electronics

MDPI

Article

TMRN-GLU: A Transformer-Based Automatic Classification Recognition Network Improved by Gate Linear Unit

Yujun Zheng, Yongtao Ma * and Chenglong Tian

School of Microelectronics, Tianjin University, Tianjin 300072, China; yujunzheng@tju.edu.cn (Y.Z.); tianchenglong@tju.edu.cn (C.T.)
* Correspondence: mayongtao@tju.edu.cn

Abstract: Automaticmodulation recognition (AMR) has been a long-standing hot topic among scholars, and it has obvious performance advantages over traditional algorithms. However, CNN and RNN, which are commonly used in serial classification tasks, suffer from the problems of not being able to make good use of global information and slow running speed due to serial operations, respectively. In this paper, to solve the above problems, a Transformer-based automatic classification recognition network improved by Gate Linear Unit (TMRN-GLU) is proposed, which combines the advantages of CNN with a high efficiency of parallel operations and RNN with a sufficient extraction of global information of the temporal signal context. Relevant experiments on the RML2016.10b public dataset show that the proposed algorithm not only has a significant advantage in the number of parameters compared with the existing algorithms, but also has improved recognition accuracy under various signal-to-noise ratios.In particular, the accuracy of the proposed algorithm improves significantly compared with other algorithms under low signal-to-noise ratio conditions. The accuracy is improved by at least 9% at low signal-to-noise ratio (6 dB) and about 3% at high signal-to-noise ratio (>2 dB).

Keywords: deep learning; automatic modulation recognition; self-attention mechanism; Transformer-based network

Citation: Zheng, Y.; Ma, Y.; Tian, C. TMRN-GLU: A Transformer-Based Automatic Classification Recognition Network Improved by Gate Linear Unit. *Electronics* **2022**, *11*, 1554. https://doi.org/10.3390/electronics11101554

Academic Editors: Roland Gautier, Florin Popescu, Annamaria Sarbu and Angela Digulescu

Received: 30 March 2022
Accepted: 10 May 2022
Published: 12 May 2022

1. Introduction

Wireless communication plays a key role in the development of various industries around the world. However, with the rapid growth of the number of Internet mobile devices and the continuous emergence of new business types, as well as the sustainable development of the next-generation mobile communication technology represented by 5 g technology, the lack of wireless spectrum resources has become a problem that must be faced [1]. Due to the scarcity of spectrum resources, in order to utilize spectrum resources efficiently, reasonably and economically, and maximize the value of licensed spectrum, Cognitive Radio (CR) technology, which can intelligently improve spectrum utilization, emerges at the right moment [2]. Automatic Modulation Recognition (AMR) refers to a technology that determines the modulation mode of an unknown signal by analyzing the electromagnetic characteristics, spectral characteristics, and statistical characteristics [3]. AMR technology is the technical basis of wireless communication parameter estimation, signal demodulation, and spectrum management in CR. Not only that, but it is also the cornerstone of optimizing spectrum utilization efficiency and realizing heterogeneous wireless network. Whether in military or civilian fields, accurate and efficient automatic modulation recognition (AMR) is an important means to realize spectrum resource monitoring and management at the same time [4].

For example, the AMR works in electronic warfare scenarios as shown in the Figure 1. In this process, the identification of the modulation of the interfering signal will be used as a crucial step. This is because the correct demodulation of the signal is a prerequisite

for the further decryption of the interfered or intercepted content. In addition to this, AMR can serve to save the bandwidth for sending modulated information in adaptive communication systems in civil communication, which is the part that must be occupied without AMR. If the communicating parties take part in non-protocol communication and the dual transmission cannot exchange modulation information via protocol, then automatic modulation identification will be necessary for successful communication.

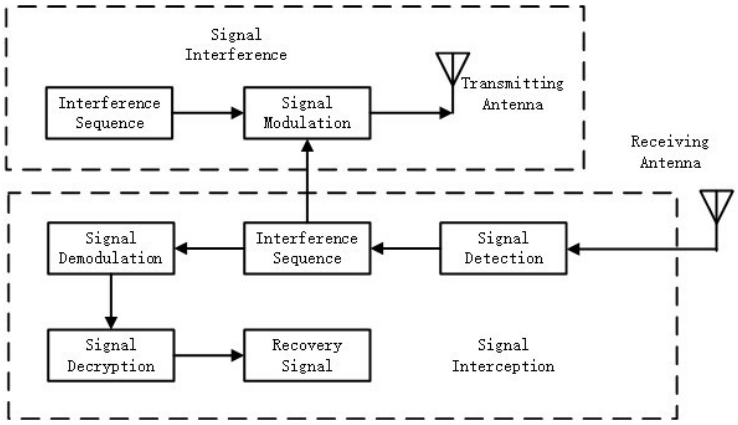

Figure 1. Military application of AMR.

In addition, as an important part of the civil communications regulatory field, wireless spectrum monitoring is important in frequency, time and space for a wide range of applications, such as spectrum enforcement by regulatory agencies, generating coverage maps for wireless operators, and applications that include wireless signal detection and localization. AMR is an integral part of spectrum enforcement. This classifier can help identify suspicious transmissions in specific wireless bands. In addition, AMR is important for interference detection and wireless environment analysis.

Deep learning (DL) is a kind of machine learning, and machine learning is the necessary path to achieve artificial intelligence. The concept of deep learning originates from the study of artificial neural networks, and a multilayer perceptron (MLP) containing multiple hidden layers is a deep learning structure. Deep learning discovers distributed feature representations of data by combining lower-level features to form more abstract higher-level representations of attribute classes or features. The motivation for studying deep learning is to build neural networks that mimic the human brain for analytical learning, which mimics the mechanisms of the human brain to interpret data. In recent years, computer hardware processing speed has developed rapidly, making deep learning techniques continue to make new breakthroughs. Deep learning has been fruitful in areas such as natural language processing [5], computer vision [6,7], intelligent health diagnosis [8] and speech signal processing [9,10] due to its powerful representational learning and nonlinear fitting capabilities. Deep learning is an end-to-end self-training technique that does not require complex and inefficient artificial expert features, so it has higher hidden information extraction and better generalization capabilities than traditional algorithms for modulation recognition tasks [11].

This feature can help deal with complex nonlinear distortions such as channel effects, receiver thermal noise, and phase clock shifts. In addition, deep learning models have excellent inductive migration capability, an ability that can improve the ability of cognitive systems to recognize complex signals and compare similar signals.

Therefore, this paper focuses on how to use Transformer, a new type of network in deep learning, to better solve the AMR problem. The rest of this paper is organized as follows. Section 2 focuses on work related to the AMR domain using both traditional algorithms and deep learning and concludes the section with a description of how our work differs from other published work. Section 3 describes the background related to the AMR problem, such as information on how the channel was modeled and the dataset

used for subsequent experiments. The details of TMRN-GLU are introduced in Section 4. The relevant experiments performed on the publicly available dataset are in Section 5. The factors that affect network performance are discussed in Section 6. Section 7 draws conclusions about this paper.

2. Related Work

Traditional AMR methods can be broadly classified into two categories: maximum likelihood hypothesis testing methods and statistical pattern recognition methods based on feature extraction. Maximum likelihood hypothesis testing methods, as explored in [12–15], first processes the likelihood function of the signal to obtain sufficient statistics for classification, and then compares it with an appropriate threshold. Such methods can theoretically achieve high recognition performance, but they always require too many known parameters. The likelihood function is very complicated, the calculation amount is large, and it is difficult to process in real time. The other is the feature-based method, whose recognition rate mainly depends on the extraction of signal feature parameters and the design of the classifier. This method does not require priori information or requires less priori information, and this method is generally divided into three parts: signal preprocessing, feature parameter extraction and classification recognition [16–19]. Although the feature based methods are theoretically suboptimal, they are much less difficult to implement and apply than the likelyhood methods. And for a well-designed feature based method, the performance of the suboptimal result can be very close to that of the likehood method. Each of these traditional algorithms has its own limitations, for example, the likelihood ratio-based approach requires all prior knowledge, while the feature-based approach relies too much on the merits of the feature extraction method.

