information

Article

LAS-Transformer: An Enhanced Transformer Based on the Local Attention Mechanism for Speech Recognition

Pengbin Fu *, Daxing Liu and Huirong Yang

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; liudx@emails.bjut.edu.cn (D.L.); yanghuirong@bjut.edu.cn (H.Y.)
* Correspondence: fupengbin@bjut.edu.cn

Abstract: Recently, Transformer-based models have shown promising results in automatic speech recognition (ASR), outperforming models based on recurrent neural networks (RNNs) and convolutional neural networks (CNNs). However, directly applying a Transformer to the ASR task does not exploit the correlation among speech frames effectively, leaving the model trapped in a sub-optimal solution. To this end, we propose a local attention Transformer model for speech recognition that combines the high correlation among speech frames. Specifically, we use relative positional embedding, rather than absolute positional embedding, to improve the generalization of the Transformer for speech sequences of different lengths. Secondly, we add local attention based on parametric positional relations to the self-attentive module and explicitly incorporate prior knowledge into the self-attentive module to make the training process insensitive to hyperparameters, thus improving the performance. Experiments carried out on the LibriSpeech dataset show that our proposed approach achieves a word error rate of 2.3/5.5% by language model fusion without any external data and reduces the word error rate by 17.8/9.8% compared to the baseline. The results are also close to, or better than, other state-of-the-art end-to-end models.

Keywords: end-to-end model; speech recognition; Transformer; local attention

Citation: Fu, P.; Liu, D.; Yang, H. LAS-Transformer: An Enhanced Transformer Based on the Local Attention Mechanism for Speech Recognition. *Information* **2022**, *13*, 250. https://doi.org/10.3390/info13050250

Academic Editor: Ognjen Arandjelović

Received: 4 April 2022
Accepted: 11 May 2022
Published: 13 May 2022

1. Introduction

With the development of deep learning techniques, end-to-end speech recognition models have received significant attention because they simplify the training and decoding process considerably. Indeed, they directly learn speech-to-text mapping with purely neural network systems, while the traditional HMM-based methods need a hand-crafted pronunciation dictionary, an independent acoustic model, and a complex decoding system [1].

End-to-end models can be broadly divided into three different categories depending on their implementations of soft alignment: connectionist temporal classification (CTC-based) [2–5], recurrent neural networks-transducer (RNN-T) [6–9], and attention-based encoder–decoder (AED) [10–13]. CTC [2] uses a single network structure to map input sequences directly to output sequences to solve the problem of data alignment and direct modeling. Essentially, CTC is a loss function, but it solves a hard alignment problem while calculating the loss. CTC assumes that the output elements are independent of each other, so interdependencies cannot be modeled in the output sequence. Graves proposed the RNN-T [6] model to solve the dependency modeling problem. The RNN-T model consists of three sub-networks—a transcription network, prediction network, and joint network—using a different path generation process and path probability calculation method to that of CTC. However, the RNN-T and CTC calculation processes include many evident unreasonable paths. To solve this problem, the AED method [10] uses the attention mechanism to calculate the soft alignment information between input and output data directly.

In 2017, Vaswani et al. proposed a network Transformer [14] based entirely on the attention mechanism and successfully applied it to natural language processing (NLP).

With the flexibility of the Transformer, data can be processed with the Transformer when they have a serialized nature. Therefore, having a typical encoder–decoder structure, Transformers have been widely used in speech recognition [15–18].

Transformers have the advantage of flexibility, but they also have the problem of insensitivity to data. Unlike the recurrent neural networks with sequential and time-domain invariance brought about by Markovian structures [19], Transformer structures lack explicit inductive bias [20]. For speech recognition, Transformers cannot exploit the high correlation of speech frames in speech data and capture insufficient local contextual information. Therefore, the question of how to accurately focus on the local context in a Transformer speech recognition system is a problem to be solved.

At the same time, due to the number of time frames of an audio sequence being significantly larger than the number of output text labels [11], inputting the speech frame sequences into the basic Transformer leads to the problems of high computational effort and excessive redundant information. To address these problems, current mainstream approaches use convolutional subsampling to reduce the length of the input speech sequence during the encoding period and obtain an embedded representation of the speech frames. However, using operations such as convolution and pooling to reduce the data size of the input speech frames introduces the problem of data loss from the speech frames, leading to performance degradation [21].

To address these problems, we propose the LAS-Transformer (local attention speech-Transformer). Specifically, we use depthwise separable convolution [22] for subsampling, and there is no pooling layer in the subsampling layer to ensure the maximum integrity of speech information at the embedding layer. Moreover, we propose a local attention module to explicitly incorporate the highly correlated features of speech frames into the attention calculation, which can effectively extract local contextual information and significantly compensate for the defects of the basic Transformer's local feature extraction. Additionally, the absolute positional embedding method is replaced by relative positional embedding to improve the representation of location information, which calculates the self-attention score more accurately.

The main contributions of this paper are as follows:

1. We propose a local attention module based on the highly correlated features of speech frames. The local self-attentive module uses a high correlation of speech frames as a priori knowledge to quickly capture the local information of the speech sequence.
2. We propose a depthwise separable convolution subsampling layer, which reduces the parameters of the model and preserves the position information to a great extent.
3. We replace the Transformer's native absolute positional embedding with relative positional embedding. The relative position encoding method can enhance the position information representation, which not only contains the relative position relationship, but also expresses the direction information.

We compare the proposed method with other end-to-end models on the public dataset LibriSpeech [23]. The experimental results show that the proposed LAS-Transformer reduces the word error rate by 17.8/9.8% compared to the baseline. We further explore the local attention model and perform ablation experiments to demonstrate the effectiveness of local attention, relative position encoding, and depthwise separable convolution subsampling.

