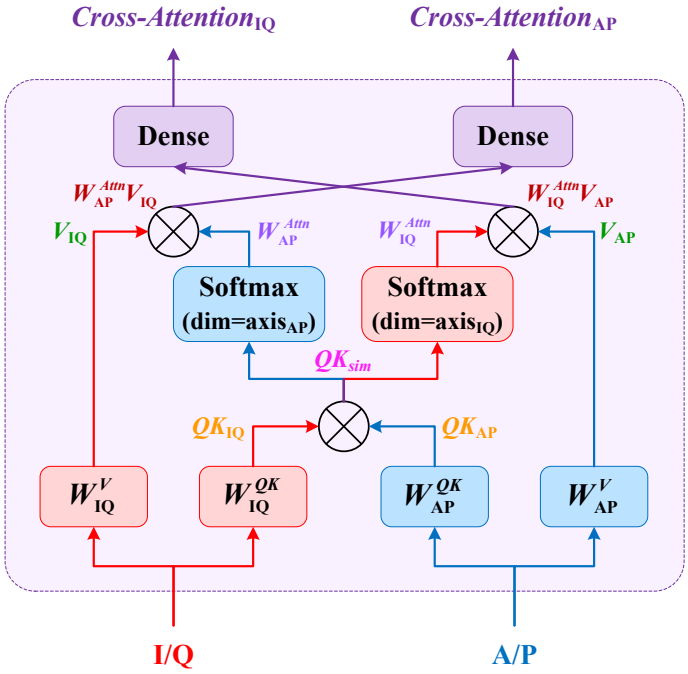

Figure 4. The detailed structure of cross-attention.

Unlike the conventional attention mechanism, the query matrix $\boldsymbol{Q}$, the key matrix $\boldsymbol{K}$, and the value matrix $\boldsymbol{V}$ for each task branch are separately calculated, and $\boldsymbol{Q}$ and $\boldsymbol{K}$ are shared, denoted as the matrix $\boldsymbol{QK}$. Next, the similarity matrix $\boldsymbol{QK}_{sim}$ for matrix $\boldsymbol{QK}_{\mathrm{IQ}}$ of I/Q task branch and $\boldsymbol{QK}_{\mathrm{AP}}$ of A/P task branch is calculated, and the Softmax function is utilized to calculate attention weights $\boldsymbol{W}_{\mathrm{AP}}^{Attn}$ along the A/P dimension and $\boldsymbol{W}_{\mathrm{IQ}}^{Attn}$ along the I/Q dimension of $\boldsymbol{QK}_{sim}$, respectively. Subsequently, the pairs $(\boldsymbol{W}_{\mathrm{IQ}}^{Attn}, \boldsymbol{W}_{\mathrm{AP}}^{Attn})$ and $(\boldsymbol{V}_{\mathrm{IQ}}, \boldsymbol{V}_{\mathrm{AP}})$ are cross-multiplied, and then passed through a dense layer, respectively. Finally, the cross-attention for the I/Q task branch is obtained, denoted as $\boldsymbol{Cross\text{-}Attention}_{\mathrm{IQ}}$, and for the A/P task branch, denoted as $\boldsymbol{Cross\text{-}Attention}_{\mathrm{AP}}$.

Mathematically, it can be expressed as:

$$\boldsymbol{QK}_{sim} = \mathrm{DotProduct}(\boldsymbol{QK}_{\mathrm{IQ}}, \boldsymbol{QK}_{\mathrm{AP}}) = \boldsymbol{QK}_{\mathrm{IQ}} \cdot \boldsymbol{QK}_{\mathrm{AP}}^{T}, \tag{4}$$

$$\begin{cases} \boldsymbol{Cross\text{-}Attention}_{\mathrm{IQ}} = Dense\left[\mathrm{Softmax}\left(\dfrac{\boldsymbol{QK}_{sim}}{\sqrt{d_{QK}}}, \mathrm{dim} = \mathrm{axis}_{\mathrm{IQ}}\right)\boldsymbol{V}_{\mathrm{AP}}\right] \\ \boldsymbol{Cross\text{-}Attention}_{\mathrm{AP}} = Dense\left[\mathrm{Softmax}\left(\dfrac{\boldsymbol{QK}_{sim}}{\sqrt{d_{QK}}}, \mathrm{dim} = \mathrm{axis}_{\mathrm{AP}}\right)\boldsymbol{V}_{\mathrm{IQ}}\right] \end{cases}, \tag{5}$$

where d_{QK} represents the dimension of the matrix $\boldsymbol{QK}$, playing a role in scaling the dot product to mitigate the vanishing gradient issue associated with the Softmax function [11]. $Dense[\cdot]$ represents the operation of passing through a dense layer. The dense layer serves to align dimensions and facilitate further processing. The interaction of I/Q and A/P data through cross-attention incorporates features from the other dimension, which has a positive impact on the final classification accuracy. The results of the ablation studies in Section 3 also demonstrate this.

2.5. The Method of Model Lightweighting

In resource-limited scenarios, deploying smaller models is necessary, which is often overlooked by many researchers. Therefore, a simple and effective model lightweighting method is designed with knowledge distillation [12]. In essence, knowledge distillation involves transferring the knowledge learned by a teacher model (which is large) through the training process to a student model (which is small), enabling the smaller model to possess the generalization capabilities of the larger one. The lightweighting method combines feature-based knowledge distillation and logits-based knowledge distillation. This design greatly facilitates the student model in learning different layers' responses of the teacher model, enhancing distillation accuracy, while achieving a higher model compression rate.

- The part of feature-based knowledge distillation.

For both the student and teacher models, if the student model can learn not only the output results of the teacher model but also its way of thinking (that is, the responses of the intermediate layers), it will be more advantageous for enhancing the effectiveness of knowledge distillation. The intermediate feature responses of an excellent student model must be similar to those of the teacher model. Inspired by [13], we output the feature maps after the TCN module in CrossTLNet (both I/Q and A/P branches) and minimize their difference between the student and teacher during the distillation process, as shown in Figure 5.

