Multi-Task Chinese Speech Recognition Method Based on the Squeezeformer Model

Ying Guo,Li Wang

***Abstract*—End-to-end training has emerged as a prominent trend in speech recognition, with Conformer models effectively integrating Transformer and CNN architectures. However, their complexity and high computational cost pose deployment challenges. To address these issues, we propose a multi-task Chinese speech recognition method based on the Squeezeformer model. We replace the FMCF structure in Conformer with an MF/CF structure, leveraging the convolutional module as a local Multi-Head Attention (MHA) module to enhance efficiency. Multi-level down-sampling and up-sampling using a time-series U-Net further reduce computational costs. By eliminating redundant LayerNorm layers and employing depthwise separable convolutions, we streamline the model, reduce parameters, and lower deployment costs. An Adaptor Layer is integrated into the MHSA module to mitigate the vanishing gradient problem, and a ScaleVar Layer is added to enhance flexibility. Additionally, the RealFormer module is introduced on the decoding side to improve context understanding. Combining Connectionist Temporal Classification (CTC) with attention-based encoding and decoding models for multi-task learning improves performance and accuracy. Experimental results show that the proposed method reduces the parameters on AISHELL-1 dataset by 16% and reduces the character error rate to 5.50%. At the same time, it also shows good performance on AISHELL-2 dataset.**

***Index Terms*—End-to-end, Automatic Speech Recognition, Multi-task, Squeezeformer.**

I. Introduction

THE components of traditional speech recognition systems typically include acoustic models, pronunciation dictionaries, and language models. In contrast, end-to-end speech recognition adopts a more streamlined approach, learning directly from raw audio data to generate corresponding text without the need for additional feature extraction or alignment processes[1]. This simplified architecture makes end-to-end speech recognition more efficient, flexible, and adaptable to various speech scenarios and languages[2]. The end-to-end Conformer model, which combines the strengths of Convolutional Neural Networks (CNNs) and Transformers, has achieved superior performance in speech recognition tasks[3], [4]. However, it still has limitations and room for improvement. For instance, the Conformer architecture uses larger convolutional kernels to better integrate global information, whereas smaller convolutional kernels are more effective for local processing capabilities of attention mechanisms[5]. Therefore, placing multi-head attention and convolutional modules back-to-back (referred to as the MC structure) is not optimal[6], [7]. Moreover, the Conformer architecture is more complex than the Transformer architecture commonly used in natural language processing. It incorporates various normalization schemes, activation functions, and Macaron-like structures, contributing to its complexity. This complexity poses challenges for effective model deployment on dedicated hardware platforms. Hence, it is crucial to improve the Conformer architecture to enhance model efficiency, simplify the structure, and facilitate deployment on specialized hardware. Achieving these improvements would significantly advance the practical application of end-to-end speech recognition technology.

In this paper, we propose a multi-task Chinese speech recognition method based on the Squeezeformer model, an optimized version of Conformer. Squeezeformer replaces the FMCF (forward + multi-head attention + convolution + forward) structure of the Conformer with a Transformer-style MF/CF (multi-head attention + forward/convolution + forward) structure. This approach treats the convolutional module as a local MHA module to fully leverage its advantages in local modeling. By incorporating a time-series U-Net that downsamples four times, then downsamples two times, and finally upsamples two times, the computational cost of the multi-head attention module on long sequences is significantly reduced. The activation function in the convolutional module is unified to Swish, simplifying the model structure and reducing deployment costs. Additionally, redundant LayerNorm is replaced with learnable LayerNorm after the reduction layer and module, facilitating the reduction and activation of output values while also reducing model parameters. Depth-separable convolution is utilized to more efficiently subsample the input signal when downsampling by a factor of two. When the Squeezeformer layer is compressed, an Adaptor layer is added to the MHSA module to prevent gradient vanishing, resulting in finer replication. The convolutional module also incorporates a ScaleVar layer to scale and bias the input, enhancing the model's flexibility and expressiveness. Furthermore, RealFormer is added to the encoder, offering better context understanding and higher quality generated results[8], [9]. For multi-task learning, we combine Connectionist Temporal Classification (CTC) with an attention-based encoding and decoding model[10], [11]. The forward-backward algorithm of CTC aligns the output sequence with the input sequence in temporal order, while the attention model's alignment is not sequence-bound and is data-driven, which can be challenging to train[12], [13], [14]. By combining CTC and attention models, we can harness the strengths of both approaches to enhance model performance. The attention mechanism helps the model focus on crucial information in the speech signal, improving the accuracy of speech recognition by highlighting important features. Meanwhile, CTC enhances the model's ability to

Manuscript received June 2, 2024; revised November 6, 2024.This work was supported by the Special Fund for Scientific Research Construction of University of Science and Technology Liaoning.

Ying Guo is a postgraduate student at School of Computer Science and software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (e-mail: 3301983966@qq.com).

Li Wang is a professor of the College of Computer Science and Technology Liaoning, Anshan 114051, China. (Corresponding author, e-mail: wangli9966@ustl.edu.cn).

learn robust frame-level alignment of speech, addressing the label alignment issue and improving overall recognition performance[15], [16].

This paper is organized as follows: Section 2 reviews related work; Section 3 describes the multi-task Squeezeformer model for Chinese speech recognition; Section 4 presents the dataset and analyzes the experimental results; and Section 5 summarizes the findings and conclusions.