2. Related Work

2.1. Transformer

NLP has made significant progress in recent years, much of which is attributed to the Transformer [14]. A typical encoder–decoder structure captures sequence order dependencies using a multi-head dot product self-attention mechanism, represents sequence positions with absolute positional embedding, and uses fully connected layer computation. It significantly reduces the training time compared to traditional CNNs and RNNs.

The dot product self-attention focuses on the similarity between the query vector and each of the key-value vectors as weights and then sums the weights over all the real-valued

vectors, as defined as Equation (1), where $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V} \in \mathbb{R}^{L\times d}$ denote the query matrix, the key-value matrix, and the real-value matrix, respectively; L denotes the length of the input vector; d denotes the dimension of the input vector; and $\boldsymbol{QK^{\mathrm{T}}}$ is the attention matrix. To prevent backpropagation from falling into regions with a small gradient, the variance of the attention matrix is reduced by the $\sqrt{d}$.

$$\mathrm{ATT}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{QK^{\mathrm{T}}}}{\sqrt{d}}\right)\boldsymbol{V} \tag{1}$$

Multi-head attention is based on self-attention and takes advantage of the attentional representation of different subspaces, defined as Equations (2) and (3). Multi-head attention calculates h times self-attentions, where h denotes the number of self-attention heads. Before calculating each self-attention, $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ are converted into more distinct representational vectors by three linear projections. Each self-attention is computed independently, and then their outputs are concatenated and projected linearly.

$$\mathrm{MHA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{Concat}[\boldsymbol{H}_1,\boldsymbol{H}_2,\ldots,\boldsymbol{H}_h]\boldsymbol{W}^h \tag{2}$$

$$\boldsymbol{H}_i = \mathrm{ATT}\left(\boldsymbol{QW}_i^Q,\boldsymbol{KW}_i^K,\boldsymbol{VW}_i^V\right) \tag{3}$$

Equation (3) has the same dimension as Equation (1). The projection matrix is $\boldsymbol{W}_i^Q \in \mathbb{R}^{d\times d_q}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d\times d_k}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d\times d_v}$, $\boldsymbol{W}^h \in \mathbb{R}^{hd_v\times d}$, $d_q = d_k = d_v = d/h$.

Absolute positional embedding is used to represent the position information as a way for the model to learn the sequential relationship of the sequence. As shown in Equation (4), pos and k represent the position and the dimension. Each dimension of the position encoding $\boldsymbol{U}$ has a sinusoidal signal, and each position can be represented by other positions through triangular transformation.

$$\begin{aligned}\boldsymbol{U}_{pos,2k} &= \sin\left(\frac{pos}{1000^{2k/d}}\right)\\ \boldsymbol{U}_{pos,2k+1} &= \cos\left(\frac{pos}{1000^{2k/d}}\right)\end{aligned} \tag{4}$$

2.2. CTC-Transformer

The attention mechanism in the Transformer is too flexible because it allows extreme out-of-order alignment before the input and output sequences. This mechanism is extremely suitable for certain tasks, such as machine translation, where the input and output words are not in the same order; however, in speech recognition, the output text sequences and the input speech sequences correspond to each other in the same order. The Joint CTC-Transformer [16] is improved to address this problem. First, the CTC structure is introduced to constrain the optimization direction of the encoder in the Transformer with the help of the monotonic property of CTC so that the output sequence and the input sequence can be aligned quickly. Secondly, individual speech frames in a speech sequence, unlike units such as words and phrases, have no evident meaning. Only several adjacent speech frames can form a more meaningful unit, such as a phoneme or character. The length of the input speech sequence is several times longer than the length of the output sequence.

To address these two problems, the Joint CTC-Transformer introduces a convolutional subsampling layer, consisting of a CNN with a temporal dimension and ReLU activation function, to shorten the length of the input sequence significantly and obtain a more meaningful and efficient audio embedding representation. Since the Joint CTC-Transformer introduces multi-objective optimization and decoding combines the CTC structure output and Transformer decoder output, it is well suited for speech recognition.

3. Methods

In this paper, we propose a new speech recognition structure, the LAS-Transformer. The model structure is shown in Figure 1. In the encoder part, audio features first pass

through a depth-separable convolutional subsampling layer and then into the stacked N_e times of encoder layers. Each encoder layer consists of two sub-layers: the first sub-layer is a multi-head local attention layer fused with relative positional embedding, and the second sub-layer is a feedforward layer. Each sub-layer has a residual structure [24] and pre-norm [25], and none of the encoder layer parameters are shared. Finally, the layer-normalized speech representation is fed to the CTC module to obtain the probability distribution of the CTC output, which consists of a fully connected layer and a softmax layer. In the decoder, one-hot encoding first passes through an embedding layer with absolute position encoding and then enters the decoder layer of stacked N_d. Each decoder layer consists of three sub-layers: the first sub-layer is a masked multi-head attention layer, the second sub-layer is a multi-head attention layer that fuses the speech representation of the encoder, and the third sub-layer is a feedforward layer. Each sub-layer has a residual structure and pre-norm, and none of the decoder layer parameters are shared. Then, after layer normalization and linear layers, the final probability distribution output is obtained using softmax. The three improvements made in this paper are in the encoder module, as will be described in detail below. The decoder module is the same as the Transformer decoder and will not be described again.