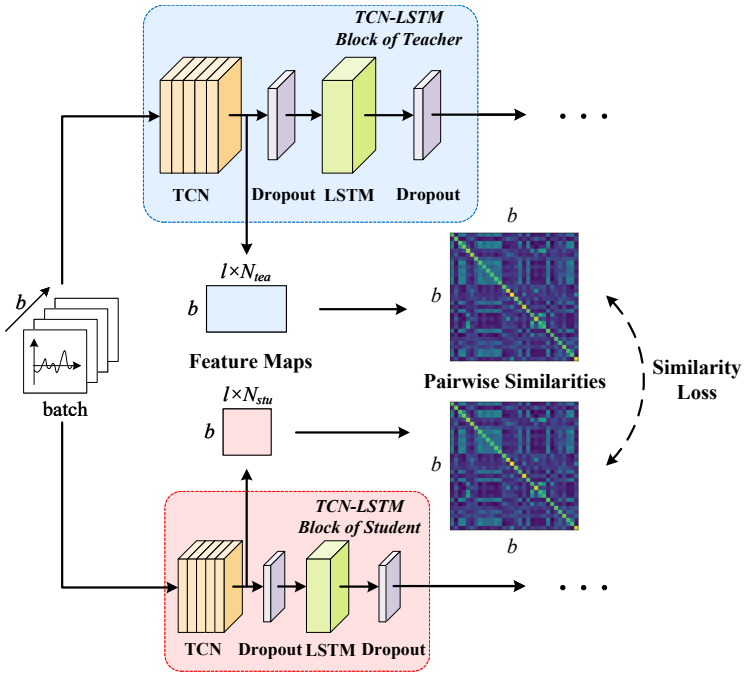

Figure 5. Feature-based knowledge distillation part of CrossTLNet. It is worth noting that the TCN modules in both the I/Q and A/P branches calculate the similarity loss in the same way.

For a batch of input with a batch-size of b, after passing through the TCN module, the teacher and student models obtain feature maps of size $b \times (l \times N_{tea})$ and $b \times (l \times N_{stu})$, respectively, denoted as $\mathcal{F}_{tea}$ and $\mathcal{F}_{stu}$. Here, l represents the sampling length of the input signal, and N_{tea} and N_{stu} represent the number of units in the TCN module for the teacher and student models, respectively. Subsequently, in order to align the dimensions of the feature maps, the feature maps are multiplied with their transpose to obtain feature maps of size $b \times b$ for both the teacher and student models. To reduce the numerical range and stabilize the model training, L2 normalization is applied to the transformed feature maps, and the results are denoted as $\mathcal{G}_{tea}$ and $\mathcal{G}_{stu}$ for the teacher and student models, respectively.

Therefore, the objective of feature-based knowledge distillation is to minimize the difference between the feature maps, which is denoted as the similarity loss $\mathcal{L}_{\text{sim}}$. Mathematically,

$$\mathcal{G}_{tea} = \frac{\mathcal{F}_{tea} \cdot \mathcal{F}_{tea}^T}{\|\mathcal{F}_{tea} \cdot \mathcal{F}_{tea}^T\|_2}; \quad \mathcal{G}_{stu} = \frac{\mathcal{F}_{stu} \cdot \mathcal{F}_{stu}^T}{\|\mathcal{F}_{stu} \cdot \mathcal{F}_{stu}^T\|_2}, \tag{6}$$

where $\|\cdot\|_2$ represents the L2 norm. Therefore, the similarity loss $\mathcal{L}_{\text{sim}}$ can be defined as:

$$\mathcal{L}_{\text{sim}} = \left.\frac{\|\mathcal{G}_{tea} - \mathcal{G}_{stu}\|_F^2}{b^2}\right|_{\text{IQ}} + \left.\frac{\|\mathcal{G}_{tea} - \mathcal{G}_{stu}\|_F^2}{b^2}\right|_{\text{AP}}, \tag{7}$$

where $\|\cdot\|_F^2$ represents the Frobenius norm. The two terms on the right-hand side of Equation (7) represent the similarity losses after the TCN modules for the I/Q branch and the A/P branch, respectively.

- The part of logits-based knowledge distillation

The reference [14] points out that the vanilla knowledge distillation with KL divergence requires an exact match of logits outputs between the teacher and student models, which is too strict. In fact, it is only necessary to ensure that the relative ranks of logits between the student and teacher are consistent, and the specific numerical values are not of concern. Inspired by this, compared to the classic KL divergence, we prefer to use the Pearson's distance d_{P} to measure the logits outputs of the teacher and student models. Denote the logits outputs of the student and teacher models as $\mathcal{Z}_{stu}$ and $\mathcal{Z}_{tea}$, respectively, then:

$$\mathcal{Y}_{stu} = \text{Softmax}(\frac{\mathcal{Z}_{stu}}{T}, \text{dim} = 1); \quad \mathcal{Y}_{tea} = \text{Softmax}(\frac{\mathcal{Z}_{tea}}{T}, \text{dim} = 1), \tag{8}$$

$$d_{\text{P}}(\mathcal{Y}_{stu}, \mathcal{Y}_{tea}) = 1 - \rho_{\text{P}}(\mathcal{Y}_{stu}, \mathcal{Y}_{tea}) = 1 - \frac{\text{Cov}(\mathcal{Y}_{stu}, \mathcal{Y}_{tea})}{\text{Std}(\mathcal{Y}_{stu})\text{Std}(\mathcal{Y}_{tea})}, \tag{9}$$

where T is the temperature coefficient, higher temperature value results in a softer probability distribution across different classes. $\mathcal{Y}_{stu}$ and $\mathcal{Y}_{tea}$ represent the softened probability distributions after temperature scaling. "dim = 1" indicates that the Softmax is computed along the rows. ρ_{P} represents the Pearson correlation coefficient, Cov represents covariance, and Std represents standard deviation, respectively.

For a batch of input with a batch-size of b and c classes, the model's logits output is a $b \times c$ matrix. Each row reflects the inter-class relation between different classes, while each column reflects the intra-class relation of the same class in a batch. In the case of logits-based knowledge distillation, the objective is to ensure that the teacher and student models exhibit comparable relative ranks in the inter-class and intra-class relation in the logits matrix, as illustrated in Figure 6.

Mathematically, we can represent the logits loss $\mathcal{L}_{\text{logits}}$ as:

$$\begin{cases} \mathcal{L}_{\text{inter}} = \frac{1}{b}\sum_{i=1}^{b} d_{\text{P}}(\mathcal{Y}_{stu|i,:}, \mathcal{Y}_{tea|i,:}); \quad \mathcal{L}_{\text{intra}} = \frac{1}{c}\sum_{j=1}^{c} d_{\text{P}}(\mathcal{Y}_{stu|:,j}, \mathcal{Y}_{tea|:,j}) \\ \mathcal{L}_{\text{logits}} = \mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}} \end{cases}, \tag{10}$$

where $\mathcal{L}_{\text{inter}}$ represents the inter-class loss, $\mathcal{L}_{\text{intra}}$ represents the intra-class loss, and the notations "$i,:$" and "$:,j$" represent row-wise and column-wise calculations, respectively.