II. Related Work

A. End-to-end Speech Recognition

End-to-end speech recognition models directly map acoustic signals to tag sequences, eliminating the cumbersome processes found in traditional methods, such as extracting acoustic features from speech audio, conducting acoustic modeling, language modeling, and performing searches based on Bayesian decision rules. This streamlined architecture simplifies building and training, leading to revolutionary advancements in speech recognition technology. A typical end-to-end speech recognition model comprises three main components: an encoder, an aligner, and a decoder. The encoder converts the input speech sequence into a feature sequence. The aligner ensures the alignment between the feature sequence and the language, while the decoder interprets the final recognition result. This integrated approach enables end-to-end speech recognition systems to more effectively capture speech features, resulting in more accurate and efficient recognition processes[17], [18].

1) The CTC-based end-to-end model effectively addresses the hard alignment problem in speech recognition. In the end-to-end LVCSR (Large Vocabulary Continuous Speech Recognition) model, CTC (Connectionist Temporal Classification) overcomes data alignment challenges by directly outputting the target transcription, thereby eliminating the need for manual alignment of input and output sequences. This approach modifies the network structure by adding an extra output node to represent an additional class, thus preserving the acoustic model's structure and only fine-tuning the output layer to meet new classification requirements. The core component of CTC is its loss function, known as the CTC loss function, which involves two stages: path probability calculation and path aggregation. CTC facilitates the inference of the target sequence by introducing a blank label and intermediate conceptual paths.

A significant advantage of CTC is that it eliminates the need to pre-determine the exact correspondence between acoustic features and text labels, allowing the model to automatically learn how to decode the input sequence and infer the most likely text output. Li et al. designed a novel ASR (Automatic Speech Recognition) approach using bidirectional long short-term memory recurrent neural networks (LSTM-RNNs) combined with connectionist temporal classification. This method directly transcribes graphemes, producing results highly competitive with phoneme transcription. Their findings indicate that increasing network depth and the number of hidden units effectively improves recognition performance. In their experiment, they designed a network with three layers: a 78-dimensional feedforward layer, an LSTM layer with 120 memory units, and another LSTM layer with 27 memory units. Song et al. implemented a system using two variants of phoneme recognition neural networks, combining CNNs (Convolutional Neural Networks) and RNNs for ASR, utilizing four CNN layers in the model structure. Inspired by the advantages of CNN and CTC methods, Zhang et al. proposed a model that combines hierarchical CNNs with CTC directly, eliminating the need for recurrent connections. This model employs ten CNN layers and three fully connected layers, concluding that model depth is proportional to recognition accuracy. Research also focuses on the depth of CTC networks. Amodei et al. designed a 9-layer network model with 7 recurrent layers, achieving accuracy that can surpass human performance in some tasks. Similarly, Zweig et al. trained a network model consisting of a nine-layer bidirectional LSTM RNN, obtaining optimal results across different datasets. In summary, while increasing the structure and depth of network models can enhance recognition accuracy, it does not necessarily mean that deeper, more complex networks will achieve better results in all situations.

2) An end-to-end model based on attention effectively handles long-distance dependencies by introducing an attention mechanism at the decoder. This allows the model to dynamically focus on different parts of the input sequence while generating the output sequence. The structure consists of three modules: the encoding network, decoding network, and attention subnetwork.Both the encoding and decoding networks utilize recurrent neural network (RNN) units. The encoding network transforms input sequences into hidden representations, typically composed of multiple layers of RNNs (such as LSTM or GRU) and other layers (such as convolutional neural networks or self-attention mechanisms) to extract more abstract feature representations. The attention subnetwork is a multi-layer perceptron with a single hidden layer that receives the output of the encoding network and the hidden state of the decoding network, calculates the correlation score, and represents the relationship between the two.The decoding network consists of a single-layer RNN and a Maxout network. The output of the encoding network is weighted and summed by the attention coefficients from the attention subnetwork to generate the target vector. This target vector is used to calculate the posterior probability of each phoneme in the output sequence, enabling the decoder to flexibly capture the semantic information of the input sequence[19].

With the continuous pursuit of model performance, encoders in attention-based models have gradually evolved into more complex structures to enhance their encoding capabilities. Initially, encoders consisted of three layers, but they have progressively developed into four, five, and six layers. By incorporating technologies such as network-in-network, batch normalization, residual networks, and convolutional LSTM, encoder networks can now reach depths of up to 15 layers. On the Wall Street Journal dataset, this deep encoder network achieved a word error rate (WER) of 10.53 percent, demonstrating excellent performance without the use of dictionaries or language models.

B. Multi-tasking Speech Recognition

Multi-task learning is a machine learning approach that aims to simultaneously learn multiple related tasks by sharing knowledge, ultimately improving overall performance.

In this approach, multiple tasks leverage the same model and achieve parameter sharing, which allows for efficient transmission and propagation of learned parameters. This technique can reduce training time, enhance model accuracy, and improve generalization to new tasks. For instance, in speech recognition, several tasks can share the same speech feature extraction model, facilitating the transfer of knowledge across tasks. This parameter sharing enhances the performance of the speech recognition model by enabling effective knowledge transfer.

The connection between multi-task learning and speech recognition is primarily reflected in the following aspects:

1) Hierarchical multi-task learning leverages the correlations between different speech recognition tasks, such as voice command and voice translation. This approach enhances model performance by integrating subtasks of varying difficulty or types into a unified network, enabling them to share underlying features[20].

2) End-to-end multi-task learning facilitates knowledge sharing across different speech recognition tasks, such as speech feature extraction and various network layers. This sharing of knowledge improves model generalization and reduces training time.

3) Generalized multi-task learning promotes task generalization across various speech recognition tasks, such as extending knowledge from a speech command task to a speech translation task. This generalization enhances model performance and enables the model to adapt to a broader range of application scenarios[21].