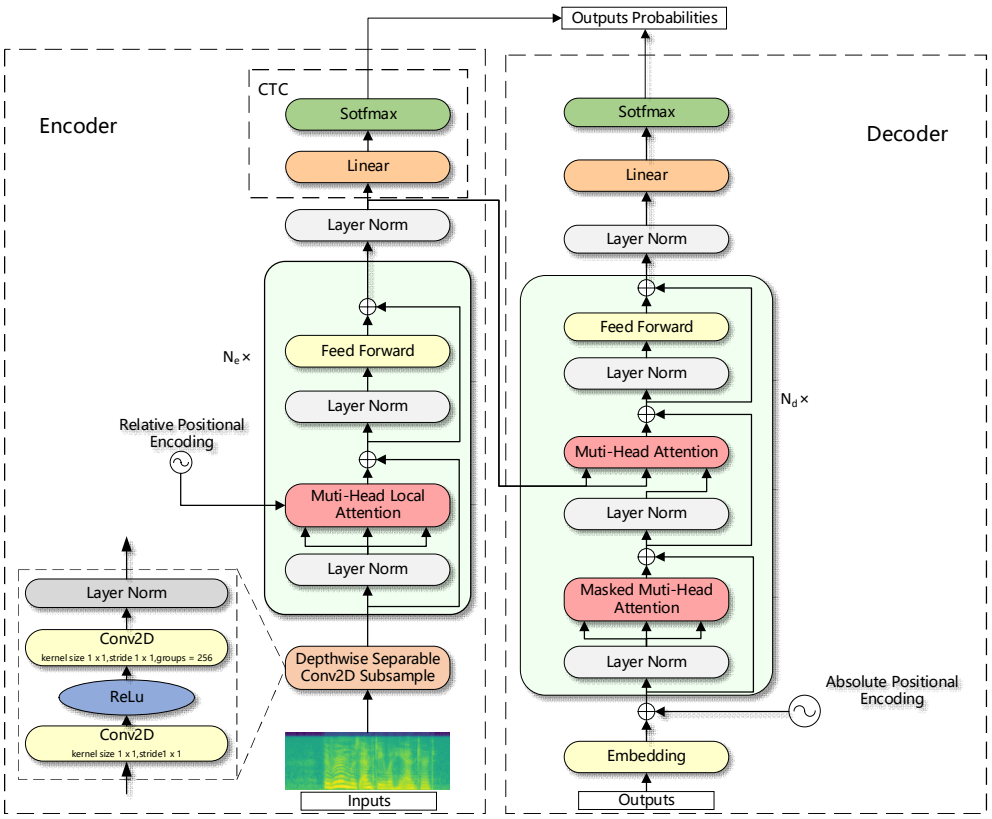

Figure 1. The overall framework of the LAS-Transformer.

3.1. Depthwise Separable Convolution Subsampling Layer

The common convolutional subsampling layer consists of the convolutional layer, ReLU activation, and a maximum pooling layer, as shown in Figure 2a. The maximum pooling layer is located at the input layer of the model and is likely to discard some contextual information. The maximum pooling layer is removed from the subsampling layer to ensure the integrity of the position information. The position information is important and contains the position relationship of the speech frames. When the speech frames are close or equal, the attention module can only use the position information to distinguish between the different outputs. Layer normalization is added to the subsampling layer. Layer normalization is able to keep the training and testing sets independent and

identically distributed, making the loss landscape smoother. Specifically, the gradient of the loss function after layer normalization processing becomes smaller. The normal convolution in the subsampling layer is replaced by a depth-separable convolution [22] with a step size of 2. The depthwise separable convolution reduces the parameters and enables channel and space separation computation. The first layer is called a depthwise convolution. It performs lightweight filtering by applying a single convolutional filter to each input channel. The second layer is a 1×1 convolution, called a pointwise convolution, which is responsible for building new features through the depthwise separable convolution subsampling layer, and is shown in Figure 2b.

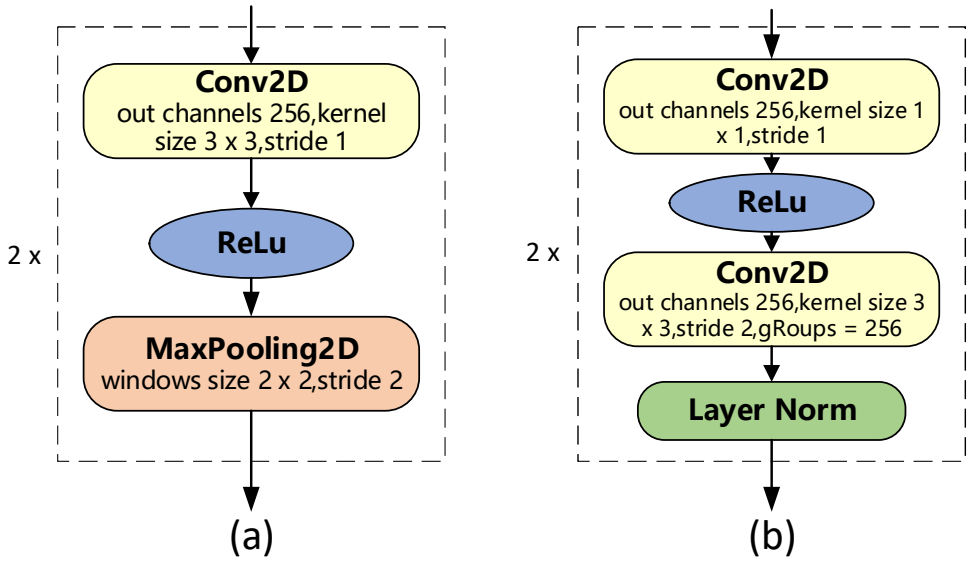

Figure 2. (**a**) The structure of the convolution subsampling layer. (**b**) The structure of the subsampling layer replaced with depthwise separable convolution.