In the field of signal processing, deep learning is a popular option for researchers to study AMR problems. Convolutional neural networks (CNN) are constructed by imitating the visual perception mechanism of biology, and can perform supervised learning and unsupervised learning. Small computational effort to learn grid-like topology features such as pixels and audio, with stable effects and no additional feature engineering requirements on the data. Recurrent Neural Network (RNN) is a type of recurrent neural network that takes sequence data as input, performs recursion in the evolution direction of the sequence, and connects all nodes (recurrent units) in a chain. Recurrent neural networks have applications in natural language processing, such as speech recognition, language modeling, machine translation, etc., and are also used in various time series forecasting. There are many similarities between radio frequency signals and audio signals. For example, different orders of quadrature amplitude modulation (MQAM) have different amplitude and phase jumps when transmitting different symbols. This is evident in the constellation diagram and this is advantageous for processing using deep learning. O'Shea et al. [20,21] introduced convolutional neural networks to the AMR domain, and for the first time directly used raw time-domain in-phase and quadrature signals as input to the network without any feature extraction. Not only has he conducted extensive research on the design of AMR networks, but as an expert in wireless communication channel research, he produced some very high quality public datasets for modulated signal identification [22]. The author in [23] showed that RNNs are effective for modulated recognition of raw time series sampled signals and have better accuracy than the related work using CNN networks in the same period, and recommended a efficient structure based on two-layer gated recurrent unit (GRU) network. Rajendran showed that a LSTM based model can learn good representations of variable length time domain sequences. And Liao in [24] further improving on this work, he used sequential convolutional recurrent networks, he combined the advantages of CNN and LSTM, the simulation proved that the network structure was more effective. Wei et al. [25] constructed a novel end-to-end sequence-based network consisting of a shallow convolutional neural network, a bi-directional long and short-term memory (Bi-LSTM) network enhanced with a self-attentive mechanism, and a dense neural network. That network allowed not only to obtain historical and future

information from a sequence of features across domains at a point in time, but also to automatically find useful and important features to facilitate recognition performance.

A new network structure Transformer [26] has recently emerged in the field of deep learning, and it has achieved excellent results in the field of natural language processing. AMR as a sequence classification problem, and introducing Transformer-related structures into AMR is a worthwhile discussion. We propose a Transformer-based modulation recognition network and replace the original feedforward network (FFN) in Transformer with gated linear units and some other improvements. We name this AMR structure as TMRN-GLU. In this work, we distinguish from other published AMR algorithms by mainly using the encoder structure as the main core unit of the network instead of CNN or RNN networks, which allows the network to not only perform parallel computation like CNN, but also to obtain excellent temporal sequence processing like RNN, LSTM capability. After extensive evaluation, TMRN-GLU has better performance with less number of parameters than existing algorithms, and its operation speed also has some advantages.

3. The Background on the AMR Problem

In this article we use a deep neural network as a classifier of the modulation kind of the received wireless signal. As an end-to-end classification method, deep learning has the advantage that it can automatically generalize the required parameters from the training data.

Usually, after the initial processing of the modulated wireless signal by the receiver we can obtain its baseband complex envelope, its general expression is:

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

$n(t)$ is the additive white noise at time t, h refers to the channel impulse response, $\otimes$ means mathematical convolution, and $s(t)$ in Equation (1) is defined as follows:

$$s(t) = a_i e^{j2\pi\Delta f t} e^{j\theta} \sum_{k=1}^{K} e^{j\phi_k} s_k^{(i)} P_{pulse}(t - (k-1)T - \sigma), 0 \le t \le KT \tag{2}$$

It is the baseband complex envelope of the received signal without considering noise. In Equation (2), a_i indicates signal amplitude, Δf is is the carrier frequency offset. θ is the carrier phase shift introduced by the propagation delay, ϕ_k is the phase jitter, $s_k^{(i)}$ represents one of K complex symbols taken from a modulation type, σ is the normalization of the time offset between the transmitter and the signal receiver, $P_{pulse}(t)$ is the emission pulse shape. The goal is to identify the modulation type from the received signal r(t). We use a machine learning classifier based on the Transformer network architecture with self-attention mechanism, where the training dataset is first processed to set the network parameters, and then the classification accuracy is calculated based on the classification output of the test dataset.

We used the RadioML2016.10b dataset generated in [20] as the base dataset for our study. For digital modulation, the Shakespeare set in ASCII format was used. For analog modulation, a continuous speech signal was used as the data source, which consists mainly of the original speech with some off times. Ten typical modulations were selected: eight digital and two analog modulations. In all subsequent experiments, we divided the data set into a training set with 720,000 data and a test set with 480,000 data. Among them, the training set is also extracted as the validation set for K-fold cross-validation at each training by 1/5 of the non-repeat during training. This allows the training and correlation tests to fully reflect all the characteristics of the channel represented by the data set, avoiding the impact of errors and making the results statistically significant. Details of the modulation types and signal-to-noise ratios of the dataset are given in the following Table 1.

Table 1. RadioML2016.10b Dataset Parameters.

Parameter Name	Parameter Values
Modulations	8 digital modulations: BPSK, QPSK, 8PSK, QAM16, QAM64, BFSK, CPFSK, PAM4 2 analog modulations :WB-FM, and AM-DSB
Number of samples per SNR per Category	6000
Length per sample	128
Signal format	In-phase and quadrature(IQ)
Signal dimention	2×128 per sample
Duration per sample	128 µs
Sampling frequency	1 MHz
SNR range	[−20 dB:2 dB:18 dB]
Total number of samples	1,200,000
Number of train and validation samples	720,000
Nunber of test samples	480,000

For the channel model, physical environmental noises like thermal noise and multipath fading were simulated. The models for generating random channel and device imperfections include sample rate offset model, noise model, center frequency offset model and fading model. In this paper we use the original signal in IQ format as input, because it is obviously more flexible and simpler to separate the real and imaginary parts in mathematical operations and processing in hardware than its complex form.

4. Proposed Automatic Modulation Recognition Net Based on Self-Attention

In the past, for AMR-related tasks researchers usually used RNN or CNN. In the above two types of algorithms, the convolutional kernel in CNN is able to separately extract local features of images or signals, etc. local features of images or signals, etc., but the single-layer convolutional kernel in early research In recent years, this problem has been solved by making the CNN deeper and jumping over. In recent years, this problem has been solved by making the CNN deeper and jumping to cover. But at the same time, the indispensable pooling layer will discard the location information and cause information loss; while the computation of RNN (or LSTM, GRU, etc.) is sequential, and the relevant The algorithms can only be computed sequentially from left to right or from right to left. This mechanism poses two problems: firstly, the computation of the time slice depends on the computation results of the previous moments, which is a big problem. This mechanism poses two problems: first, the computation of the time slice depends on the computation results of previous moments, which largely limits the parallelism of the model. Second, the prior information is lost during the sequential computation. Although researchers have proposed gate structures such as the LSTM, they only alleviate the long-term dependence loss to a certain extent. Although researchers have proposed gate structures such as LSTM, they have only alleviated the problem of long-term dependence loss to a certain extent, but not completely solved it. The problem is not completely solved. Unlike other deep learning methods, the network structure proposed in this paper consists entirely of an attention module (Self-Attention), except for the first feature extraction layer. Its relevant features are a good solution to the regret of CNN and RNN.

4.1. Multi-Head Attention and the Encoder Architecture Based on It

4.1.1. Multi-Head Self Attention

The attention mechanism has been applied in deep learning for a long time, for example, the article [27] simulates the human characteristics of processing visual information with some focus, and combines RNN with attention to get better image classification results at that time. And the article [28] proposes the use of the Attention mechanism for simultaneous translation and alignment on machine translation tasks, and their work is the first to apply the Attention mechanism to the field of NLP. The attention mechanism mentioned above can effectively improve the generalization ability of the model. Recently a new DL

network, Transformer, which is composed entirely of self-attentive mechanism [26]. It combines the advantages of convolutional neural networks and recurrent neural networks and is highly unified, and has achieved outstanding results in the fields of natural language processing [29] and image processing [30].

The Transformer-based net model is mostly based on two modules, Encoder and Decoder, which were first conceptualized from the translation task in natural language processing. In this article, we will only use Encoder. So we will explain the self-attentive mechanism used and the Encoder composed by it.