In summary, the overall training loss $\mathcal{L}$ can be composed of three parts: the original classification loss $\mathcal{L}_{\text{cls}}$, the feature-based loss $\mathcal{L}_{\text{sim}}$, and the logits-based loss $\mathcal{L}_{\text{logits}}$, which can be defined as:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{cls}} + \beta\mathcal{L}_{\text{sim}} + \gamma\mathcal{L}_{\text{logits}}, \tag{11}$$

where α, β, and γ are weighting factors. In this way, we can utilize the total loss $\mathcal{L}$ for knowledge distillation to lightweight the proposed CrossTLNet. The principle of Occam's

Razor suggests that the simplest explanation is often the best. Therefore, compared to many sophisticated knowledge distillation methods, our method is quite straightforward and simple. This means it is easier to implement in practice, which is beneficial for the actual application and deployment of CrossTLNet.

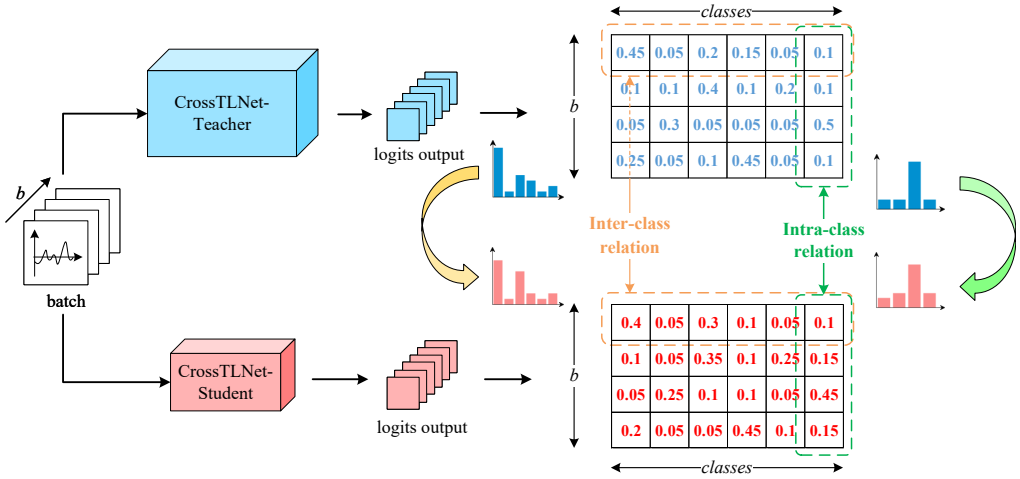

Figure 6. Logits-based knowledge distillation part of CrossTLNet.

3. Experiments and Results Analysis

3.1. Dataset and Training Setting

The public RML2016.10A dataset [2,15] is utilized for training and evaluating the model. This dataset serves as a popular benchmark for AMC and has been widely adopted by researchers. It comprises 220,000 samples of modulated signals, each represented as a 128×2 vector in I/Q form. The dataset encompasses 11 common modulation schemes, including 8 digital modulations and 3 analog modulations. Among them, QAM16 and QAM64 are two examples of high-order modulations included in the dataset. Additionally, the dataset simulates a range of real-world channel impairments and includes varying levels of noise, and the Signal-to-Noise Ratio (SNR) varies over the range of −20–18 dB, increasing in 2 dB intervals. Each SNR includes 11,000 samples of modulated signals. The detailed channel model parameters of the RML2016.10A dataset are shown in Table 2.

Table 2. RML2016.10A channel model parameters.

Parameter Name	Value
Sampling frequency	200 KHz
Sampling rate offset standard deviation	0.01 Hz
Maximum sampling rate offset	50 Hz
Carrier frequency offset standard deviation	0.01 Hz
Maximum carrier frequency offset	500 Hz
Number of sinusoids used in frequency selective fading	8
Maximum doppler frequency used in fading	1
Fading model	Rician
Rician K-factor	4
Fractional sample delays for the power delay profile	[0.0, 0.9, 1.7]
Magnitudes corresponding to each delay time	[1, 0.8, 0.3]
Filter length to interpolate the power delay profile	8
Standard deviation of the AWGN process	$10^{-\frac{SNR}{10}}$

Our work is implemented using TensorFlow 2.13.0 and all experimental procedures are performed on an Intel Core i9-10900X@3.70GHz CPU with a GeForce RTX 2080 Ti GPU from NVIDIA company. In the event that the validation loss exhibits no improvement over a span of 5 epochs, a 50% reduction in the learning rate is implemented. To prevent

overfitting, an early stopping strategy is applied to terminate the training process early if there is no improvement in the validation loss over a span of 25 epochs. Table 3 lays out the detailed settings of hyperparameters for our proposed CrossTLNet.

Table 3. Hyperparameter settings of CrossTLNet.

Hyperparameter Name	Value
Optimizer	Adam
Initial learning rate	0.0001
Batch size	32
Early-stop patience	25
Training–validation–testing ratio of dataset	0.6:0.2:0.2

As for the knowledge distillation process, the teacher model is the CrossTLNet trained before. The student model has 64 units in the TCN–LSTM block in *Stage 2* and 32 units in the LSTM layer in *Stage 4*. The weighting factors α, β, and γ in Equation (11) are set to 1, 9×10^5, and 1, respectively. This is because the similarity loss $\mathcal{L}_{\text{sim}}$ applies L2 normalization, resulting in very small values that require a large weight to balance. The temperature coefficient T is set to 1.5. The patience for learning rate decay increases from 5 to 10, and the early stopping patience increases from 25 to 50. All other configurations remain consistent with the previous settings.

3.2. Results and Discussion

There are six representative models selected as baseline for comparison with the proposed CrossTLNet, as described in [6,8,9,16]. Their *SNR-ACC* curves are illustrated in Figure 7.