3.2. Relative Positional Embedding

The original Transformer uses trigonometric absolute positional embedding, and the attention matrix $\boldsymbol{A}^{abs}$ can be factorized into four parts as Equation (5), where $\boldsymbol{E} \in \mathbb{R}^d$, $\boldsymbol{W} \in \mathbb{R}^{d \times d}$, and $\boldsymbol{U} \in \mathbb{R}^d$ represent the input vector, the weight matrix, and the absolute position vector, respectively, and i and j represent the query vector position and the key vector position, respectively. Among the four parts of $abcd$, only d contains both the query vector position and the key vector position and is most likely to contain relative position information. It has been shown that the relative position information is included in the absence of $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$, but the relative position relationship is lost when $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ are introduced [18].

$$A_{i,j}^{abs} = \underbrace{\boldsymbol{E}_{x_i}^T \boldsymbol{W}_q^T \boldsymbol{W}_k \boldsymbol{E}_{x_j}}_{(a)} + \underbrace{\boldsymbol{E}_{x_i}^T \boldsymbol{W}_q^T \boldsymbol{W}_k \boldsymbol{U}_j}_{(b)} + \underbrace{\boldsymbol{U}_i^T \boldsymbol{W}_q^T \boldsymbol{W}_k \boldsymbol{E}_{x_j}}_{(c)} + \underbrace{\boldsymbol{U}_i^T \boldsymbol{W}_q^T \boldsymbol{W}_k \boldsymbol{U}_j}_{(d)} \tag{5}$$

Therefore, following Transformer-XL [26], the absolute positional embedding is replaced by the relative positional embedding, defined as Equation (6). Relative positional embedding replaces both the projection $\boldsymbol{U}_i^T \boldsymbol{W}_q^T$ of the query vector position in part c directly with the learnable parameter $u \in \mathbb{R}^d$ and the projection $\boldsymbol{U}_i^T \boldsymbol{W}_q^T$ of the query vector position in part d directly with the learnable parameter $v \in \mathbb{R}^d$. Since the query vectors are the same for all query positions, the deviations of different positions are represented by the same learnable parameter. The absolute position, $\boldsymbol{U}_j \in \mathbb{R}^{d \times L_{\max}}$, of the key-value vector is replaced by the relative position vector $\boldsymbol{U}_{i-j} \in \mathbb{R}^{d \times L_{\max}}$, and $i - j$ is used to represent the relative position, which is calculated using Equation (4). Relative positional embedding

can represent not only the relative position information, but also the direction information. Direction information implies that $\boldsymbol{U}_{i-j}$ is not equal to $\boldsymbol{U}_{j-i}$.

$$\boldsymbol{A}_{i,j}^{\text{rel}} = \underbrace{\boldsymbol{E}_{x_i}^T \boldsymbol{W}_q^T \boldsymbol{W}_{k,E} \boldsymbol{E}_{x_j}}_{(a)} + \underbrace{\boldsymbol{E}_{x_i}^T \boldsymbol{W}_q^T \boldsymbol{W}_{k,U} \boldsymbol{U}_{i-j}}_{(b)} + \underbrace{\boldsymbol{u}^T \boldsymbol{W}_{k,E} \boldsymbol{E}_{x_j}}_{(c)} + \underbrace{\boldsymbol{v}^T \boldsymbol{W}_{k,U} \boldsymbol{U}_{i-j}}_{(d)} \tag{6}$$

3.3. Local Attention Layer

In the NLP field, the lower layers in Transformer tend to capture the short-term dependencies of adjacent words and higher layers tend to capture long-term dependencies. In the encoder part of the Joint CTC-Transformer, both the lower and higher levels tend to capture the short-term dependencies of speech frames, and the self-attention weight matrix is shaped in the manner of a banded diagonal array, as shown in Figure 3.

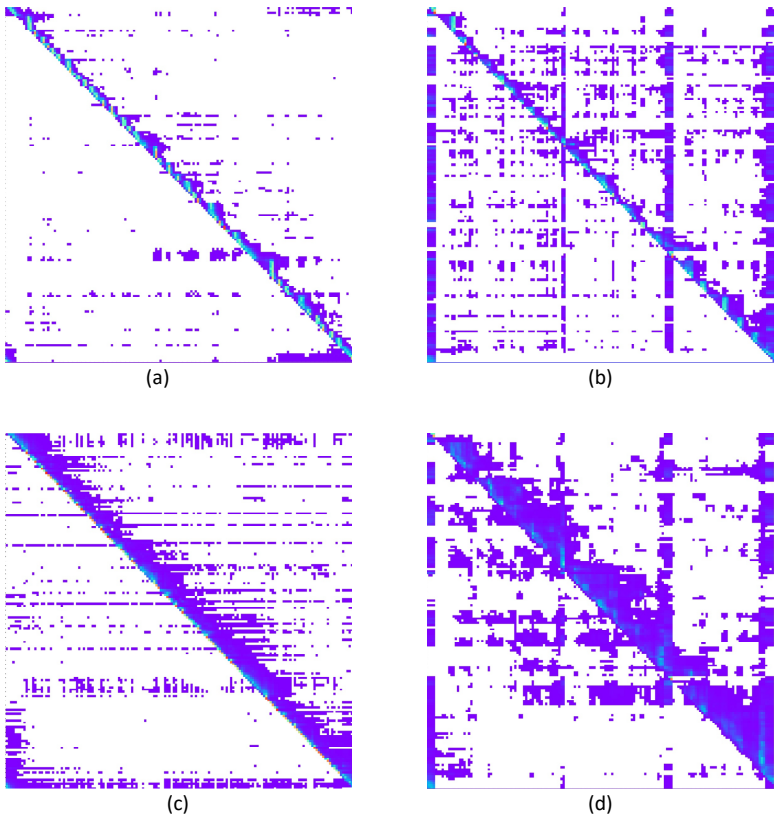

Figure 3. (**a**) Encoder first-layer attention matrix heat map. (**b**) Encoder fourth-layer attention matrix heat map. (**c**) Encoder eighth-layer attention matrix heat map. (**d**) Encoder twelfth-layer attention matrix heat map.