The essence of the Attention Mechanism is to identify which features in the input are important for the target and which features are not by generating a weighting coefficient to weight the input to sum up for a given target. In order to implement the Attention Mechanism, we treat the input data as <Key, Value> key-value pairs and calculate the similarity coefficient between Key and Query based on the query value Query in the given task objective to obtain the weight coefficient corresponding to the Value value, and then use the weight coefficient to weight the Value value to obtain the output. Before the appearance of the Transformer-based model, attention was not a unified model, it was just a mechanism by which Query, Key and Value were sourced in different ways in different application domains, meaning that different domains had different implementations. For convenience, Q, K, V is used later to refer to Query, Key, Value.

The self-attentive mechanism, proposed by the Google team and applied to the Transformer language model, can be used separately in encoding or decoding. Compared to the original attention mechanism, it focuses more on the internal connection of the inputs, the difference being that Q, K and V come from the same data source, i.e., Q, K and V are derived from the same matrix through different linear transformations, it is showed in Figure 2. The output of Self-attention is calculated as follows:

$$Attention(\mathrm{Q},\mathrm{K},\mathrm{V}) = softmax\left(\frac{\mathrm{QK^T}}{\sqrt{D_k}}\right)\mathrm{V} = \mathrm{AV} \tag{3}$$

A is the attention matrix,$\sqrt{D_k}$ is the scaling factor. Equation (3) is named scaled dot-product attention, the purpose of scaling is to alleviate the gradient disappearance problem caused by softmax.

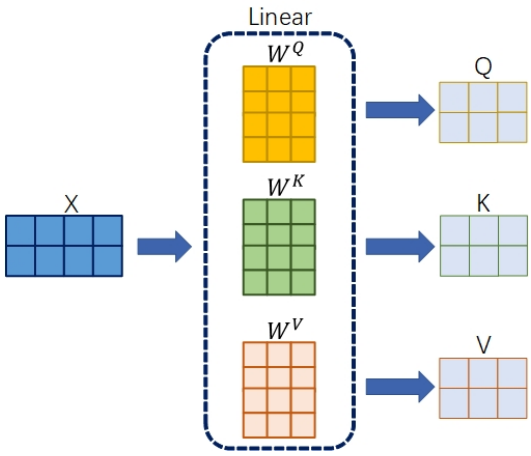

Figure 2. The generation process of Query, Key, Value.

For example, let the input sequence be $X = [x_1, x_2, \ldots, x_n]$, and the output sequence $B = [b_1, b_2, \ldots, b_n]$, then the process of obtaining b_1 from X by self-attentiveness is shown in Figure 3. First the input sequence is mapped to generate different q_i, k_i, and v_i mapping vectors for each element, and each element corresponding to q_i and all k_i is obtained through a series of calculations to obtain the attention weighting of that element corresponding to each other element, which is combined with the corresponding v. The final sum of the

above results is the output corresponding to that element. The other elements in the output sequence B are calculated in the same way. And this process is fully computable in parallel, without waiting for the previous time slice to be computed like LSTM or RNN. At the same time, the attention mechanism considers all elements of the input sequence during the operation, and its computation is global.

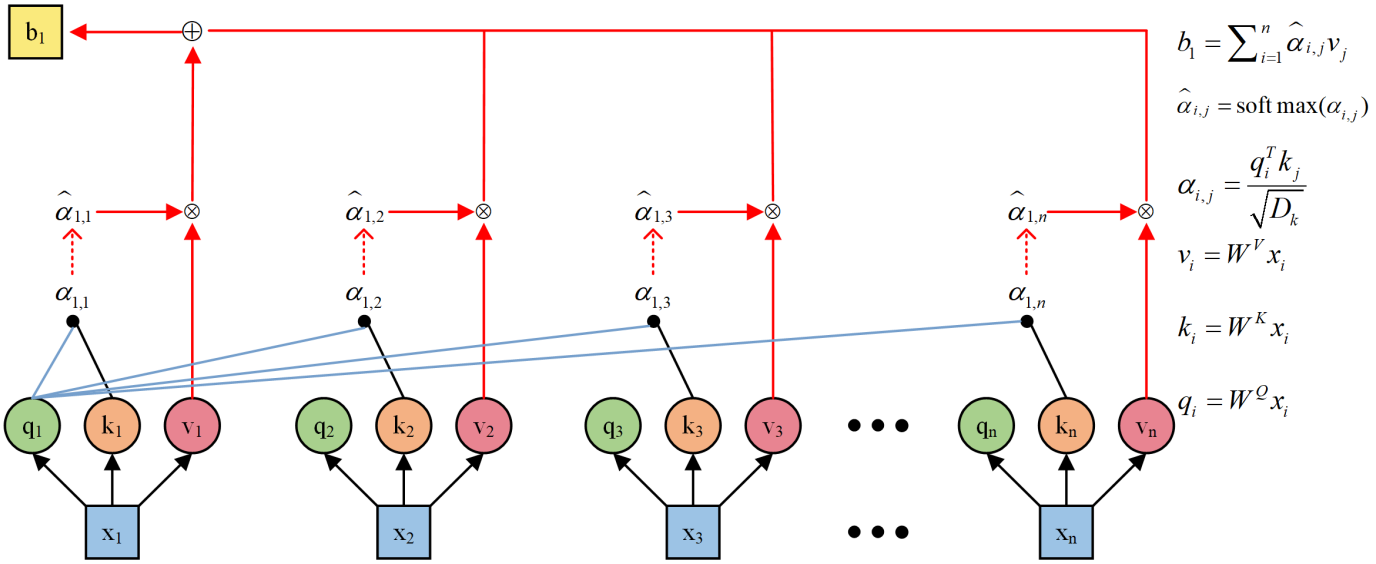

Figure 3. The process of obtaining b_1 in the output sequence B from the input sequence X by self-attentive computation.

The Table 2 summarizes the computational complexity of different kinds of networks for an input sequence of length n and dimension d, the number of sequential operands, and the path lengths required for the interaction of two elements of distance n.

Table 2. Comparison of different network types.

Network Type	Complexity Per-Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 d)$	$O(1)$	$O(1)$
RNN	$O(nd^2)$	$O(n)$	$O(n)$
CNN	$O(knd^2)$	$O(1)$	$O(log_k(n))$

From the above discussion, the disadvantage that traditional sequence models such as LSTM must be sequential is well solved, because the computation of Q, K, and V is synchronized for all positions on the sequence, which can also be said to be parallelized processing.

Analogous to the role of using multiple filters simultaneously in CNNs, here multi-headed attention is used to increase the diversity of the network like Figure 4. Multi-headed attention allows the model to jointly focus on information from different locations and different subspace representations.

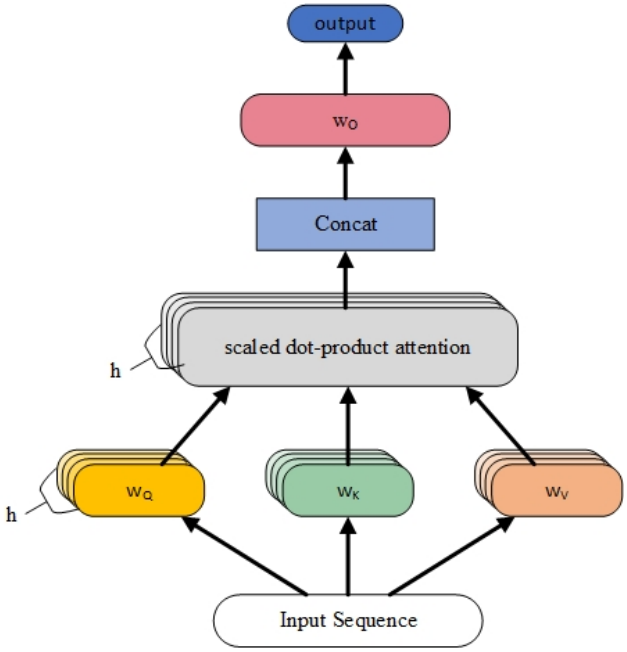

Figure 4. Schematic of Multi-Head Attention.

And Equation (4) shows the mathematical expressions of the output of multi-head attention:

$$\begin{aligned} MultiHead(\mathrm{Q},\mathrm{K},\mathrm{V}) &= Concat(head_1, head_2, \cdots, head_H)W_O \\ head_i &= Attention(XW_{\mathrm{Q}}^i, XW_{\mathrm{K}}^i, XW_{\mathrm{V}}^i) \end{aligned} \tag{4}$$

4.1.2. Composition of the Encoder and Its Improvement with GLU

The original Encoder in [26], with the multi-headed attention mentioned above as the core unit, has the structure shown in the Figure 5.