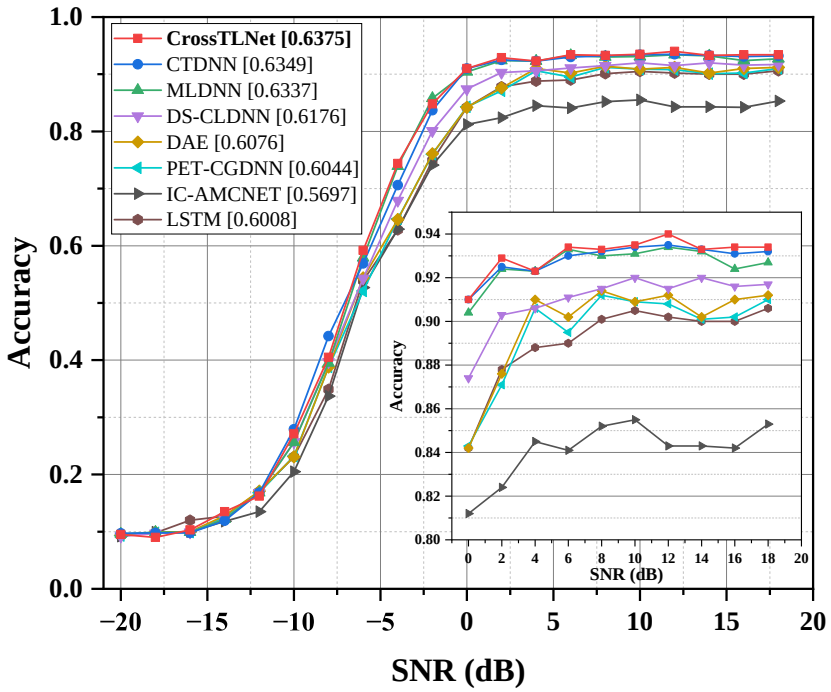

Figure 7. Comparison of CrossTLNet and other baseline models. The inner figure depicts a magnified view for $SNR \geq 0$ dB.

It is evident from the results that CrossTLNet attains an overall classification accuracy of 63.75%, surpassing all baseline models. When $SNR = 0$ dB, CrossTLNet already achieves an accuracy of 91%, making it the earliest model among all to reach 90% accuracy. Additionally, at $SNR \geq 0$ dB, the accuracy of CrossTLNet also surpasses all baseline

models. Even at low SNR, CrossTLNet exhibits impressive performance, indicating strong robustness. Furthermore, CrossTLNet is the only model to break through 94% accuracy ($SNR = 12$ dB). It is worth noting that, although CTDNN and MLDNN show similar performance compared to the proposed CrossTLNet, CTDNN utilizes a transformer structure, which implies no assumptions about the structural bias towards the input data, thus requiring a larger training set. As mentioned in [9], CTDNN utilizes 80% dataset for training, while CrossTLNet only uses 60%. As for MLDNN, it uses four different classification heads to handle signals under different scenarios. This requires the model to simultaneously optimize four loss functions during training, making the model harder to train and apply in practice. In comparison, CrossTLNet only has single input and single output, making it simpler for both training and deployment.

For a more detailed analysis of the classification performance for each modulation type, confusion matrices are generated for CrossTLNet and some selected baseline models at $SNR = 4$ dB, as shown in Figure 8. It can be observed that CrossTLNet accurately distinguishes the majority of modulation types. Particularly noteworthy is that, compared to other models, CrossTLNet is able to precisely classify the two typical high-order modulations, QAM16 and QAM64, while other models exhibit confusion between them. Specifically, CrossTLNet achieves a classification accuracy of 97% for QAM16 and QAM64, while DS-CLDNN, PET-CGDNN, IC-AMCNet, and LSTM only achieve accuracies of only 77.5%, 85%, 53%, and 85%, respectively. This represents an improvement of over 12%. When $SNR \geq 0$ dB, CrossTLNet also achieved a classification accuracy of 96.55% for QAM16 and QAM64. This phenomenon arises from the fact that QAM16 can be viewed as a sparse subset of QAM64, resulting in a substantial overlap in their constellation distributions. Furthermore, they exhibit similar time and frequency domain features in the I/Q data, leading to a high inter-class similarity. When DL models employ conventional CNN for feature extraction, the convolutional layers focus on local features rather than global features. Since both QAM16 and QAM64 are based on the same QAM constellation distribution, their local features are highly similar. Consequently, they can be easily confused. In contrast, the proposed CrossTLNet possesses a stronger processing capability for sequences with long-term dependency features, allowing it to accurately distinguish between them. Additionally, all models exhibit confusion between WBFM and AM-DSB. This can be attributed to the limited time for observation and the slow information transfer rate of these two modulations. The sequence length of 128 bits is insufficient to effectively differentiate the modulation signal features between them [17].

CrossTLNet employs a multi-task learning architecture based on I/Q and A/P. For the investigation of the gains brought by this architecture, CrossTLNet trained solely with I/Q or A/P are compared, denoted as TCN-LSTM-IQ and TCN-LSTM-AP, respectively. Figure 9 illustrates the comparison results between them. It can be observed that training with I/Q leads to a higher classification accuracy compared to training with A/P, and combining both in CrossTLNet achieves better performance. This is because using I/Q signals allows for a more direct capture of the signal's time and frequency domain features. Compared to A/P signals, I/Q signals provide richer information for the model to differentiate. CrossTLNet interacts with both, directly learning information from each other, introducing a richer feature combination. This improvement cannot be stimulated by training with I/Q or A/P alone.