III. Methods

A. Multi-task Chinese Squeezeformer Speech Recognition Model

In this paper, we propose a multi-task Chinese Squeezeformer speech recognition model, as shown in Figure 1. The model is primarily composed of three main components. First, acoustic features are downsampled for dimensionality reduction through a depthwise separable convolutional module. Then, the reduced-dimensional acoustic features are transformed into hidden layer features by stacking 12 Squeezeformer modules. Finally, LAS and CTC are combined to optimize the entire model, with Squeezeformer being further optimized based on the Conformer model. The Transformer-style MF/CF (multi-head attention + forward/convolution + forward) structure replaces the FMCF (forward + multi-head attention + convolution + forward) structure in the Conformer, treating the convolutional module as a local MHA module to fully leverage its advantages in local modeling. By applying a time-series U-Net that downsamples four times, then downsamples two more times, and finally upsamples twice at the end of the model, the computational cost of the multi-head attention module on long sequences is significantly reduced. The activation function in the convolutional module is unified to Swish, simplifying the model structure and reducing deployment costs. Additionally, redundant LayerNorm is replaced with learnable LayerNorm after the reduction layer and module, facilitating the reduction and activation of output values while also reducing model parameters. Depth-separable convolution is employed to subsample the input signal more

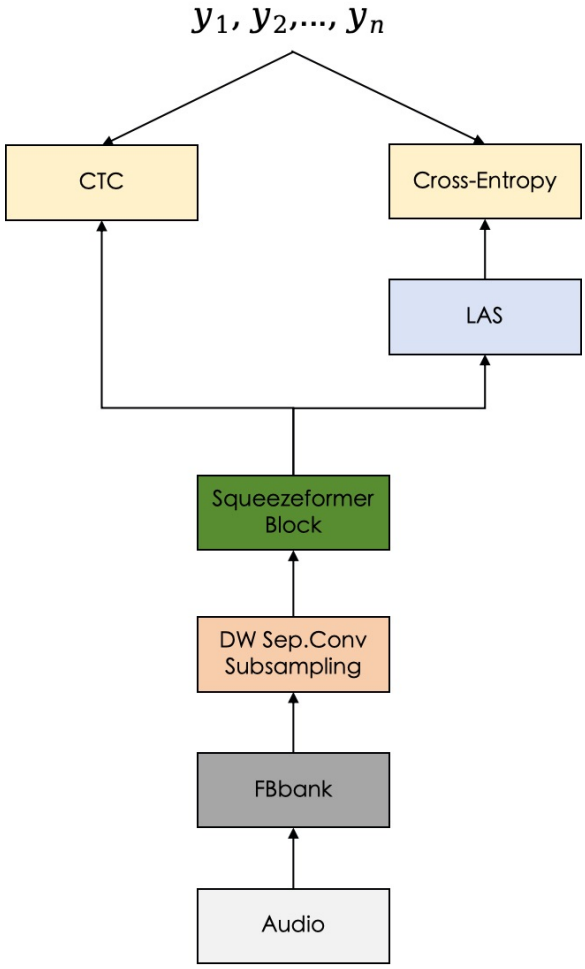

Fig. 1. Speech recognition model of multi-task Chinese Squeezeformer model

efficiently when downsampling by a factor of 2. When the Squeezeformer layer is compressed, an Adaptor layer is added to the MHSA module to prevent gradient vanishing during data restoration, resulting in finer replication. The convolutional module also incorporates a ScaleVar layer to scale and bias the input, enhancing the model's flexibility and expressiveness. Furthermore, RealFormer is added to the encoder, offering improved context understanding and higher-quality generated results.

B. Squeezeformer Block

The Squeezeformer block is composed of a Multi-head Self-Attention module, a Feed Forward module, and a Convolution module, with residual connections applied to each module. The Macaron structure is replaced by an MF/CF structure (self-attention + FFN + convolution module + FFN), similar to the Transformer network, where the convolutional module functions as a local MHA module to fully leverage its advantages in local modeling. Combined with the time-series U-Net structure, which includes 4x downsampling, followed by 2x downsampling, and 2x upsampling at the end of the model, the computational cost of the multi-head attention module on long sequences is reduced.Squeezeformer maintains a 4x downsampling rate, similar to Conformer, up to the 7th block, after which it

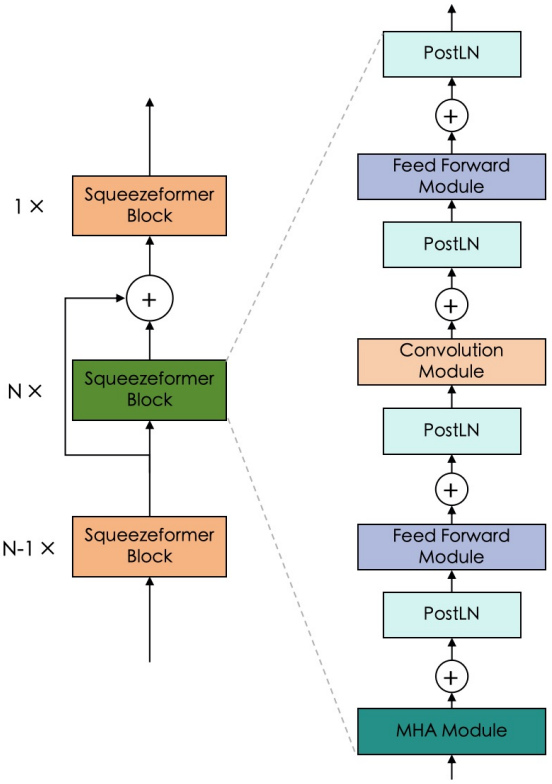

Fig. 2. Squeezeformer Block

undergoes additional 2x downsampling through a pooling layer. The pooling layer uses depth-separable convolution with a stride of 2 and a kernel size of 3 to merge adjacent embedding vectors. However, 8x downsampling in the time dimension can lead to unstable training. This instability may arise because the decoder is responsible for mapping the embedded vectors corresponding to speech frames to the modeling units, requiring sufficient resolution to decode the entire sequence. After 8x downsampling, the decoder may lack the necessary resolution. Inspired by U-Net, Squeezeformer restores resolution at the end of the network through an upsampling layer. The upsampling module takes the 4x and deeper 8x subsampled embedded vectors, combines them through skip connections, and outputs the final 4x subsampled embedded vectors.