Figure 3 shows the results of the normalization of the self-attention matrix of utterance 3081-166546-0015 in LibriSpeech dev-clean at the encoder, in which the rows indicate the query vector positions and the columns correspond to the key vector positions. It can be seen that most of the speech frames have strong dependencies only with a number of neighboring frames. However, the advantage of the original self-attention mechanism is that it captures global context dependencies and is inadequate for capturing local context dependencies. To address this problem, we propose a parameterized local attention mechanism that adds location-based a priori information $\boldsymbol{B}$ to the native attention. The structure is shown in Figure 4.

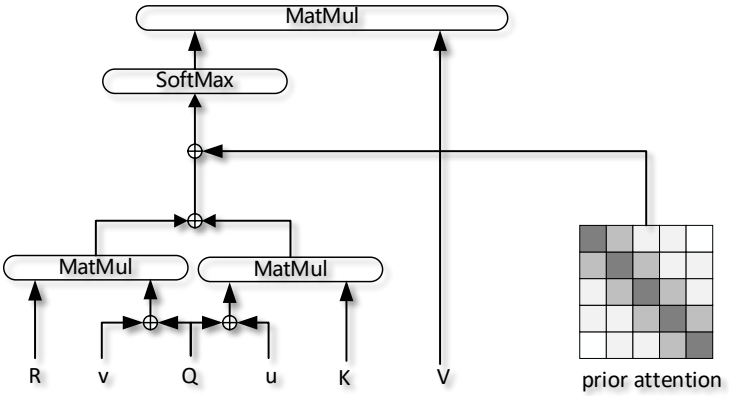

Figure 4. The structure of the local attention layer.

The a priori attention matrix is fused with the generated attention matrix according to Equation (7):

$$\mathrm{ATT}(\boldsymbol{Q},\boldsymbol{K}) = \mathrm{softmax}\left(\frac{\boldsymbol{A}^{\mathrm{rel}}_{Q,K}}{\sqrt{d}} + \boldsymbol{B}\right) \tag{7}$$

where $\boldsymbol{A}^{\mathrm{rel}}_{Q,K}$ is calculated by Equation (6); $\boldsymbol{B} \in \mathbb{R}^{L\times L}$ denotes the location of the a priori information; L denotes the sequence length; and d denotes the dimensionality of each head of the model. Due to the exponential operation in the softmax function, adding a bias $b_{i,j} \leq 0$ to the attention matrix is approximated by multiplying the weights $w \in (0,1]$. $b_{i,j}$ is calculated by Equation (8).

$$b_{i,j} = \begin{cases} -\frac{(i-j)^2}{l_i^2}, & if\ i-j \leq s \\ -\frac{s^2}{l_i^2}, & if\ i-j > s \end{cases} \tag{8}$$

where i denotes the position of the query vector; j denotes the position of the key-value vector; l_i denotes the window size of the query vector concern, and s denotes the truncation distance. A priori information is truncated when i and j are far away, relative to each other to avoid the approximate one-hot output caused by a priori information with an absolute value that is too large. Considering the variability of different pronunciations, the window size should be different for different query vector concerns; thus, a parameterized window size calculation method is used rather than a fixed window size. The calculation method is shown in Equation (9).

$$l_i = I\cdot\sigma\left(\boldsymbol{U}^T g(\boldsymbol{W}(\boldsymbol{E}_{x_i} + \boldsymbol{u} + \boldsymbol{v}))\right) \tag{9}$$

where $\boldsymbol{E}_{x_i}$ denotes the query vector at position i; $\boldsymbol{u}$ and $\boldsymbol{v}$ are the trainable relative positional embedding parameters in Equation (6); and $\boldsymbol{W} \in \mathbb{R}^{d\times 2d}$, $\boldsymbol{U} \in \mathbb{R}^{2d}$ are the learnable parameters. g denotes the tanh activation function; σ denotes the sigmoid activation function; and I denotes the length of the sequence. Following the Transformer feedforward layer, the window size information is first projected to a higher-dimensional space ($2d$) so that, using the tanh activation function, the sparse data are easier to obtain, and then they are linearly transformed to the original dimension. Finally, the data are normalized to $[0,1]$ by the sigmoid function and then multiplied with the sequence length to obtain the final window size.

3.4. Training and Decoding

During model training, the decoder and CTC modules calculate the posterior probability distributions $P_{\mathrm{att}}(\boldsymbol{Y}|\boldsymbol{X})$,$P_{\mathrm{ctc}}(\boldsymbol{Y}|\boldsymbol{X})$, respectively. The loss function is a weighted sum of the negative likelihood logarithms of the posterior distributions, calculated according to Equa-

tion (10). In Equation (10), $-\log\ P_{\text{att}}(\boldsymbol{Y}|\boldsymbol{X})$ is the cross-entropy loss and $-\log\ P_{\text{ctc}}(\boldsymbol{Y}|\boldsymbol{X})$ is the CTC loss. α is the hyperparameter used to measure the weight of the CTC module.

$$L = -1 - \alpha \log\ P_{\text{att}}(\boldsymbol{Y}|\boldsymbol{X}) - \alpha \log\ P_{\text{ctc}}(\boldsymbol{Y}|\boldsymbol{X}) \tag{10}$$

In the decoding phase, given the speech sequence $\boldsymbol{X}$ and the token predicted in the previous step, the next token is computed by beam search combined with the decoder, CTC, and language model, as defined in Equation (11). $\boldsymbol{y}^*$ in Equation (11) denotes the set of hypotheses for the target sequence, and λ and γ are both hyperparameters used to adjust the CTC module and language model score weights during decoding.

$$\widehat{\boldsymbol{Y}} = \underset{\boldsymbol{Y} \in \boldsymbol{y}^*}{\operatorname{argmax}} (1-\lambda) \log\ P_{\text{att}}(\boldsymbol{Y}|\boldsymbol{X}) + \lambda \log P_{\text{ctc}}(\boldsymbol{Y}|\boldsymbol{X}) + \gamma \log P_{\text{lm}}(\boldsymbol{Y}) \tag{11}$$

4. Experiments

We replicated the Espnet Transformer model [16] as the baseline and trained it using the LibriSpeech public dataset to compare it with other advanced speech recognition methods. The effectiveness of our proposed model was verified by comparison and ablation analysis experiments on the LibriSpeech dataset.