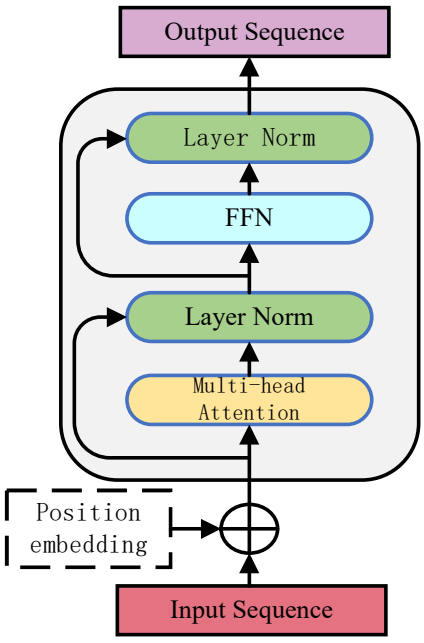

Figure 5. The original encoder structure in paper [26].

The Encoder block consists of a multi-headed attention, and a feedforward network(FFN). A positional embedding is first appended to the input sequence in the original Encoder, because there is no absolute positional relationship between words and words,

image blocks and image blocks as natural language processing and image processing tasks, so it is necessary to artificially assign a trainable location information. The embedded sequences, which comes from the embedding transformation of each word in NPL and from the mapping of image blocks in image processing, are first fed into the multi-headed attention layer after appending over the location encoding, mapped from the input sequence to the attention-weighted output sequence, and fed into the feed-forward network. The feedforward network in the original encoder consists of two fully connected layers, which are defined as in Equation (5).

$$\begin{aligned} \mathrm{FFN}(x, W_1, W_2, b_1, b_2) &= W_2(\mathrm{relu}(xW_1 + b_1)) + b_2 \\ &= \max(0, xW_1 + b_1)W_2 + b_2 \end{aligned} \tag{5}$$

However, when performing the related task on wireless signals, since the signals are sampled in temporal order, the wireless signals themselves contain location information and no additional signals need to be added manually. Therefore, the N Encoder layers used in this paper directly use the feature matrix extracted by the convolution layer as the input.

Gate linear unit (GLU) is presented in the article [31] and its structure, although simple, is defined as Equation (6):

$$\mathrm{GLU}(x, U, V, b, c) = \sigma(xU + b) \odot (xV + c) \tag{6}$$

In article [31], this gating structure makes CNNs more competitive in modeling methods for language models compared to recurrent neural networks, taking full advantage of the fast speed of parallelized computation of CNNs. According to the article [32], in terms of its computation, the self-attention mechanism is self-learning which part of the features need to be paid attention to by the data, while the convolutional network is considered to specify the convolutional kernel to learn the features in a fixed spatial range, so the CNN can be seen as a special attention mechanism to some extent. Therefore, we try to use GLU, which is used in CNN to obtain contextual information, to replace FFN in Encoder so that it reinforces contextual relations as Figure 6.

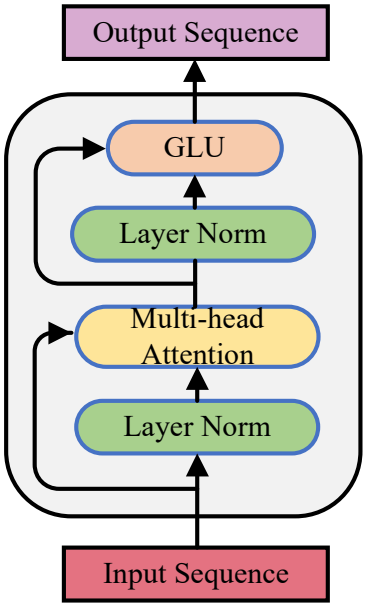

Figure 6. Schematic diagram of the improved encoder.

We found that the first linear transformations of GLU and FFN are similar without considering the activation function, so we tried to replace FFN using GLU and replace the

activation function from sigmoid to GELU, a stronger activation function for the task of processing sequences, and the transformed $\mathrm{FFN}_{\mathrm{GEGLU}}$ is formulated as follows.

$$\mathrm{FFN}_{\mathrm{GEGLU}}(x, U, V, W_O) = (\mathrm{GELU}(xU) \odot xV)W_O \tag{7}$$

Compared with the original FFN, though the trainable weight matrix is changed from two to three. But by reducing the second dimension of U and V with the first dimension of wo by $1/2$, the amount of parameters of $\mathrm{FFN}_{\mathrm{GEGLU}}$ is reduced by $1/4$ compared to FFN.

The effect of this item on the improvement of the encoder will be discussed in the Section 5.

4.2. Tmrn-Glu:Transformer-Based Automatic Modulation Recognition Network Improved by Gate Linear Unit

The model for automatic modulation recognition based on Transformer architecture with gate linear unit is proposed in Figure 7. The input of the model is a time series complex signal that splits the real part from the imaginary part, also called in-phase and quadrature(IQ) signal. Its length is 128 and its dimension is 2. The input signal first passes through a one-dimensional convolutional layer, which kernal size is k with an input channel of 2 and an output channel of n_c. This output dimension is also the input dimension of Encoder. After resize, the length is swapped with the dimension. Then, the feature matrix extracted by the CNN is fed to the Encoder layers. We choose SELU when an activation function is needed on the overall network framework, which enables the mean value of the activated vectors to be as zero as possible, speeding up the convergence of the model, which is especially important for signals from different SNRs.

$$\mathrm{SELU}(z) = \lambda \begin{cases} z & z > 0 \\ \alpha(\exp(z) - 1) & z <= 0 \end{cases} \tag{8}$$

There are N encoder layers and each Encoder has h heads with input dimension n_c. Each Encoder receives a matrix of shape $128 \times n_c$ and outputs a matrix of the same shape. The features generated by the first 4 heads of the last Encoder are extracted and flattened, the shape of that output vector is $d = 4n_c$. Because of the global attention property, the output vector already contains the information needed to classify and identify the input signal after a good training. The last is a simple classification network that maps the feature vector to the number of signal species through two linear layers with activation functions. The output vector is passed through the Softmax function to obtain the probability that the signal may be of each type of modulation.

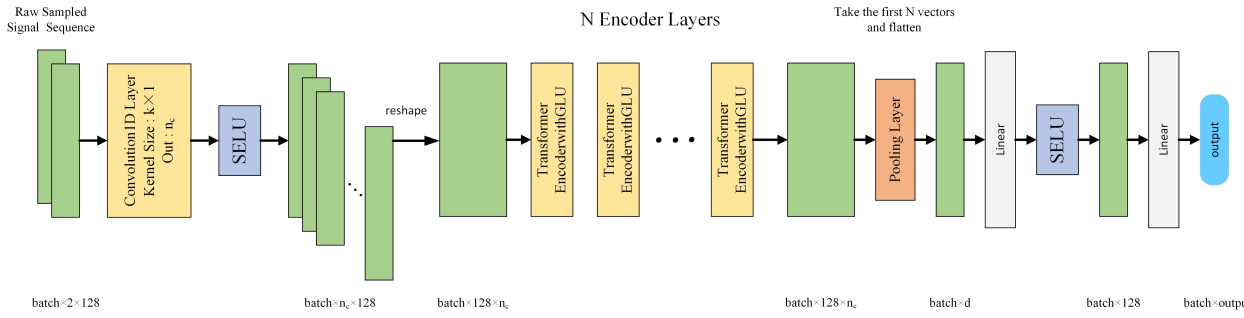

Figure 7. Transformer-based automatic modulation recognition network improved by gate linear unit.

In this paper we use the weight initialization method proposed in the article [33] in the convolutional layer, and we find that when the convolutional layer is not initialized with effective weights, it is difficult to converge when the model is trained. The training-time optimizer uses AdamW, which solves Adam's parameter overfitting problem well by introducing itself when updating the parameters. And we use cosine annealing algorithm to adjust the learning rate during training.

5. Performance Evaluation

The following experiments will be carried out: first, the encoder performance before and after the improvement will be compared; then the classification performance of the standard architecture will be compared with other baseline methods.