Conformer uses the Swish activation function in most modules but employs the Gated ar Unit (GLU) as the activation function in the Convolution module. To simplify the model structLineure and reduce deployment costs, Squeezeformer uses the Swish activation function throughout the entire model. Additionally, layer normalization is simplified by replacing the previous Layer Norm (preLN) with a learnable deflate layer, which both deflates and activates the output values.

$$Scaling\,(x) = \gamma(x) + \beta \tag{1}$$

Where,γ and β are learnable parameters with the same dimension as the input value x. Squeezeformer applies Layer Norm (postLN) after the module and learns to shrink the Layer Norm (preLN) before the replacement module. This approach ensures that the entire model only retains Layer Norm (postLN) after the module. The second subsampling, which involves an additional 2x subsampling on top of the initial 4x, uses depthwise separable convolution to reduce computational complexity. The overall structure of the Squeezeformer Block is illustrated in Figure 2.

1) MHA Module: The Adapter layer is added to the attention module, which features a simple structure, as illustrated in Figure 4. It projects the input down to a smaller dimension, passes it through a nonlinear activation function, and then projects it back up to the original dimension. Additionally, a residual connection is established between the input and output of the entire Adapter layer. Introducing the Adapter layer to the attention module in speech recognition tasks offers several benefits. From a structural perspective, the Adapter layer adds additional training modules, allowing the model to ”adapt” to specific downstream tasks using a small set of parameters. First, the simple structure of the Adapter layer effectively reduces the computational burden by lowering and then restoring the dimensionality of features. The use of nonlinear activation functions further enhances the model's representational capacity. Second, this structure enables the model to flexibly adjust the feature extraction process according to the specific task requirements, thereby better capturing the diversity of speech signals and improving generalization. Additionally, the inclusion of residual connections not only accelerates training convergence but also enhances model stability. The overall structure of the MHA module is shown in Figure 3.

2) Convolution Module: The convolution module utilizes ScaleVar layers, post-norm residuals, pointwise convolution, depthwise separable convolution, and Swish activation functions (Scaled Exponential Linear Unit with Squared Highway). The ScaleVar layer, by learning the scaling and bias of the data, enables the model to better adapt to the characteristics and distribution of the input data, thereby enhancing performance, generalization ability, and training stability. The structure of the convolutional module is illustrated in Figure 5.

C. Connected Temporal Classification (CTC) Model

As an end-to-end speech recognition model, the CTC (Connectionist Temporal Classification) model achieves its modeling objectives by optimizing the target function rather than adjusting the model itself. Given the inherent instability of speech and its typical representation in frames, there are often far more audio features in the input than corresponding outputs in speech recognition. To address this many-to-few relationship, a straightforward solution is to remove duplicate input sequences.

Suppose there is an input sequence $x = (x_1, x_2, \ldots, x_T)$, the goal is to generate a target sequence aligned with the input sequence, where T is the length of the $y = (y_1, y_2, \ldots, y_u)$ input sequence and U is the length of the target sequence. The goal of the CTC loss function is to maximize the conditional probability of generating the target sequence y for a given input sequence x. Therefore. You need to define a mapping function π: $x \rightarrow y \cup \{blank\}$ that maps the input sequence to the target sequence along with whitespace. Next, we define the set B of all possible aligned sequences, which contains all possible sequences obtained by inserting whitespace. Therefore, the goal is to find an optimal alignment sequence π^*. Make it match the target sequence y.