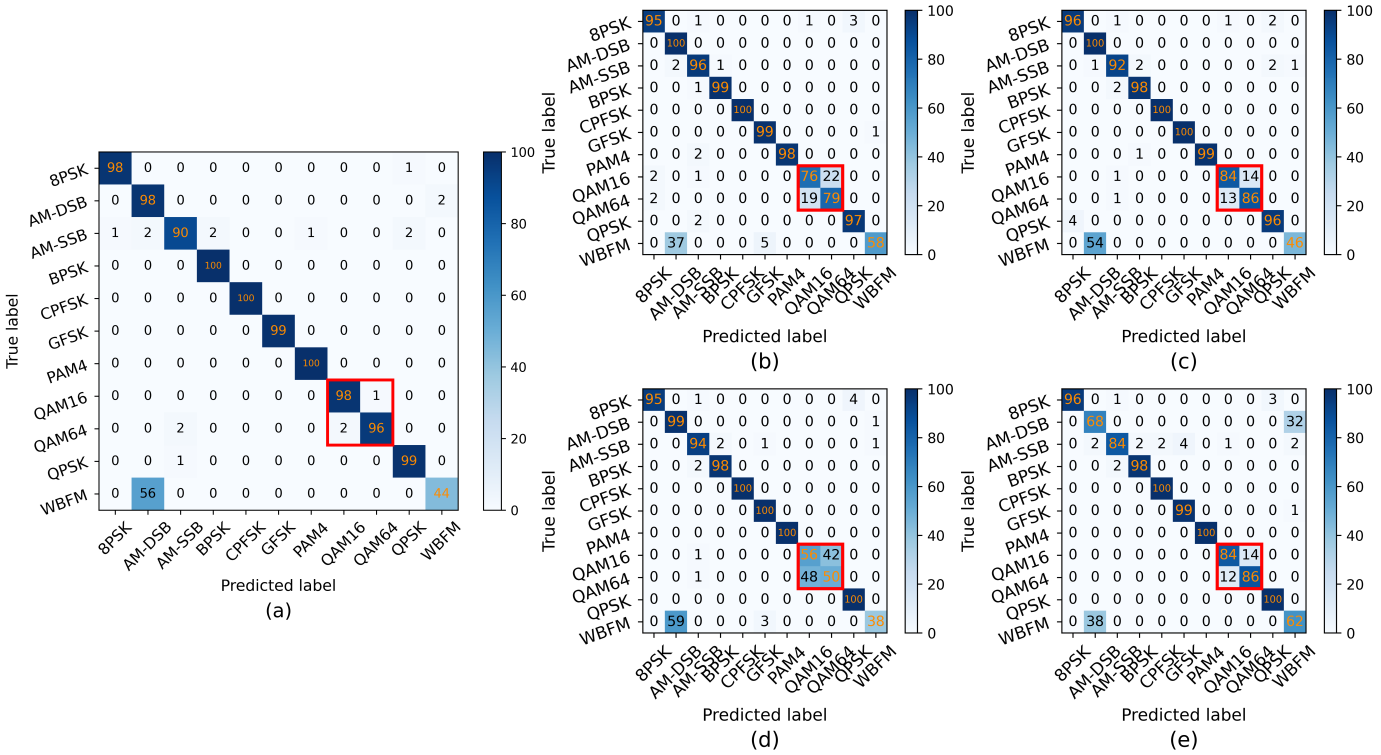

Figure 8. Confusion matrices for different models at SNR = 4 dB. (**a**) CrossTLNet. (**b**) DS-CLDNN. (**c**) PET-CGDNN. (**d**) IC-AMCNet. (**e**) LSTM. The red frame highlights the comparison between QAM16 and QAM64, two high-order modulations.

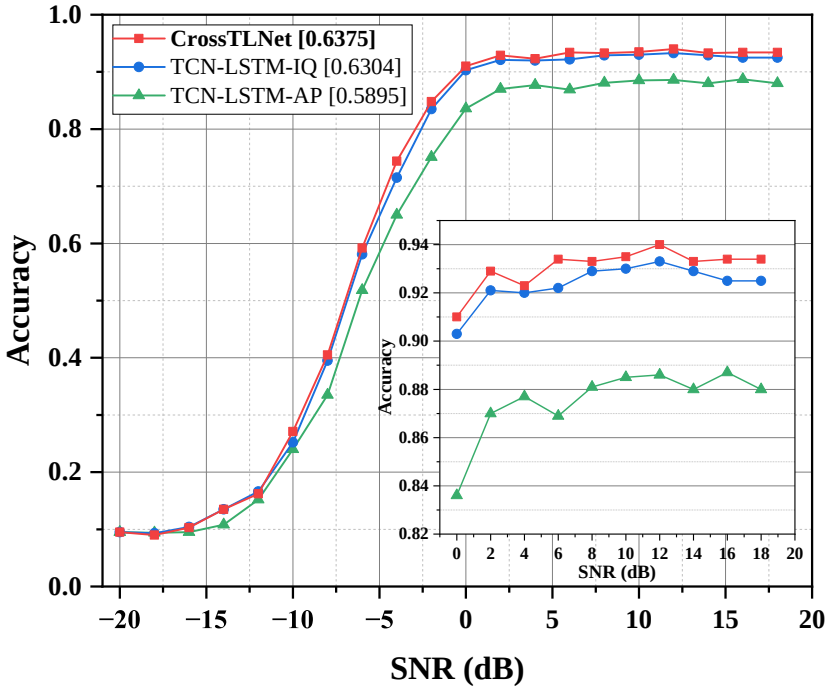

Figure 9. Comparison of CrossTLNet and algorithms only based on I/Q or A/P. The inner figure depicts a magnified view for $SNR \geq 0$ dB.

It is worth emphasizing that, despite the design of a multi-task learning structure, the proposed CrossTLNet remains a symmetric single-input–single-output neural network. This means that, during training, only a traditional cross-entropy loss needs to be optimized,

making the training process stable and straightforward. In usage, it only requires passing a single I/Q signal, just like with previous researches, which is very convenient.

Additionally, the effect of varying the number of units within the TCN–LSTM block of CrossTLNet on its overall performance is also investigated. Denoting CrossTLNet with N units in TCN–LSTM block as CrossTLNet-N, the comparison results are presented in Table 4. As N increases, the classification accuracy of CrossTLNet gradually improves. When $N = 256$ (proposed), it achieves the best accuracy of 63.75%. However, further increasing N leads to a slight decrease in accuracy. This is attributed to an excess of neurons causing overfitting issue, which in turn impacts the model's performance.

Table 4. Classification accuracy of CrossTLNet with different numbers of units in TCN–LSTM block. The bold font represents the best value for that metric.

Model	Accuracy	Accuracy ($SNR \geq 0$ dB)
CrossTLNet-128	0.6326	0.9221
CrossTLNet-192	0.6360	0.9263
CrossTLNet-256 (proposed)	**0.6375**	**0.9305**
CrossTLNet-320	0.6354	0.9264

3.3. Ablation Study

The experiments of ablation studies analyze the impact of both the TCN module and the cross-attention block on CrossTLNet, thereby providing a comprehensive evaluation of the efficacy of the proposed method. Table 5 and Figure 10 present the results from the ablation study.

Table 5. Ablation study results of CrossTLNet. The bold font represents the best value for that metric.