We use a kernel size of 65, an encoder layer number of 4, and an encoder hidden size of 32 as a standard model for the TMRN-GLU proposed in this paper, which takes into account the complexity of the model and the classification performance. In order to avoid the influence of errors, all experiments in this section use K-fold cross-validation. It means to perform K times of training, save the best weight each time, conduct relevant experiments on the K times of training network on an independent test set, and take the average of the K times of results as the final result.

All models were trained on the RML2016.10b dataset until convergence, the relevant situation of this dataset and how to divide it have been introduced above. Model training was stopped when there was no improvement in results on the test set within 10 consecutive epochs. We used a Pytorch-based deep learning platform running on a Linux machine configured a I7 7800X with six cores, 32 g RAM, and two Nvidia Titan Xp GPUs.

5.1. Effectiveness of Encoder Improvements

This subsection will verify the effectiveness of the encoder improvement using this standard architecture as the base model. The only difference between the two methods compared is the use of the original encoder or the improved encoder using GLU.

The average classification accuracy of the two schemes for the training set in each Epoch at training time is first compared. The result is shown in Figure 8. As shown it, the curve of the Encoder with GLU is always above the image. In the 5th epoch, the classification accuracy of the encoder with GLU on the verification set reaches the effect of the original encoder after 10th epoch. It shows that the improved encoder has faster convergence speed and stronger learning ability of information.

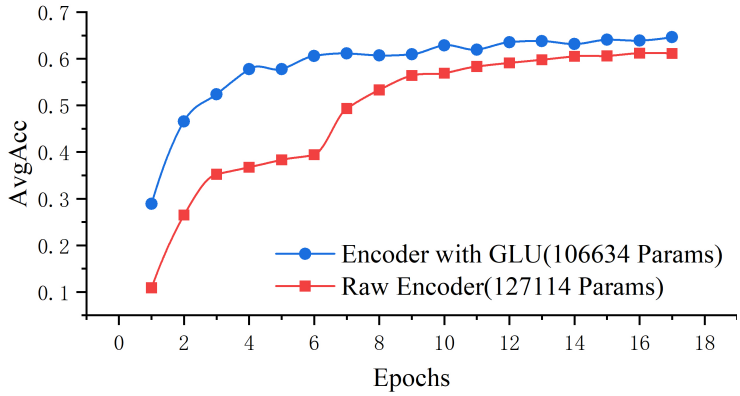

Figure 8. Average accuracy of the improved Encoder vs. the original encoder on the validation set for each epoch in the training process.

The Figure 9 and Table 3 show the recognition accuracy and related performance evaluation metrics of the two codes at each signal-to-noise ratio, respectively. The indicators in the Table 3 are defined as follows: The overall rate is the ratio of the number of pairs predicted by the model to the overall amount of predicted data; the average accuracy is the mean of the classification accuracy of each modulation category; the Kappa coefficient is a quantitative expression of the confusion matrix, which reflects the consistency of the

network in recognizing different kinds of modulations, and the larger the value, the better the consistency. kappa is defined as in Equation (9).

$$\text{Kappa} = \frac{\text{OA} - \frac{\sum_{i=1}^{N_{\text{class}}} C_i \times N_i}{N \times N}}{1 - \frac{\sum_{i=1}^{N_{\text{class}}} C_i \times N_i}{N \times N}} \tag{9}$$

N is the total sample size, C_i is the true sample size of a category, and N_i is the predicted sample size of a category.

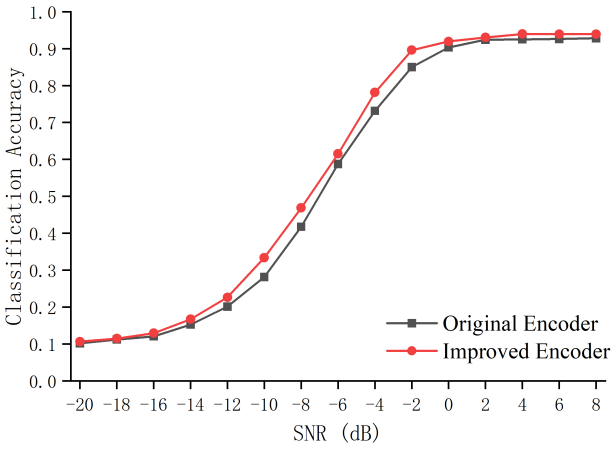

Figure 9. Classification accuracy of two types of encoders vs. SNRs.

Table 3. Comparison of Evaluation Metrics of original Encoder and improved Encoder.

Evaluation Metrics	Original Encoder	Improved Encoder
Model Parameters	127,114	106,634
Overall accuracy	63.14%	65.33%
Average Accuracy	63.93%	66.05%
Maximum Accuracy	92.61%	94.22%
Kappa	0.6267	0.6517

It can be seen that the curve representing the improved encoder is always at the top of the graph, indicating that the improved encoder has higher modulation recognition accuracy for signals at each SNR than the original encoder. Combined with the analysis of the data in the table, it can be concluded that for the improved encoder using GLU, the amount of parameters is reduced by 16% and the average accuracy is increased by more than 2%. the use of GLU maintains and even improves the classification accuracy of the model while reducing the complexity of the model. The above can prove that the improvement of the encoder structure is effective.

5.2. Comparison of Classification Accuracy with Baseline

In our experiments we selected the following typical deep learning-based AMR algorithms as baselines, which contain various types of mainstream frameworks for deep learning and each has its own advantages. the CNN model in [20] as the first use of deep learning in the AMR domain, to which we added two new convolutional layers, making its recognition accuracy to be improved over the original model. We also use a modified version of CNN, DensesNet, as one of the baseline, which has a similar structure to CNN network, but with shortcut links in the non-directly connected two layers. Due to the ability to extract past implicit information, RNNs prove to be effective when dealing with time-domain information, and we use a CLDNN architecture similar to that in [34], which inserts a layer of LSTM in the CNN. next, we go a step further and refer to algorithms

that mainly use LSTM as the main structure of the network ScLSTM [24], which improves on [35], and its structure presents a visually opposite pattern to the CLDNN, reducing the number of parameters in the LSTM network by two layers of CNN with one layer of MaxPool. ResNet is first associated with [36] and used in [21] in the AMR domain. We use a kernel size of 65, an encoder layer number of 4, and an encoder hidden size of 32 as a standard model for the TMRN-GLU proposed in this paper, which takes into account the complexity of the model and the classification performance. We also provide a TMRN-GLU-Small with a hidden size of 8 as a reference comparison, which has a much smaller number of parameters compared with the standard model. And the relationship between performance of TMRN-GLU and its structural parameters will be discussed in the next sub-section.

Figure 10 shows the performance of the model proposed in the article compared to the baseline. It can be intuitively perceived from the figure that for all the displayed AMR algorithms their classification accuracy increases with the increasing SNR. Our algorithm starts from a SNR of −10 dB and its recognition accuracy is better than the baseline. Compared with our baselin reproduced and trained on the same platform using the same dataset and batchsize conditions, its average recognition accuracy is 65.7% on all SNRs on average, and the highest recognition accuracy is 93.53% at high SNRs. It has a clear advantage over ResNet, the best result in baseline, which has an average recognition accuracy of 62.15% at all SNRs and a maximum recognition accuracy of 91.43%.

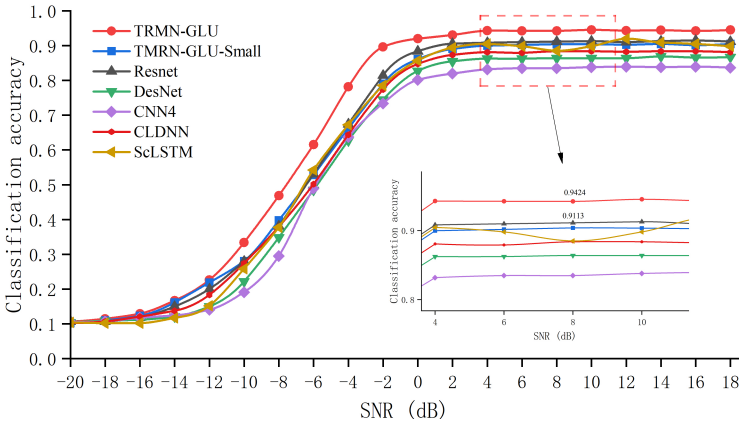

Figure 10. Classification accuracy of DL models under different SNR.