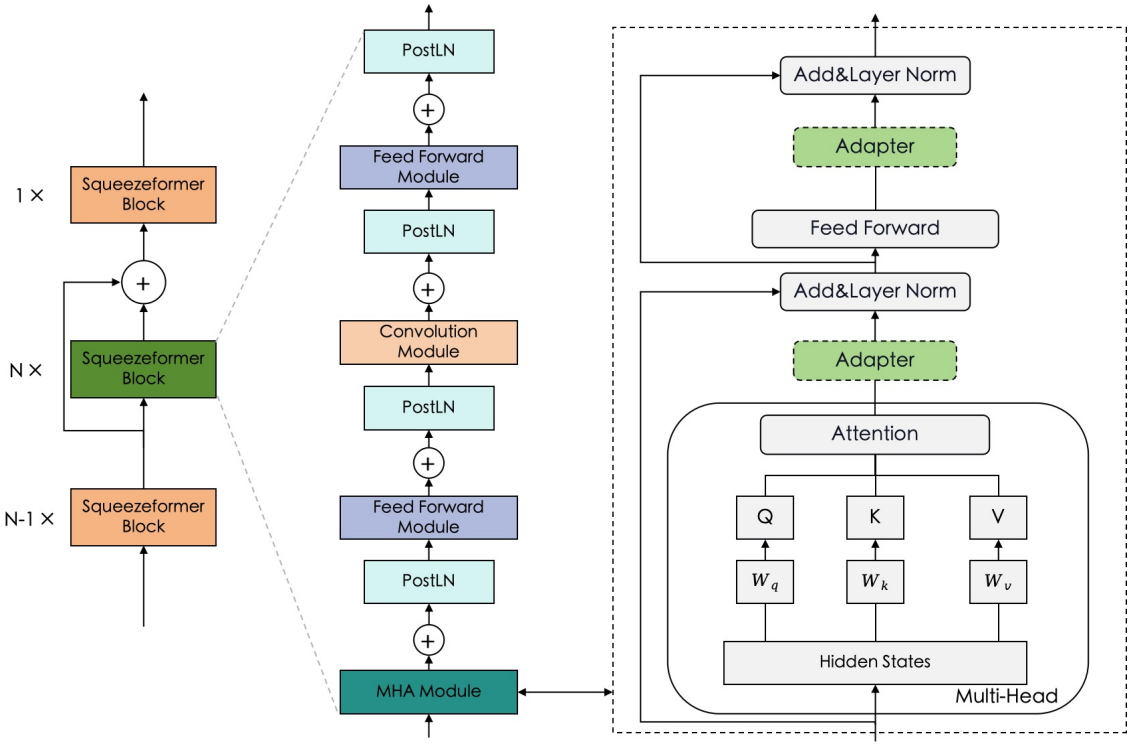

Fig. 3. MHA Moduel

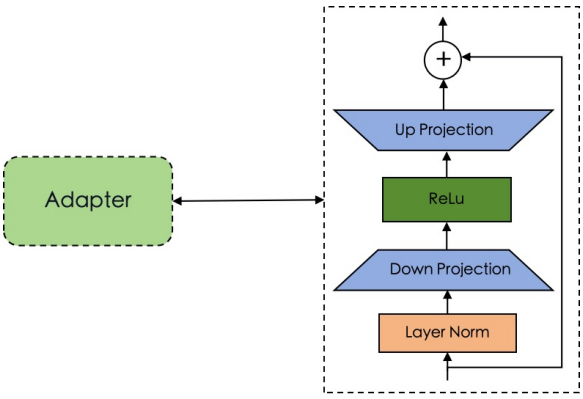

Fig. 4. Adapter Layer

$\alpha_t(s)$ is used to represent the forward probability when the target sequence $y_1, y_2, \ldots, y_s$ is generated in time step t. The forward probability is calculated by dynamic programming algorithm and updated by recursive relation.

$$\alpha_t(s) = \sum_{\pi:|\pi|=s} \alpha_{t-1}\left(\Pi_{t-1}\right) \cdot P\left(y_s \mid x_t\right) \tag{2}$$

Where $a_{t-1}(\pi_{t-1})$ represents the forward probability of generating the subsequence π_{t-1} at time t-1, and $P(y_s|x_t)$ represents the probability from the input feature to the target character y_s. The backward algorithm, as opposed to the forward algorithm, is used to calculate the probability of the subsequent partial target sequence from the current time step to the end of the sequence. In the calculation process, the characters, whitespace, and possibly repeated characters in the target sequence are also considered, and all possible aligned sequences are summed. As shown in formula (3), a dynamic programming algorithm is used to calculate the backward probability, and a recursive relationship is used to update the backward probability.

$$\beta_t(s) = \textstyle\sum_{\Pi:|\Pi|=U-s} \beta_{t+1}\left(\Pi_{t+1}\right) \cdot P\left(\Pi_{t+1} \mid x_{t+1}\right) \cdot p\left(blank \mid x_t\right) \tag{3}$$

Where $\beta_t(s)$ represents the backward probability of generating target sequence $y_{s+1}, y_{s+2}, \ldots, y_U$ from features t+1 to t at time step T, $\beta_{t+1}(\pi_{t+1})$ represents the backward probability of generating subsequence π_{t+1} at time step t+1, $P(\pi_{t+1}|x_{t+1})$ the probability of generating subsequence π_{t+1} from input features, and $P(blank|x_t)$ the probability of generating whitespace from input features. In CTC, the output sequence generated by the model contains characters, whitespace, and possibly duplicate characters from the target sequence. With alignment, we can map the model output sequence back to the target sequence. The alignment process is usually based on the results of the forward and backward algorithms, as well as the output probability distribution of the model. By comparing the forward and backward probabilities, as well as the model output probability distribution, we can determine the most likely alignment to map the model output sequence back to the target sequence. For the input sequence x, the probability of the alignment sequence can be calculated from the forward probability and backward probability:

$$P(\pi|x) = \alpha_T(|\pi|) \cdot \beta_1(|\pi|) \tag{4}$$

Where $\alpha_T(|\pi|)$ is the forward probability of generating the target sequence at the last time step, and is the backward probability of generating the target sequence π at the first time step. Finally, the CTC loss function can be calculated by summing the probabilities of all possible alignment sequences:

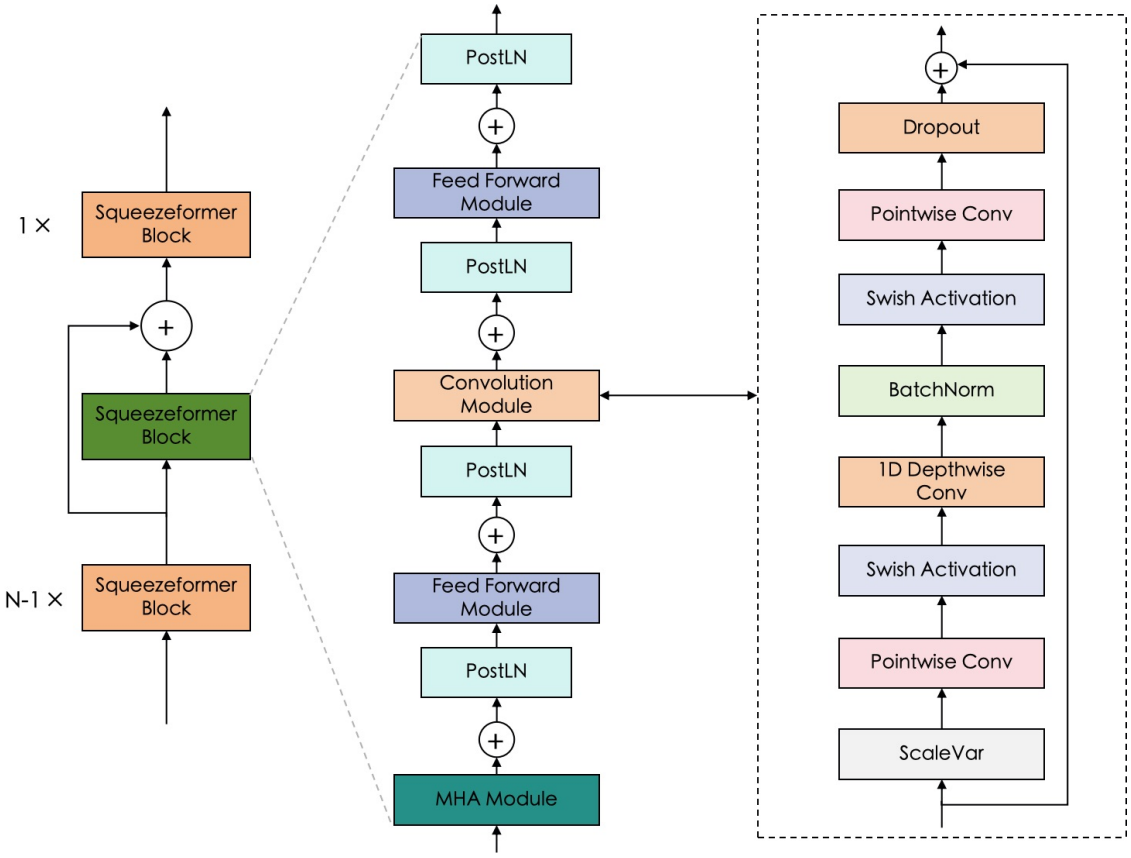

Fig. 5. Convolution Module

$$CTCLoss(x, y) = -ln \sum_{x \in B} P(\pi|x) \tag{5}$$

D. Attention-based Encoding and Decoding Model

LAS (Label-Attentional Sequence-to-Sequence) is a sequence-to-sequence (Seq2Seq) model designed for speech recognition tasks. Unlike traditional speech recognition systems based on deep neural networks, LAS incorporates an attention mechanism that allows the model to dynamically focus on different parts of the input sequence while generating the output sequence. This article also introduces RealFormer into LAS, integrating the Attention Score into the Softmax layer by incorporating the Attention Score from the previous layer, as shown in Equation 6.

$$ResidualAttention(Q', K', V', Prev') = Softmax(\frac{Q'k'^T}{\sqrt{d_k}} + prev')V' \tag{6}$$

This enhancement helps the model better understand the relationships within the input data, improves the model's focus on different parts, and enables it to capture important information more effectively, resulting in more accurate predictions. The structure of the LAS model is shown in Figure 6.

Assume that the input sequence $X = (x_1, x_2, \ldots, x_T)$ and the target sequence $Y = (y_1, y_2, \ldots, y_U)$ are given. The probability that the model generates symbol y_t in the m target sequence at each time step t is $P(y_t|\hat{y} < t, X)$, where $\widehat{y} < t$ represents all symbols up to the seventh position in the sequence generated by the model. The goal is to maximize the conditional probability of a given sequence of targets. First, we assume that each symbol in the target sequence is generated independently, that is, given the input sequence X and the model-generated sequence $\widehat{Y}$, each symbol is generated independently of each other. Therefore, we can write the conditional probability of a given target sequence as a multiplication of the probability of generating each symbol:

$$P(Y|\hat{Y}, X) = \prod_{t=1}^{u} P(y_t|\hat{y} < t, X) \tag{7}$$

For each time step t, the goal is to compute the conditional probability $P(y_t|\hat{y} < t, X)$ of the model generating symbol y_t in the target sequence. This probability can be obtained through the output layer of the model, for example by using the softmax function to convert the model output into a probability distribution. Next, the log-likelihood loss function is defined as the negative log-probability of the sequence generated by the model given the target sequence. Log-likelihood loss is defined as follows:

$$loss = -lnP(Y|\hat{Y}, X) \tag{8}$$

Expanding the logarithmic likelihood loss according to formula 7, we get:

$$Loss = -\ln \sum_{t=1}^{U} p(y_t|\hat{y} < t, X) = -\sum_{t=1}^{U} \ln P(y_t|\hat{y} < t, X) \tag{9}$$

Finally, we train the model by minimizing the loss function to make the model's predictions as close to the target sequence as possible. Optimization algorithms like gradient descent are typically used to achieve this goal.