Model	TCN Module	Cross-Attention	Accuracy	Accuracy ($SNR \geq 0$ dB)
CrossTLNet-A			0.5476	0.8137
CrossTLNet-B		✓	0.5649	0.8353
CrossTLNet-C	✓		0.6341	0.9273
CrossTLNet (proposed)	✓	✓	**0.6375**	**0.9305**

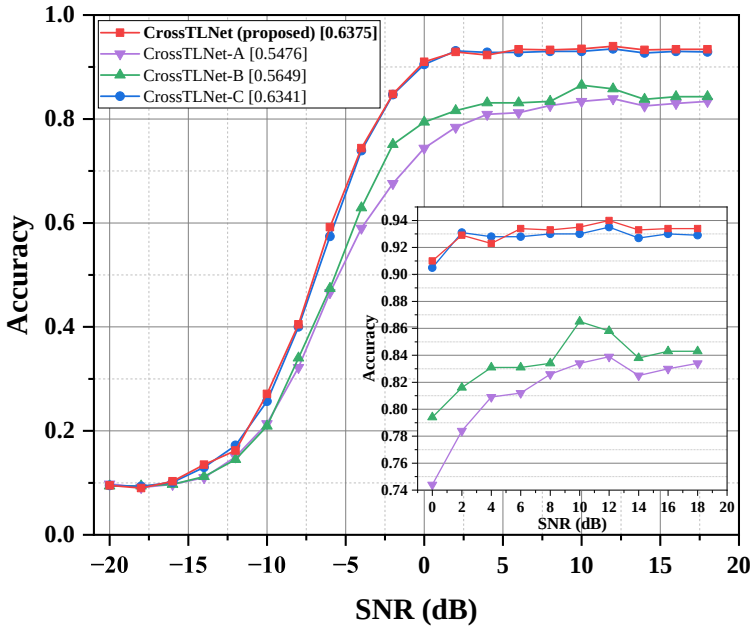

Figure 10. Comparison of CrossTLNet with different architectures in ablation study. The inner figure depicts a magnified view for $SNR \geq 0$ dB.

Regarding the TCN module, by comparing CrossTLNet-A with CrossTLNet-C, it can be observed that the use of the TCN module in CrossTLNet-C results in an overall accuracy improvement of 8.65%. Similarly, comparing CrossTLNet-B with CrossTLNet (proposed) leads to a similar conclusion. This demonstrates the effectiveness of the TCN module.

Regarding the cross-attention block, by comparing CrossTLNet-A with CrossTLNet-B, it can be observed that the use of the cross-attention block in CrossTLNet-B results in an overall accuracy improvement of 1.73%. Similarly, comparing CrossTLNet-C with CrossTLNet (proposed) leads to a similar conclusion. This demonstrates the effectiveness of the cross-attention block.

Note that the improvement of CrossTLNet (proposed) compared to CrossTLNet-C is relatively small. This is because the TCN module plays an excellent role in feature extraction, approaching the upper limit of state-of-the-art models and thus overshadowing the positive effect of cross-attention. However, when the number of units in the TCN module becomes very small, the performance of CrossTLNet decreases significantly. In this scenario, the role of cross-attention becomes more prominent, as demonstrated by the comparison between CrossTLNet-A and CrossTLNet-B. This also implies that cross-attention can play a more significant role in improving accuracy when compressing the model size in subsequent steps.

It is evident that the TCN module brings a significant improvement to the performance of CrossTLNet. Figure 11 presents the confusion matrices for CrossTLNet-B and CrossTLNet (proposed) at $SNR = 4$ dB. It can be seen that CrossTLNet-B exhibits severe confusion between QAM16 and QAM64, culminating in a decline in classification accuracy. Furthermore, the feature maps of CrossTLNet-B and CrossTLNet (proposed) for the same QAM16 and QAM64 signal ($SNR = 4$ dB) before entering the output block, i.e., the output of *stage 4*, are separately plotted, as shown in Figure 12.

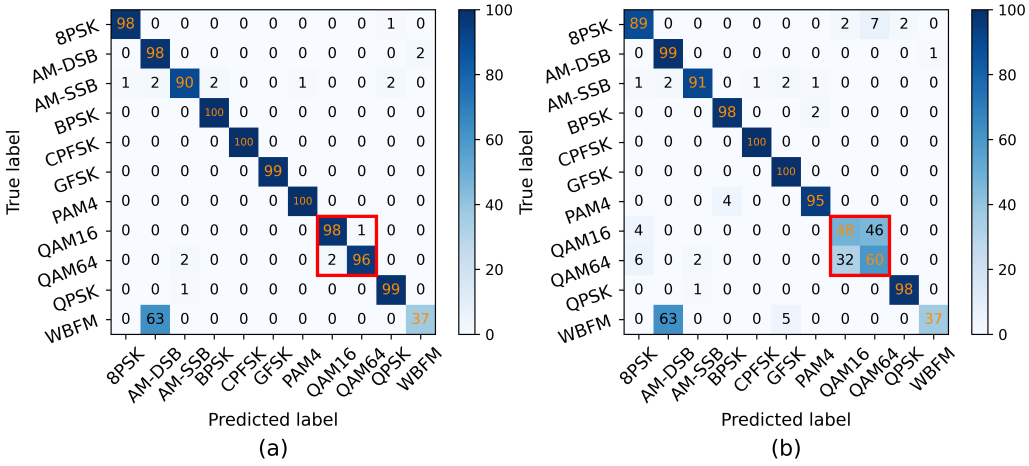

Figure 11. Confusion matrices for different CrossTLNet at SNR = 4 dB. (**a**) CrossTLNet. (**b**) CrossTLNet-B. The red frame highlights the comparison between QAM16 and QAM64, two high-order modulations.

It can be observed that the feature maps of CrossTLNet (proposed) with the TCN module are more distinct, extracting features that help distinguish QAM16 from QAM64. In contrast, the feature maps of CrossTLNet-B do not exhibit significant differences. This provides additional evidence for the efficacy of the proposed method.