The Table 4 shows the statistical performance evaluation metrics of the proposed algorithm and other networks. From the analysis of the data in the table, the proposed TMRN-GLU algorithm significantly outperforms other algorithms in terms of classification accuracy. For example, ResNet is a representative CNN algorithm, and TMRN-GLU improves the average accuracy over it by more than 4% with the obvious advantage of number of parameters and operation speed, which is a significant improvement in the field of deep learning. While ScLstm is based on LSTM by using CNN to downsample the input in length to shorten the operation speed, TMRN-GLU has a 6.7% higher average classification accuracy than ScLstm with a similar operation speed.

Table 4. Performance evaluation metrics for AMR algorithms.

Evaluation Metrics	TMRN-GLU	TMRN-GLU-Small	ResNet	DenseNet	ScLstm	CLDNN
Model Parameters	107 k	25 k	171 k	286 k	398 k	306k
Operation Time	1.14 ms	1.03 ms	15.58 ms	17.35 ms	0.83 ms	13.43 ms
Overall accuracy	65.33%	61.72%	61.31%	57.10%	58.67%	57.94%
Average Accuracy	66.05%	62.03%	62.22%	58.10%	60.90%	60.03%

The Figure 11 shows the relationship between the number of parameters for each of the above AMR models and the average accuracy over the entire test set. The number of parameters of a model determines its resource consumption. It also reflects to some extent the inference speed and the convergence speed during training. For example, an RNN network with the same number of parameters is much slower than a CNN, because an RNN needs the previous moment of inference to be completed before the current unit of inference can be performed, while a CNN can perform multiple convolutional kernels at the same time thanks to the development of hardware. The average accuracy reflects the overall recognition performance of the model on the test set, and a better average accuracy means that the model is better at the classification task. In general, the closer to the top left of the image the better the overall performance of the model. As shown in the figure, the average accuracy of the standard structure of TMRN-GLU is 66.05%, and its average accuracy is about 4% better than the best baseline, while the number of parameters is only 107 k, which is closer to the upper left corner of the image than all comparison methods. the recognition accuracy of TMRN-GLU-Small is similar to that of ResNet, but its parameters are only 14% of ResNet, which indicates that this method is more expressive than the ResNet model. The self-attentive mechanism has the advantages of better contextual information induction similar to RNN and CNN integration of local features, and has good computational parallelism, and the inference speed is at the same level as CNN.

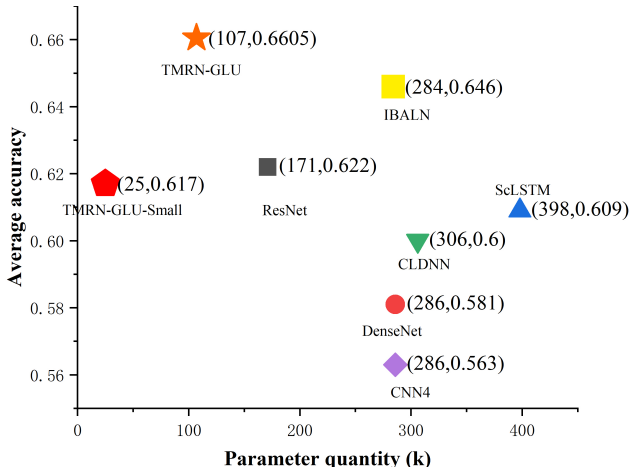

Figure 11. Average classification accuracy at all SNRs vs. Parameter quantity.

To better understand the performance of the TMRN-GLU, the confusion matrix of the TMRN-GLU at various SNR is showed at Figure 12, and the classification accuracy of all types of modulation with signal-to-noise ratio is shown in Figure 13.

As shown in Figure 12, at higher signal-to-noise ratios, as in Figure 12d, the confusion matrix diagonal is clearer as the SNR increases, indicating that the classification accuracy increases as the signal-to-noise ratio increases, despite considerable difficulty in separating the AM-DSB and WBFM classes. The difficulty in identifying AM-DSB and WBFM is mainly due to the fact that AM-DSB and WBFM modulated signals are generated from real audio sources with a silent period when the dataset is produced. It is difficult to distinguish AM and FM based signals at the same carrier frequency during the silent period [20]. At −6 dB SNR, as in Figure 12b, there is a degree of confusion between QAM16 and QAM64, as the former is a subset of the latter, which can be easily misclassified due to noise at lower SNRs, as demonstrated in the signal constellation diagram in the previous section. However, it can be clearly seen that the classification of QAM is more accurate under high SNR conditions, and TMRN-GLU eliminates this confusion very well. At a lower −12 dB SNR in Figure 12a, all signal confusion is more severe. Combined with the variation of classification accuracy with SNRs given in Figure 10.

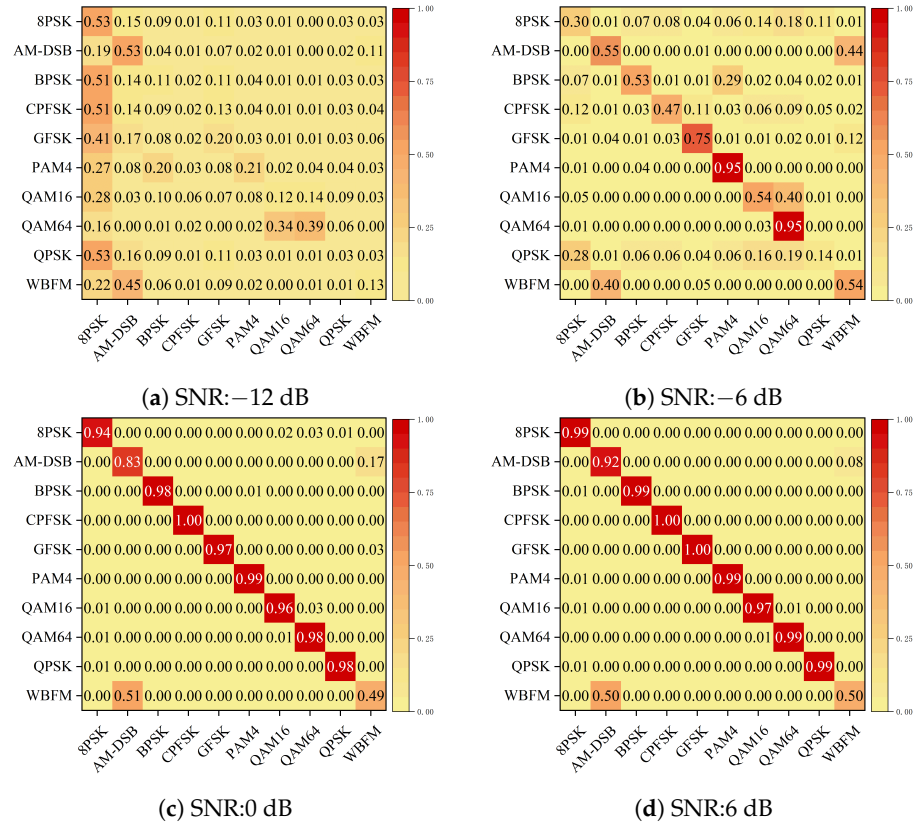

Figure 12. Confusion matrix for TMRN-GLU. The (i, j) denotes the probability of being classified as a test instance of the modulation type. (**a**) is under −12 dB SNR; (**b**) is under −6 dB SNR; (**c**) is under 0 dB SNR; (**d**) is under 6 dB.

And the classification accuracy of all types of modulation with signal-to-noise ratio is shown in Figure 13. Under high SNR conditions, only WBFM and AM-DSB were not classified with high accuracy, and they were confused with each other. In general, the accuracy of digital modulation classification basically increases with SNR.