4.1. Datasets and Evaluation Metrics

The performance of our proposed model was verified using the LibriSpeech public dataset [23], which has a total of 980 h of English speech data and corresponding text data from the Internet Archive and the Gutenberg Project. The dataset is divided into two parts: the clean part and the other part. The clean part has less noise and is easy to recognize. The other part has poor audio quality and is more difficult to recognize. Furthermore, the dataset is further divided into train-clean-100, train-clean-360, train-other-500, dev-clean, dev-test, test-clean, and test-other.

Word error rate (WER) was used to evaluate the performance of the model, as defined in Equation (12), where S is the number of words replaced; D is the number of words deleted; I is the number of words inserted; $S + D + I$ is the shortest edit distance; and N is the number of words in the correct word sequence:

$$WER = 100 \times \frac{S + D + I}{N}\% \tag{12}$$

4.2. Experimental Details

The input data of the model were 80-dimensional Fbank features. We performed experiments using $N_e = 12$ and $N_d = 6$ Transformer layers for the encoder and decoder, respectively, with $d = 256$, $h = 4$ and $s = 10$. The number of mask windows in the SpecAugment layer [27] was 2 in both time and frequency dimensions, and the window size was 30. Considering the variable length of input data sequences, we used dynamic Batch to improve the memory utilization. Batchbin is 5,000,000. The gradient accumulation was 4. We used the Adam optimizer, with a learning rate scheduling similar to [16]. The dropout method in both the position encoding and attention matrix was 0.1. Label smoothing was used to calculate the cross-entropy loss; the smoothing weight was 0.1; and the CTC loss weight $\alpha = 0.3$. The ten models with the lowest error rate were saved to calculate the average as the final result, according to the word error rate size sorting at the end of the training.

In the decoding stage, the score weight of the CTC module was 0.3, the score weight of the language model was 0.6, and the bundle width was 10. The language model used a Transformer encoder structure, where the number of encoder layers was 6, the model dimension was 512, the number of heads was 8, and the attention dimension was 256.

We used the same data enhancement in all experiments to ensure fairness, including speed perturbation [28] and SpecAugment [27]. The parameters were set to the same case for all experiments.

4.3. Comparison Experiments

To verify the effectiveness of the models, we used the Espnet Transformer model as the baseline system and selected several popular speech recognition models for comparison. We used WER to verify the overall recognition performance of the LAS-Transformer and the comparison models. The experimental results are shown in Table 1.

Table 1. Comparison of WER of each model on LibriSpeech dataset.

Network Structure	Dev-Clean	Dev-Other	Test-Clean	Test-Other
QuartzNet	-	-	2.69	7.25
Espnet Transformer	2.2	5.6	2.6	5.7
LAS+Specaugment	-	-	3.2	9.8
LSTM Transducer	2.17	5.28	2.23	5.74
Hybrid model with Transformer rescoring	-	-	2.60	5.59
baseline	2.5	5.9	2.8	6.1
LAS-Transformer	2.2	5.2	2.2	5.5

As Table 1 shows, the error rates for our baseline experiments were 2.5/5.9/2.8/6.1%, with an increase in error rates in relation to the original Espnet Transformer model. On the one hand, our baseline model dimension ($d = 256$) is half that of the Espnet Transformer model, so the number of parameters is half that of the original paper, which must lead to an increase in error rate. On the other hand, a larger batchbin = 30,000,000 was used in the Espnet Transformer, and it is noted that a larger batchbin significantly improves the results. Therefore, in this paper, we avoided long training times with GPU memory limitations and used $d = 256$ and batchbin = 5,000,000, which causes performance loss.

Comparing the baseline experiments with our proposed model, our proposed approach achieves relative WER reductions of 21.4% and 9.8% for the test-clean and 2.2% and 5.5% for the test-other, respectively. This validates the reasonableness and effectiveness of our proposed LAS-Transformer model. Table 1 shows some advanced speech recognition models. QuartzNet's [29] design is a convolutional model trained with CTC loss. Espnet Transformer [16] is a Joint CTC-Transformer model. LAS+Specaugment [27] is a sequence to sequence framework with Specaugment. LSTM Transducer [9] improves external language model and an estimated internal LM. The hybrid model with Transformer rescoring [30] leverages the Transformer to improve hybrid acoustic modeling. Our proposed model has a lower error rate than them, a 18.2/15.4/31.2/1.4/15.4% error rate reduction on the test-clean subset, and a 24.1/3.5/43.8/4.1/1.6% error rate reduction on the test-other subset. Specifically, LAS-Transformer achieves better performance with half the number of parameters compared to Espnet Transformer. This indicates that there is indeed a problem in indirectly applying Transformer to the ASR task, and the proposed model solves the problem to some extent.

4.4. Ablation Experiments

To further determine the effectiveness of our proposed method and explore the contribution of each module, each module was separated for the experiments in this paper, and Table 2 shows the performance of the proposed LAS-Transformer model and the separately improved model for each subset of LibriSpeech recognition.