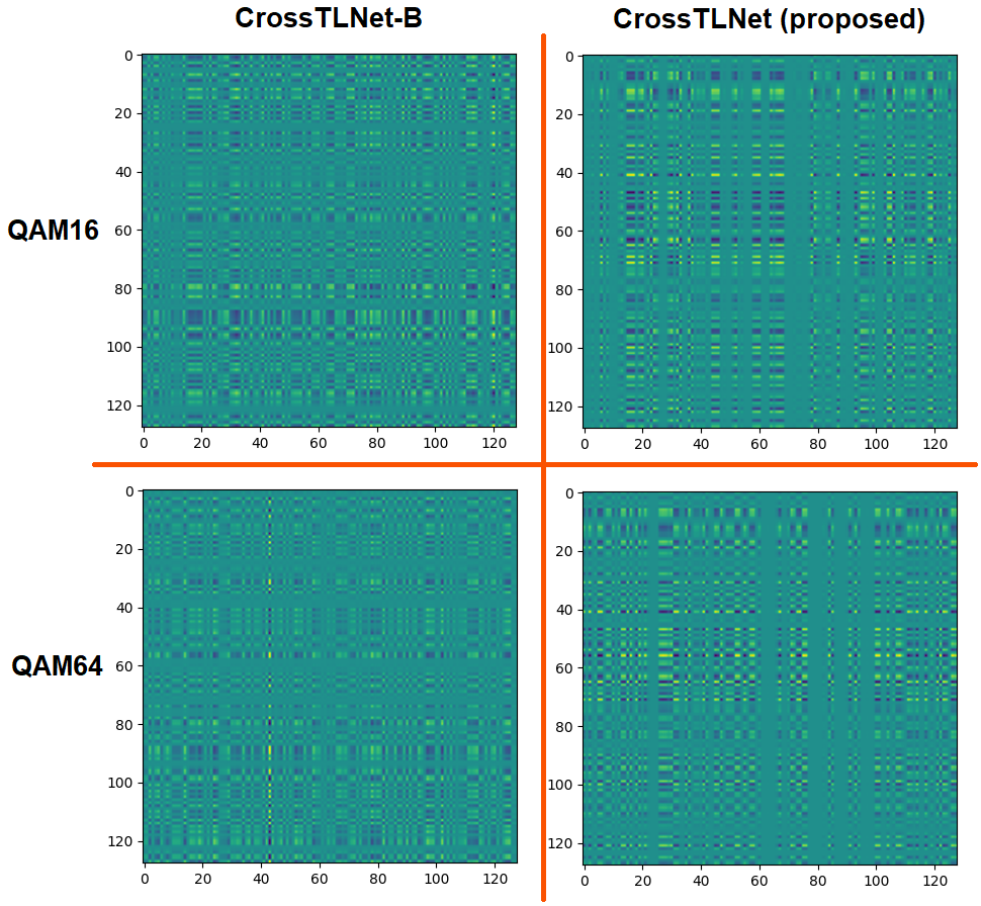

Figure 12. Feature maps of CrossTLNet-B and CrossTLNet (proposed) for QAM16 and QAM64.

3.4. Lightweight Model

To distinguish it, the lightweight CrossTLNet is denoted as Mini-CrossTLNet. The Mini-CrossTLNet trained with the designed knowledge distillation method (denoted as Mini-CrossTLNet (KD)) is compared with the Mini-CrossTLNet trained from scratch (denoted as Mini-CrossTLNet (no KD)). The result is shown in Figure 13.

It can be observed that, through the knowledge distillation method, Mini-CrossTLNet (KD) achieves a commendable accuracy of 63.44%, which is comparable to the standard CrossTLNet's accuracy of 63.75%, effectively learning the knowledge from CrossTLNet as the teacher model. In contrast, the Mini-CrossTLNet trained from scratch only achieves an accuracy of 58.3%, much lower than CrossTLNet's accuracy of 63.75%. This stark contrast underscores the effectiveness of the designed lightweighting method.

More specifically, we plot the confusion matrices for Mini-CrossTLNet (KD) and Mini-CrossTLNet (no KD) at $SNR = 4$ dB in Figure 14, aiming to explore the key distinctions causing this performance gap. It can be observed that the main difference appears in distinguishing between QAM16 and QAM64, while there is no significant performance difference for other modulation types. This is because, when the number of units in the TCN–LSTM block is reduced, under the same training conditions, the model's ability to capture long-term dependency features also weakens. When the number of units becomes too low, the model struggles to correctly differentiate between different high-order modulations. However, this does not mean that the model itself lacks the potential for accurate differentiation. Through the designed knowledge distillation method, Mini-CrossTLNet, acting as the student model, is induced to imitate and learn the intermediate layer features and output distribution of the high-performance teacher model CrossTLNet. This in turn stimulates the potential of Mini-CrossTLNet to differentiate high-order modulations, ultimately improving the classification accuracy.

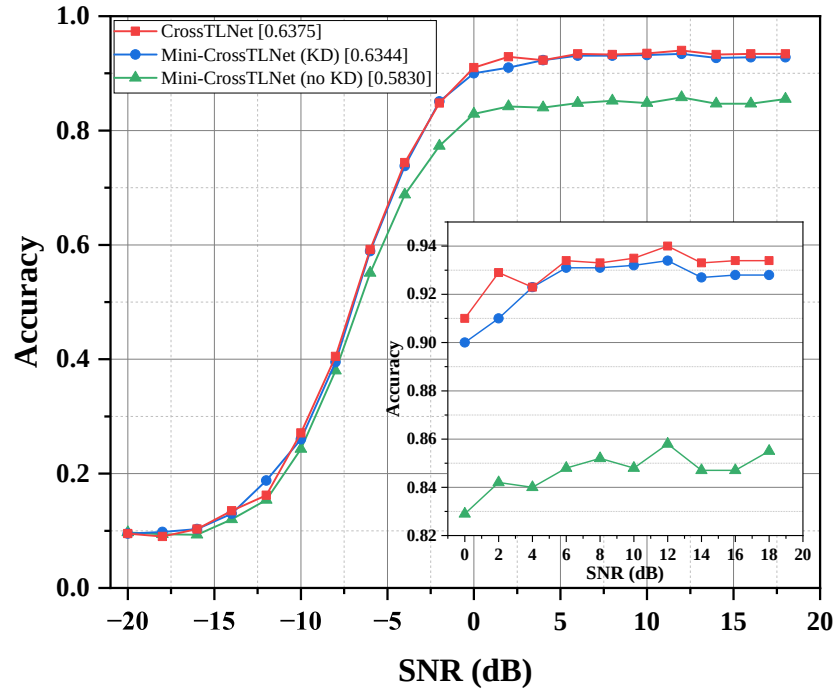

Figure 13. Comparison of different Mini-CrossTLNet training methods. *Mini-CrossTLNet (KD)* represents training with the designed knowledge distillation method, and *Mini-CrossTLNet (no KD)* represents training from scratch without knowledge distillation. The inner figure depicts a magnified view for $SNR \geq 0$ dB.