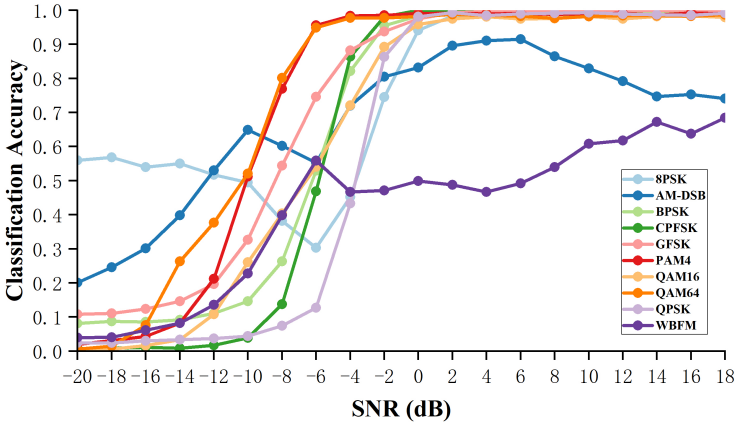

Figure 13. Classification accuracy of different kinds of modulation patterns vs. SNRs.

6. Discussion of Factors Affecting Network Performance

6.1. Discussion of the Impact of Hyperparameters on TMRN-GLU Performance

The relationship between the classification performance of the TMRN-GLU and the parameters of the components that make up the TMRN is discussed in this section. As mentioned above, the TMRN-GLU consists of three main components, namely the convolutional layer that extracts the sequence features, the tail judgment network that is responsible

for integrating the information and giving the classification probability, and the most important Encoder network which is in the middle of the first two. We will analyze the effect of the kernel size of the convolutional layer, the dimension of the encoder, i.e., the hidden size, and the number of encoder layers on the classification performance of the network and on the size of the model. Since the judgment network is simply the connection of two fully connected layers, we will not analyze it in this artical. The basic TMRN-GLU architecture mentioned previously is used as a basis for the analysis of a single variable, while the other network structures are kept constant to obtain the direct effect of this parameter on the performance.

The kernel size of the convolutional layer which extracts sequence features, mainly affects the range covered by each kernel. The Figure 14 shows the effect of the kernel size of the convolutional layer on the classification performance of the network. From the results, we can analyze that when the kernel size increases, the classification accuracy increases in the range of SNR values from −8 to 2. However, it is clear that this increase is not significant and the kernel size does not have a significant gain on the classification accuracy.

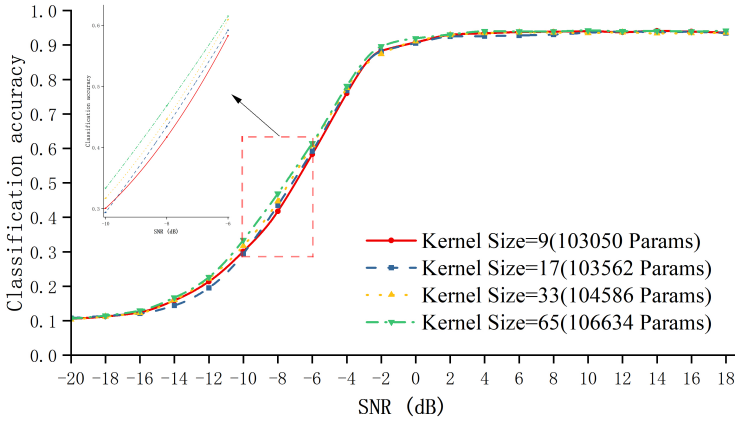

Figure 14. Classification accuracy vs. SNRs for TMRN-GLU with various number of Kernel Size.

Encoder, as the most important component of the network proposed in this article, will be analyzed in the next step for the impact of its different parameters on the performance of Transformer-based network in modulation classification task. The Figure 15 demonstrates the effect of the number of encoder layers on the classification accuracy.

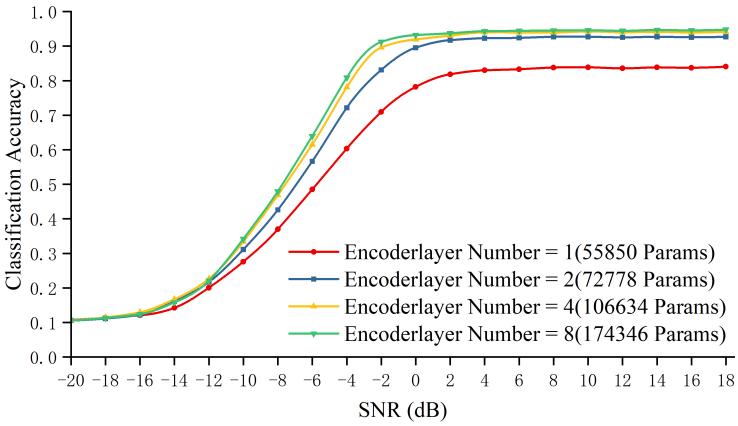

Figure 15. Classification accuracy vs. SNRs for TMRN-GLU with different number of Encoder layers.

The number of encoder layers and the number of parameters of the model are essentially linear, but the performance improvement does not leap with the multiplication of the parameters. When the number of encoder layers is increased from one to two, the classification accuracy is greatly improved by nearly 10%; when the number of encoder layers is

increased from two to four, the classification accuracy of the model is also improved from −10 dB to 2 dB to some extent, but when the number of encoder layers is increased from four to eight, the model depth is multiplied but the accuracy improvement is very limited. Of course, this is related to the fact that the accuracy of the model is already relatively high at the 4-layer encoder, and the room for improvement is not much. The authors of the article [37] find that the attention maps of the models in the deep Transformer network are very similar or even turn out to be the same at deeper layers. This suggests that as the Transformer network goes deeper, self-attention becomes less effective in generating different attention to capture rich representations. The authors call this attention collapse.

The effect of hidden size of single encoder on classification accuracy is similar to that of the number of encoder layers. The Figure 16 shows the performance for the different hidden sizes, when the hidden size increases from 8 to 16 there is a significant improvement in performance, from 16 to 32 there is also a slight improvement but mainly in the range of −4 dB to 0 dB. When the hidden size is increased from 32 to 64, there is no improvement in accuracy in some positions.

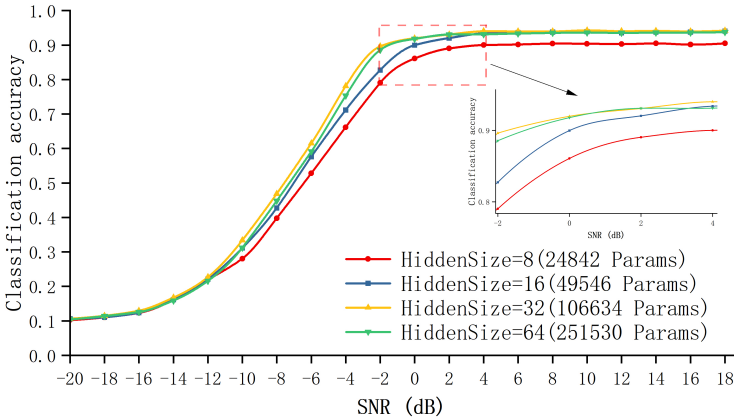

Figure 16. Classification accuracy vs. SNRs for TMRN-GLU with different Encoder hidden size.

The results of the experiments in this section are summarized in the following table, Table 5. From this table, we can see that the classification accuracy is in line with the marginal utility as the parameters increase linearly with a constant amount of data, so a variety of factors such as the complexity of the model and the gain it brings to the performance should be considered when selecting the actual model.

Table 5. Comparison of different network hyperparameters.

Hyperparameters	TotNum of Parameters	MaxAcc (%)	AvgAcc (%)
$k = 9$	103,050	94.09	64.49
$k = 17$	103,562	93.91	64.42
$k = 33$	104,586	93.69	64.79
$k = 65$	106,634	94.22	66.05
$N = 1$	55,850	84.04	57.09
$N = 2$	72,778	92.70	63.95
$N = 4$	106,634	94.22	66.05
$N = 8$	174,346	94.74	66.29
$n_c = 8$	24,842	90.47	61.72
$n_c = 16$	49,546	94.03	64.45
$n_c = 32$	106,634	94.22	66.05
$n_c = 64$	251,530	93.70	65.03

6.2. The Difference between Using Different Orders of Magnitude of Data Sets

In this experiment, we used the standard model of TMRN-GLU to validate the effect of different orders of magnitude of data sets on the results. rml2016.10A is from DeepSig like RML2016.10B, which is mainly used in the previous section. Their main difference is the different amount of data. The amounts of data for both is shown in the following Table 6.

Table 6. Datasets comparison.