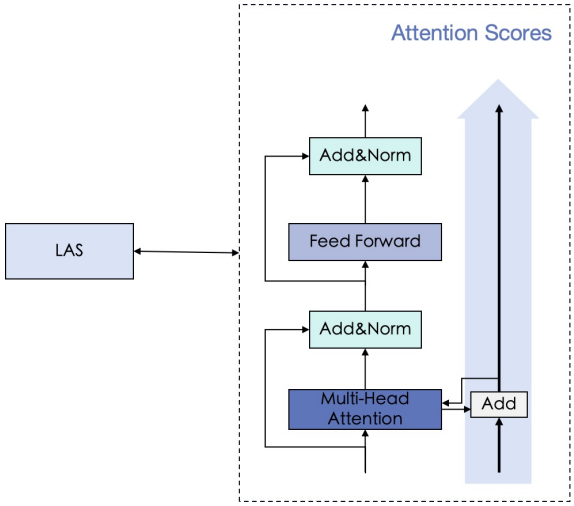

Fig. 6. LAS Model

E. Ctc-attention Joint Model

The CTC model and attention-based encoder-decoder model each have their strengths and limitations. The CTC model assumes independence among the posterior probabilities of the output sequence, which leads to relatively weak modeling of correlations between symbols. Consequently, a language model is often required to further enhance performance. In contrast, the attention-based encoder-decoder model can perform data-driven alignment learning and effectively model long-distance dependencies in context. However, without the monotonicity constraints present in CTC, attention-based models are more susceptible to noise interference, and their complexity increases significantly with sequence length, making training more challenging. To address these limitations, this paper proposes a CTC-attention joint model based on the concept of multi-task learning (MTL). The CTC-attention joint model combines the strengths of both the CTC and attention-based models, allowing them to complement each other. The CTC model provides output probabilities under the independence assumption, while the attention-based model excels at handling long-distance dependencies. Joint learning effectively integrates the advantages of both models, enhancing sequence modeling capabilities.

By using the multi-task training method, we take the optimization objective of the CTC model as an additional training task, and conduct joint training together with the optimization objective of the LAS model. This is equivalent to introducing two different training objectives of CTC and LAS into the whole model, making them share the same encoder, and conducting joint training according to their respective objective functions in the decoding stage. In this way, the training of the entire model can benefit from the way the CTC model learns the monotonic alignment between inputs and outputs. The final loss function is a weighted sum of the losses of the two models to consider the contributions of the CTC and LAS models in the training process. Its loss function is expressed as:

$$L_{MLT} = \lambda L_{CTC} + (1-\lambda)\, L_{LAS} \tag{10}$$

Where, λ is the hyperparameter used to control the weight of the two tasks, $0 \leq \lambda \leq 1$.

IV. Experiments and Analysis

A. Dataset

In this study, we used two publicly available AISHELL-1 and AISHELL-2 datasets, high-quality Mandarin speech recognition corpus recorded in a quiet indoor environment. The AISHELL-1 dataset is widely utilized in the voice community across various fields, including smart homes, autonomous driving, industrial production, and more. The dataset has a total duration of 178 hours, making it of moderate size. It was recorded at a sampling rate of 16 kHz, in mono, with 16-bit resolution, and is stored in WAV format. The AISHEll-2 dataset builds on its predecessor and provides a vast resource for training deep learning models, with more than 20,000 audio samples. Additionally, we divided the dataset into training, validation, and test sets in an 8:1:1 ratio.

B. Model Parameter Configuration

In this experiment, 80-dimensional FBank (Filter bank) feature is used as the input of the model. The frame window size is 25ms and frame shift size is 10ms in the frame dividing stage. The acoustic features in the data set are extracted by Kaldi tool. The data set is also enhanced by variable-speed factors of 0.9x and 1.1x.The input dimension of Squeezeformer module is 256 dimensions, and 4 self-attention heads are set to learn rich feature extraction patterns.Adam optimizer was used to optimize the parameters, and Noam learning rate was used to train the model better.

In this paper, 60 epochs are trained on AISHELL-1 and AISHELL-2 Chinese speech datasets. Four NIVDIAA100-SXM4-40G Gpus were used in all experiments.

C. Evaluation Index

In this paper, characters are used as the fundamental unit, and the evaluation metric adopted is the Character Error Rate (CER). The calculation formula for this metric is as follows:

$$CER = \frac{I+S+D}{N} \tag{11}$$

In the formula, I, S, and D represent the number of inserted, substituted, and deleted characters, respectively, while N represents the number of actual label characters.

D. Acoustic Feature Selection

In this paper, FBank spectral features and Mel frequency cepstrum coefficients are employed as candidate acoustic features for comparative experiments, aiming to select appropriate acoustic features to achieve superior model recognition. To simulate the nonlinear perception characteristics of audio signals by the human ear, FBank successively conducts operations on the original speech signal, such as pre-accentuation, framing, addition of the Hamming window, Fourier transform, squaring of the spectrum amplitude, and application of the Mel filter bank to obtain the logarithmic power spectrum of the corresponding frequency band. MFCC performs an additional Discrete Cosine Transform (DCT) operation based on FBank. This paper extracted 80-dimensional FBank and MFCC features from the dataset, and the baseline

TABLE I
COMPARISON OF ACOUSTIC CHARACTERISTICS BETWEEN FBANK AND MFCC

Acoustic characteristics	$CER(\%)$
MFCC	7.44
FBank	5.51

model was trained with the two acoustic features in a non-pre-training mode. The experimental results are presented in Table 1. It can be observed from Table 1 that the experimental effect of FBank acoustic features is significantly superior to that of MFCC acoustic features. This is because the acoustic characteristics of FBank are more consistent with the essence of the sound signal in fitting the reception characteristics of the human ear. The MFCC feature is the result of the discrete cosine transform of the FBank feature, which increases the time consumption and computational cost and also loses some nonlinear components in the original speech signal. Therefore, in this experiment, the FBank feature is adopted as the acoustic feature of this paper.

E. Comparative Experiment And Result Analysis

Table 1 presents the recognition results on the public AISHELL-1 and AISHELL-2 datasets across different baselines using the same settings. The experimental results demonstrate that the Squeezeformer baseline model proposed in this paper outperforms others, particularly in FBank features, by enhancing the fusion between the attention mechanism and convolutional layers and optimizing the integration of these modules at different levels. The recognition accuracy of the Squeezeformer model is 28% higher than that of the previous Conformer model, highlighting its advantages in speech recognition tasks.