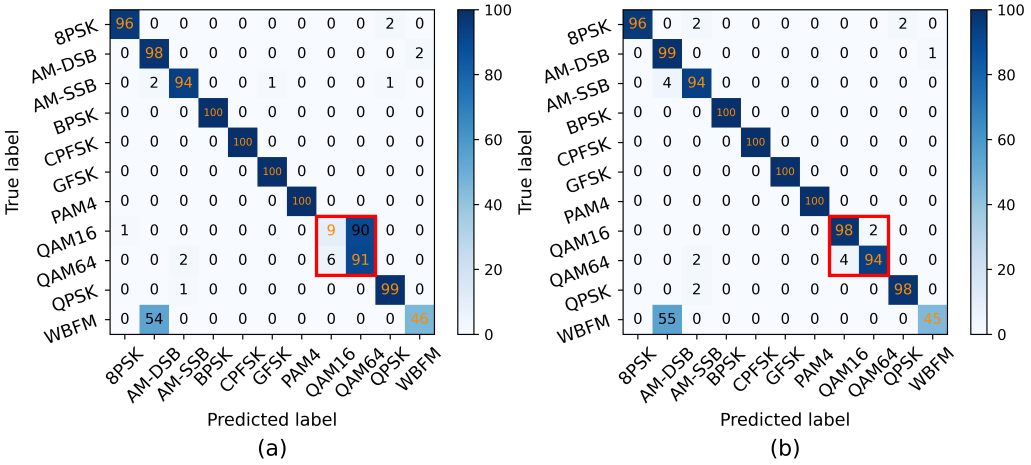

Figure 14. Confusion matrices for Mini-CrossTLNet trained with different methods at $SNR = 4$ dB. (**a**) Mini-CrossTLNet (no KD). (**b**) Mini-CrossTLNet (KD). The red frame highlights the comparison between QAM16 and QAM64, two high-order modulations.

Next, the compression performance of Mini-CrossTLNet is discussed. Compare Mini-CrossTLNet with the best-performing CTDNN, MLDNN, and DS-CLDNN in Section 3.2, and the results are shown in Table 6. It can be observed that Mini-CrossTLNet (KD) has the minimum parameters of 575K, compressing over 91.49% of parameters compared to CrossTLNet, while still maintaining comparable accuracy to state-of-the-art models. This further demonstrates the excellent design of the proposed model.

In summary, Figure 7 illustrates the comparison results between the proposed CrossTLNet and other baseline models. Figure 9 showcases the efficacy of the designed multi-task learning structure, while the results of the ablation studies in Figure 10 confirm the effectiveness of the proposed method. Additionally, Figure 13 demonstrates the potency of the designed model lightweighting method.

Table 6. Comparison of model parameters and performance for different models. The bold font represents the best value for that metric.

Model	Parameters	Accuracy
CrossTLNet	6760.5 K	**0.6375**
Mini-CrossTLNet (KD)	**575.0 K**	0.6344
CTDNN	2577.2 K	0.6349
MLDNN	899.25 K	0.6337
DS-CLDNN	1144.7 K	0.6176

4. Conclusions

This paper has proposed a hybrid multi-task learning model named CrossTLNet for accurate AMC. The proposed CrossTLNet receives input signal in the form of I/Q and processes it through a pre-processing block to yield I/Q and A/P components. By employing a symmetric multi-task learning framework, the features associated with these two tasks can be learned separately. To mitigate the issue of confusion in high-order modulations, a method with TCN–LSTM is proposed which enhances the model's capability to capture long-term dependency features. Simultaneously, a cross-attention method is innovatively introduced to enable the interaction of features from both I/Q and A/P dimensions. Moreover, considering real-time or resource-limited scenarios, the proposed model is lightweighted with the designed knowledge distillation method. Experiments on the public dataset RML2016.10A have revealed that CrossTLNet attains an overall classification accuracy of 63.75%. For high-order modulations like QAM16 and QAM64, the proposed method exhibits its efficacy with a classification accuracy of 96.55% when $SNR \geq 0$ dB. Through the lightweighting method, the proposed model managed to compress over 91.49% of the parameters while still maintaining a comparable level of accuracy to state-of-the-art models. The results obtained in our research strongly support the potential of CrossTLNet in addressing practical problems, and we believe it will showcase its effectiveness across diverse real-world scenarios.

Author Contributions: Conceptualization , G.G.; Methodology, G.G.; Software, G.G.; Validation, B.L.; Formal analysis, B.L.; Investigation, B.L.; Resources, X.H.; Writing—original draft, G.G.; Writing—review & editing, X.H.; Supervision, F.M.G.; Project administration, W.W.; Funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China (No. 2020YFC1511801).

Data Availability Statement: Data are contained within the article.

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

References

1. Wang, Y.; Wang, J.; Zhang, W.; Yang, J.; Gui, G. Deep learning-based cooperative automatic modulation classification method for MIMO systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4575–4579. [CrossRef]
2. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In Proceedings of the Engineering Applications of Neural Networks: 17th International Conference, Aberdeen, UK, 2–5 September 2016; pp. 213–226. [CrossRef]
3. Liu, X.; Yang, D.; El Gamal, A. Deep neural network architectures for modulation classification. In Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 915–919. [CrossRef]
4. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. [CrossRef]
5. Zhang, F.; Luo, C.; Xu, J.; Luo, Y. An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation. *IEEE Commun. Lett.* **2021**, *25*, 3287–3290. [CrossRef]
6. Zhang, Z.; Luo, H.; Wang, C.; Gan, C.; Xiang, Y. Automatic modulation classification using CNN–LSTM based dual-stream structure. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13521–13531. [CrossRef]
7. Wang, M.; Fan, Y.; Fang, S.; Cui, T.; Cheng, D. A Joint Automatic Modulation Classification Scheme in Spatial Cognitive Communication. *Sensors* **2022**, *22*, 6500. [CrossRef] [PubMed]

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. Ying, S.; Huang, S.; Chang, S.; Yang, Z.; Feng, Z.; Guo, N. A convolutional and transformer based deep neural network for automatic modulation classification. *China Commun.* **2023**, *20*, 135–147. [CrossRef]
10. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271. https://doi.org/10.48550/arXiv.1803.01271.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010. [CrossRef]
12. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531. https://doi.org/10.48550/arXiv.1503.02531.
13. Tung, F.; Mori, G. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1365–1374. [CrossRef]
14. Huang, T.; You, S.; Wang, F.; Qian, C.; Xu, C. Knowledge distillation from a stronger teacher. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 33716–33727. [CrossRef]
15. DeepSig Team. RF Datasets For Machine Learning. Available online: https://opendata.deepsig.io/datasets/2016.10/RML2016.10a.tar.bz2 (accessed on 1 November 2023).
16. Zhang, F.; Luo, C.; Xu, J.; Luo, Y.; Zheng, F.C. Deep learning based automatic modulation recognition: Models, datasets, and challenges. *Digit. Signal Process.* **2022**, *129*, 103650. [CrossRef]
17. O'shea, T.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [CrossRef]