Parameter Name	RML2016.10A	RML2016.10B
modulation types	8PSK, AM-SSB, AM-DSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, WBFM	8PSK, AM-DSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, WBFM
SNR range	−20:2:18	−20:2:18
Amount of data per subgroup	6000	1000
Total data volume	1,200,000	220,000
Number of train sets	720,000	132,000
Number of test sets	480,000	88,000

We used the same TMRN-GLU standard model with the batchsize set to 128 and the initial learning rate set to 0.002, and used the cosine annealing algorithm to train to the position of no more convergence, and compared the classification accuracy at each signal-to-noise ratio using different data volume datasets, and the results are shown in Figure 17. As presented in the figure, the performance of the network trained using RML2016.10B as the dataset is much better than that of the network trained using RML2016.10A. It can be seen that the Transformer-based network architecture is affected by the volume of data; the more data, the better the performance.

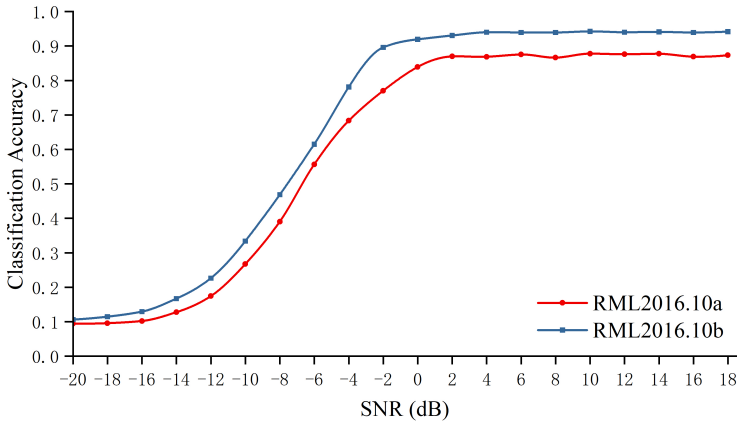

Figure 17. Classification accuracy of different datasets vs. SNRs.

7. Conclusions

We propose a Transformer-based classification network improved by GLU on the AMR task called TMRN-GLU. After experimental validation, the proposed TMRN-GLU standard structure achieves a good balance betweem model complexity and classification performance. Meanwhile, TMRN-GLU achieves better results than the reference algorithm on the same dataset with a smaller total number of model parameters. Finally, it was found that the present algorithm works significantly better on large data sets than on small data sets.

There are some limitations of the work in this paper. It can be seen that the complexity of the self-attentive mechanism is $O(n^2)$ for the sequence length at the time of computation. How to make it linear in complexity for longer communication signals by optimizing the internal structure is an issue to be explored in future research. In addition, we found that the performance of this method is affected by the collective amount of data and deteriorates

as the amount of data decreases. How to achieve a satisfactory recognition accuracy with a small sample size is another direction to be investigated in the future.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z. and Y.M.; software, Y.Z.; validation, Y.Z.; writing original draft preparation, Y.Z.; writing—review and editing, Y.M., Y.Z. and C.T.; project administration, Y.M. All authors read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61671318.

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

References

1. Gao, Y.; Qin, Z.; Feng, Z.; Zhang, Q.; Holland, O.; Dohler, M. Scalable and reliable IoT enabled by dynamic spectrum management for M2M in LTE-A. *IEEE Internet Things J.* **2016**, *3*, 1135–1145. [CrossRef]
2. Haykin, S. Cognitive radio: Brain-empowered wireless communications. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 201–220. [CrossRef]
3. Wang, Z.; Sun, D.; Gong, K.; Wang, W.; Sun, P. A Lightweight CNN Architecture for Automatic Modulation Classification. *Electronics* **2021**, *10*, 2679. [CrossRef]
4. Zhu, Z.; Nandi, A.K. *Automatic Modulation Classification: Principles, Algorithms and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
5. Gou, Y.; Lei, Y.; Liu, L.; Dai, Y.; Shen, C. Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems. *arXiv* **2020**, arXiv:2010.05740.
6. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9759–9768.
7. Luo, G.; Zhou, Y.; Sun, X.; Cao, L.; Wu, C.; Deng, C.; Ji, R. Multi-task collaborative network for joint referring expression comprehension and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10034–10043.
8. An, F.P.; Liu, Z.W. Medical image segmentation algorithm based on feedback mechanism CNN. *Contrast Media Mol. Imaging* **2019**, *2019*. [CrossRef] [PubMed]
9. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control* **2019**, *47*, 312–323.
10. Wang, J.; Xue, M.; Culhane, R.; Diao, E.; Ding, J.; Tarokh, V. Speech emotion recognition with dual-sequence LSTM architecture. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6474–6478.
11. Wang, N.; Liu, Y.; Ma, L.; Yang, Y.; Wang, H. Multidimensional CNN-LSTM Network for Automatic Modulation Classification. *Electronics* **2021**, *10*, 1649. [CrossRef]
12. Chugg, K.M.; Long, C.S.; Polydoros, A. Combined likelihood power estimation and multiple hypothesis modulation classification. In Proceedings of the Conference Record of the Twenty-Ninth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 30 October–1 November 1995; Volume 2, pp. 1137–1141.
13. Wei, W.; Mendel, J.M. Maximum-likelihood classification for digital amplitude-phase modulations. *IEEE Trans. Commun.* **2000**, *48*, 189–193. [CrossRef]
14. Huang, S.; Yao, Y.; Wei, Z.; Feng, Z.; Zhang, P. Automatic modulation classification of overlapped sources using multiple cumulants. *IEEE Trans. Veh. Technol.* **2016**, *66*, 6089–6101. [CrossRef]
15. Tian, J.; Pei, Y.; Huang, Y.D.; Liang, Y.C. Modulation-constrained clustering approach to blind modulation classification for MIMO systems. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 894–907. [CrossRef]
16. Azzouz, E.E.; Nandi, A.K. Automatic identification of digital modulation types. *Signal Process.* **1995**, *47*, 55–69. [CrossRef]
17. Lin, X.; Eldemerdash, Y.A.; Dobre, O.A.; Zhang, S.; Li, C. Modulation classification using received signal's amplitude distribution for coherent receivers. *IEEE Photonics Technol. Lett.* **2017**, *29*, 1872–1875. [CrossRef]
18. Xie, W.; Hu, S.; Yu, C.; Zhu, P.; Peng, X.; Ouyang, J. Deep learning in digital modulation recognition using high order cumulants. *IEEE Access* **2019**, *7*, 63760–63766. [CrossRef]
19. Swami, A.; Sadler, B.M. Hierarchical digital modulation classification using cumulants. *IEEE Trans. Commun.* **2000**, *48*, 416–429. [CrossRef]
20. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In *International Conference on Engineering Applications of Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 213–226.
21. O'Shea, T.J.; Roy, T.; Clancy, T.C. Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [CrossRef]
22. O'shea, T.J.; West, N. Radio machine learning dataset generation with gnu radio. In Proceedings of the GNU Radio Conference, Boulder, CO, USA, 12–16 September 2016; Volume 1.

23. Hong, D.; Zhang, Z.; Xu, X. Automatic modulation classification using recurrent neural networks. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 695–700.
24. 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]
25. Wei, S.; Qu, Q.; Zeng, X.; Liang, J.; Shi, J.; Zhang, X. Self-attention bi-lstm networks for radar signal modulation recognition. *IEEE Trans. Microw. Theory Tech.* **2021**, *69*, 5160–5172. [CrossRef]
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
27. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–11 December 2014; Volume 27.
28. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
29. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In Proceedings of the CIKM'19 International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1441–1450.
30. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.H.; Tay, F.E.; Feng, J.; Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 558–567.
31. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, NSW, Australia, 6–11 August 2017; pp. 933–941.
32. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. Cvt: Introducing convolutions to vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 22–31.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
34. Sainath, T.N.; Vinyals, O.; Senior, A.; Sak, H. Convolutional, long short-term memory, fully connected deep neural networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 4580–4584.
35. 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]
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
37. Zhou, D.; Kang, B.; Jin, X.; Yang, L.; Lian, X.; Jiang, Z.; Hou, Q.; Feng, J. Deepvit: Towards deeper vision transformer. *arXiv* **2021**, arXiv:2103.11886.