Table 2. Results of the ablation study of the proposed LAS-Transformer model.

Network Structure	Dev-Clean	Dev-Other	Test-Clean	Test-Other
baseline	2.5	5.9	2.8	6.1
baseline + relative position coding	2.3	5.6	2.4	5.7
baseline + local attention	2.4	5.5	2.5	5.6
baseline + depthwise separable convolution subsampling	2.5	5.8	2.7	5.9
LAS-Transformer	2.2	5.2	2.2	5.5

According to Table 2, the comparison of the baseline experiment and the baseline + relative positional embedding model reveals that the relative positional model embedding reduces the error rate of each test subset by 8.0/5.0/14.3/6.5%. This justifies the replacement of absolute positional embedding with relative positional embedding and indicates the effectiveness of the relative position in the field of speech recognition, which improves the representation of temporal information to some extent. Comparing the baseline experiment and the baseline + local attention model, the local attention model reduces the error rate of each test subset by 4.0/5.1/10.7/8.2%; adding the local attention module can effectively improve the performance of the model. Further observation shows that, compared with relative positional embedding, the local attention on both dev-other/test-other performance is improved by 10.0/26.1%. Combined with the unclear characteristics of speech in these two subsets, this indicates that the local attention module can effectively capture the local dependencies of speech frames. Comparing the baseline experiment and the baseline + depthwise separable convolution subsampling model, it was found that the depthwise separable convolution subsampling model improves the performance to a lesser extent, reducing the word error rate by 1.6/3.5/3.2% under dev-other/test-clean/test-other.

In addition, Table 3 shows that the performance of our LAS-Transformer model is further improved after fusing three modules (relative position encoding, local attention, and depthwise separable convolution subsampling), achieving a word error rate of 2.2/5.5% on test-clean/test-other. This proves the effectiveness of our proposed LAS-Transformer model and also verifies that local attention can effectively obtain local contextual information to make up for the Transformer's shortcomings in this area. LAS-Transformer can incorporate the highly correlated features of speech frames to a certain extent.

Table 3. Results of the exploratory experiments of the LAS-Transformer.

Network Structure	Dev-Clean	Dev-Other
baseline	2.5	5.9
baseline + mixed local attention	2.5	6.2
baseline + full local attention	2.4	5.5

4.5. Exploratory Experiments

To further explore the effect of local attention, we designed a comparison experiment between full local attention and mixed local attention, in which full local attention indicates that all heads in self-attention add a priori knowledge, and mixed local attention indicates that half of the heads add and the other half do not add a priori knowledge. This is because most of the self-attention matrices in the baseline experiments are shaped in the manner of diagonal arrays. However, there are still small parts of the attention matrices that take global information into account, so mixed local attention aims to learn information at different scales in different subspace models and avoid full local attention from introducing too much a priori knowledge to affect the model's performance. The variation curves of loss and accuracy for the validation set dev-other are shown in Figure 5, and the final results are compared in Table 3

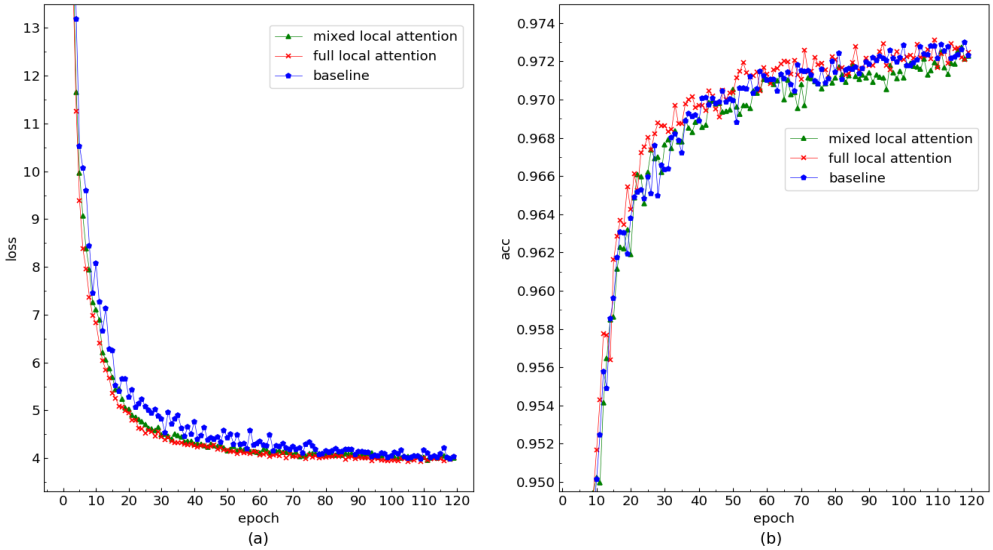

Figure 5. (**a**) Loss curve with dev-other. (**b**) Accuracy curve with dev-other.

Figure 5a shows that, after 120 training iterations, the loss value starts to oscillate, indicating that the neural network model converged after training. After adding local attention, the loss of the validation set is lower than that of the baseline throughout the training process, and the change in the loss curve is smoother than that of the baseline. This indicates that, when adding a local attention mechanism to the model, the model obtains a more accurate gradient descent direction during training.

From Table 3 and Figure 5b, we can observe that, compared to the baseline experiment, the hybrid local attention model shows a degraded performance. The dev-other error rate increases by 5.1%, the accuracy rate is significantly lower than that of the baseline system, and the experimental results are not as expected. It is conjectured that the distribution of the attention space with added a priori knowledge and the subspace without added a priori knowledge are different, and one projection matrix cannot transform these two representation vectors, which leads to the degradation of the model by adding half of the a priori knowledge.