This paper employs a multi-task learning model architecture, with an equal ratio of CTC and ATT tasks (1:1). The model consists of a six-layer neural network designed for speech recognition tasks. The attention mechanism enables the model to focus more effectively on the crucial parts of the input, thereby enhancing recognition accuracy. The character error rate (CER) was lower when Squeezeformer used only the attention mechanism (ATT), but the improvement was not as pronounced as when both the attention mechanism and Connectionist Temporal Classification (CTC) were utilized. CTC allows the model to learn alignment and segmentation of input sequences during training. Although models using only CTC performed slightly worse, CTC played a positive role in boosting model performance when combined with ATT. When Squeezeformer used both ATT and CTC, the CER was significantly reduced, reaching an optimal level of 5.50%. This indicates that the combined effect of the attention mechanism and CTC can effectively enhance the performance of the Squeezeformer model in speech recognition tasks. The specific experimental results are presented in Table 2.

As shown in Table 3, the Squeezeformer + CTC configuration has the smallest number of parameters, at only 19.03M. Both the Squeezeformer + multitasking and Squeezeformer + ATT configurations have 30.69M parameters. This indicates that our model offers a distinct advantage in terms of parameter efficiency. Compared to LSTM and Conformer, the number of parameters is significantly reduced, and when compared to Squeezeformer + ATT, our model achieves improved accuracy while maintaining the same number of parameters.

V. CONCLUSION

In this paper, we propose a multi-task Chinese speech recognition method based on the Squeezeformer model, optimizing the existing Conformer model to enhance performance and efficiency. By replacing the FMCF structure in Conformer with the MF/CF structure in a Transformer style, we fully leverage the convolutional module's strengths in local modeling while employing sequential U-Net for multi-level downsampling and upsampling, effectively reducing computational costs. Additionally, by unifying activation functions and simplifying the model structure through the replacement of redundant LayerNorm and other normalization schemes, we reduce deployment costs and improve the model's practicality and deployability. The introduction of a ScaleVar Layer to the convolutional module enhances the model's flexibility and expressiveness, while the addition of an Adaptor Layer to the MHSA module prevents gradient vanishing and refines the results. On the decoding side, the RealFormer module significantly improves context understanding and the quality of generated results. Crucially, we combine CTC and attention-based encoding and decoding models for multi-task learning, effectively leveraging their strengths to significantly boost the model's performance and speech recognition accuracy. Experimental results demonstrate that the proposed method outperforms the Conformer model on the AISHELL-1 dataset, reducing the number of parameters by 16% and lowering the character error rate (CER) to 5.50%, showcasing superior performance in Chinese speech recognition tasks. This research not only advances speech recognition accuracy through structural optimization and multi-task learning but also offers a more efficient and cost-effective solution for practical applications. Future work may explore further optimizations and broader applications of this approach to other speech recognition scenarios.

REFERENCES

[1] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," *arXiv preprint arXiv:2005.03191*, 2020.

[4] M. Burchi and V. Vielzeuf, "Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 8–15.

[5] C. Zhang, W. Chen, and C. Xu, "Depthwise separable convolutions for short utterance speaker identification," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. IEEE, 2019, pp. 962–966.

[6] X. Chang, A. S. Subramanian, P. Guo, S. Watanabe, Y. Fujita, and M. Omachi, "End-to-end asr with adaptive span self-attention." in *INTERSPEECH*, 2020, pp. 3595–3599.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON THE EXPOSED DATASET AISHELL-1 AND AISHELL-2

Method	Feature	AISHELL-1 $CER(\%)$	AISHELL-2 $CER(\%)$
Speech-Transformer[22]	FBank	11.28	10.17
CTC-based	FBank	15.82	13.20
Speech-Conformer	FBank	8.75	8.62
Squeezeformer	FBank	6.32	5.51

TABLE III
RESULTS OF ABLATION RESEARCH ON THE PROPOSED SQUEEZEFORMER SPEECH RECOGNITION MODEL AND MULTI-TASK METHOD

Method	ATT	CTC	AISHELL-1	AISHELL-2
Squeezeformer	$\surd$	-	11.28	11.08
Squeezeformer	-	$\surd$	6.91	6.92
Squeezeformer	$\surd$	$\surd$	5.50	5.01

TABLE IV
COMPARISON OF PARAMETERS QUANTITY AND PERFORMANCE OF DIFFERENT MODELS ON AISHELL-1

Model	Parameter Quantity	$CER(\%)$
Speech-Transformer	34.28M	11.28
CTC-based	43.90M	15.82
Speech-Conformer	36.71M	8.75
Squeezeformer+ATT	30.69M	6.32
Squeezeformer+CTC	19.03M	6.91
Squeezformer+ATT+CTC	30.69M	5.50

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[8] R. He, A. Ravula, B. Kanagal, and J. Ainslie, "Realformer: Transformer likes residual attention," *arXiv preprint arXiv:2012.11747*, 2020.

[9] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1871–1880.

[10] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[11] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[12] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts," *arXiv preprint arXiv:1906.00672*, 2019.

[13] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[14] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," *Advances in neural information processing systems*, vol. 31, 2018.

[15] S. Kim, A. Gholami, A. Shaw, N. Lee, K. Mangalam, J. Malik, M. W. Mahoney, and K. Keutzer, "Squeezeformer: An efficient transformer for automatic speech recognition," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9361–9373, 2022.

[16] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[17] V. SOKOLOVSKII, "End-to-end speech recognition for low-resource languages," *Signal Processing (ICASSP)*, vol. 5884, p. 5888, 2018.

[18] S. Karita, "Recent advances in end-to-end speech recognition," 2019.

[19] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.

[20] F. Tao and C. Busso, "End-to-end audiovisual speech recognition system with multitask learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 1–11, 2020.

[21] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[22] J. Li, X. Wang, Y. Li *et al.*, "The speechtransformer for large-scale mandarin chinese speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7095–7099.