5. Conclusions

In this paper, we studied the application of transformers for use in speech recognition and proposed an enhanced transformer based on a local attention mechanism called the LAS-Transformer. Specifically, we used depthwise separable convolution for subsampling, and there was no pooling layer in the subsampling layer to ensure the maximum integrity of position information at the embedding layer. Additionally, to compensate for the defects of the basic Transformer's local feature extraction, we proposed a local attention module to explicitly incorporate the highly correlated features of speech frames into the attention calculation, which can effectively extract local contextual information. Additionally, we replaced absolute positional embedding with relative positional embedding to improve the representation of location information, which calculates the self-attention score more accurately.

We conducted experiments on the LibriSpeech dataset; the experimental results demonstrate the effectiveness of the approach proposed in this paper.

In the future, we will improve the LAS-Transformer model in the following directions. LAS-Transformer is a non-streaming recognition method that requires a complete speech sequence for each speech recognition. Today, streaming speech recognition has better application prospects; thus, we will further improve the structure of the proposed model so that it can fulfill the needs of streaming speech recognition. Moreover, as the use of lightweight networks is becoming mainstream, we will further investigate the use of knowledge distillation to reduce the network parameters of the proposed model.

Author Contributions: Conceptualization, P.F. and D.L.; methodology, P.F. and D.L.; software, D.L.; validation, D.L.; data curation, D.L.; writing—original draft preparation, P.F. and D.L.; writing—review and editing, H.Y.; visualization, D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant NO. 61772048, Beijing Municipal Natural Science Foundation under Grant NO. 4153058.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Povey, D.; Peddinti, V.; Galvez, D.; Ghahremani, P.; Manohar, V.; Na, X.; Wang, Y.; Khudanpur, S. Purely sequence-trained neural networks for asr based on lattice-free MMI. *Proc. Interspeech* **2016**, *1*, 2751–2755. [CrossRef]
2. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
3. Eyben, F.; Wöllmer, M.; Schuller, B.; Graves, A. From speech to letters-using a novel neural network architecture for grapheme based ASR. In Proceedings of the Workshop on Automatic Speech Recognition & Understanding, Moreno, Italy, 13 November–17 December 2009; pp. 376–380.
4. Song, W.; Cai, J. End-to-end deep neural network for automatic speech recognition. *Standford CS224D Rep.* **2015**, *1*, 1–8.
5. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 173–182.
6. Graves, A. Sequence transduction with recurrent neural networks. *arXiv* **2012**, arXiv:1211.3711.
7. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
8. Rao, K.; Sak, H.; Prabhavalkar, R. Exploring Architectures, Data and units for streaming end-to-end speech recognition with RNN-Transducer. In Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, Japan, 16–20 December 2017; pp. 193–199.
9. Zeyer, A.; Merboldt, A.; Michel, W.; Schlüter, R.; Ney, H. Librispeech transducer model with internal language model prior correction. *arXiv* **2021**, arXiv:2104.03006.
10. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. In Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; pp. 577–585.
11. Bahdanau, D.; Chorowski, J.; Serdyuk, D.; Brakel, P.; Bengio, Y. End-to-end attention-based large vocabulary speech recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4945–4949.
12. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, Attend and Spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964.
13. Weng, C.; Cui, J.; Wang, G.; Wang, J.; Yu, C.; Su, D.; Yu, D. Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition. *Proc. Interspeech* **2018**, *1*, 761–765.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
15. Dong, L.; Xu, S.; Xu, B. Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5884–5888.
16. Karita, S.; Soplin, N.E.Y.; Watanabe, S.; Delcroix, M.; Ogawa, A.; Nakatani, T. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In Proceedings of the Interspeech 2019, ISCA, Graz, Austria, 15–19 September 2019; pp. 1408–1412.
17. Karita, S.; Chen, N.; Hayashi, T.; Hori, T.; Inaguma, H.; Jiang, Z.; Someki, M.; Soplin, N.E.Y.; Yamamoto, R.; Wang, X. A comparative study on transformer vs rnn in speech applications. In Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU), Sentosa, Singapore, 14–18 December 2019; pp. 449–456.

18. Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y. Conformer: Convolution-augmented transformer for speech recognition. *arXiv* **2020**, arXiv:2005.08100.
19. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, arXiv:1806.01261.
20. Guo, Q.; Qiu, X.; Liu, P.; Shao, Y.; Xue, X.; Zhang, Z. Star-Transformer. *arXiv* **2019**, arXiv:1902.09113.
21. Hinton, G.E.; Krizhevsky, A.; Wang, S.D. Transforming auto-encoders. In Proceedings of the International Conference on Artificial Neural Networks, Espoo, Finland, 14–17 June 2011; pp. 44–51.
22. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
23. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An asr corpus based on public domain audio books. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5206–5210.
24. 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.
25. Xiong, R.; Yang, Y.; He, D.; Zheng, K.; Zheng, S.; Xing, C.; Zhang, H.; Lan, Y.; Wang, L.; Liu, T. On layer normalization in the Transformer architecture. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 10524–10533.
26. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv* **2019**, arXiv:1901.02860.
27. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv* **2019**, arXiv:1904.08779.
28. Ko, T.; Peddinti, V.; Povey, D.; Khudanpur, S. Audio augmentation for speech recognition. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.
29. Kriman, S.; Beliaev, S.; Ginsburg, B.; Huang, J.; Kuchaiev, O.; Lavrukhin, V.; Leary, R.; Li, J.; Zhang, Y. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6124–6128.
30. Wang, Y.; Mohamed, A.; Le, D.; Liu, C.; Xiao, A.; Mahadeokar, J.; Huang, H.; Tjandra, A.; Zhang, X.; Zhang, F. Transformer-based acoustic modeling for hybrid speech recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6874–6878.