Article

An Improvement to Conformer-Based Model for High-Accuracy Speech Feature Extraction and Learning

Mengzhuo Liu and Yangjie Wei *

College of Computer Science and Engineering, Northeastern University, Wenhua Street 3, Shenyang 110819, China; 2071873@stu.neu.edu.cn
* Correspondence: weiyangjie@cse.neu.edu.cn

Abstract: Owing to the loss of effective information and incomplete feature extraction caused by the convolution and pooling operations in a convolution subsampling network, the accuracy and speed of current speech processing architectures based on the conformer model are influenced because the shallow features of speech signals are not completely extracted. To solve these problems, in this study, we researched a method that used a capsule network to improve the accuracy of feature extraction in a conformer-based model, and then, we proposed a new end-to-end model architecture for speech recognition. First, to improve the accuracy of speech feature extraction, a capsule network with a dynamic routing mechanism was introduced into the conformer model; thus, the structural information in speech was preserved, and it was input to the conformer blocks via sequestered vectors; the learning ability of the conformed-based model was significantly enhanced using dynamic weight updating. Second, a residual network was added to the capsule blocks, thus, the mapping ability of our model was improved and the training difficulty was reduced. Furthermore, the bi-transformer model was adopted in the decoding network to promote the consistency of the hypotheses in different directions through bidirectional modeling. Finally, the effectiveness and robustness of the proposed model were verified against different types of recognition models by performing multiple sets of experiments. The experimental results demonstrated that our speech recognition model achieved a lower word error rate without a language model because of the higher accuracy of speech feature extraction and learning using our model architecture with a capsule network. Furthermore, our model architecture benefited from the advantage of the capsule network and the conformer encoder, and also has potential for other speech-related applications.

Keywords: end-to-end; Chinese speech recognition; capsule network; conformer; bi-transformer

1. Introduction

Accurate feature extraction from target speech signals is significant to speech processing tasks. As compared with traditional speech processing technologies, a deep learning framework exhibits the advantages of strong modeling capability, fast training speed, high learning accuracy, and high robustness. Therefore, in recent years, it has become one of the most commonly used methods in the speech processing field, such as speech enhancement [1], speech separation [2,3], speaker recognition [4], and speech recognition [5–8].

Earlier methods for speech feature extraction and learning under the framework of deep learning are mostly based on recurrent neural networks (RNNs), especially long short-term memory (LSTM) [9]. However, the limitations of RNNs, such as gradient vanishing and gradient exploding [10], influence the performance of these models based on RNNs. To address these problems, convolutional neural networks (CNNs) have been proposed [9,11]. However, there are also some limitations of CNN-based models in practical applications, including a large number of parameters, more data for learning, and less expressive ability. To improve the performance of CNNs, a series of new networks have been proposed: the AlexNet network first successfully applied many operations in CNNs, such as rectified
linear units, dropout, and local response normalization, except for overlapping maximum pooling, novel data enhancement, and computing acceleration with a general processing unit [12]; VGGNet replaced the larger convolutional kernels in AlexNet with several consecutive $3 \times 3$ convolutional kernels and used $1 \times 1$ convolutional kernels for operations to improve the recognition performance of networks [13]; GoogLeNet realized high-speed computations using dense matrices [14]. These networks have improved the performance of CNNs from different aspects; however, their basic theory is still based on convolutional networks for exploration and optimization. Therefore, if they are used in speech-related tasks, there is a risk of losing valuable information in speech due to the maximum pooling operation in convolutional networks, thus, reducing the accuracy and speed of feature extraction and model learning.

As compared with CNNs, the basic theory of capsule networks is obviously different because they can reasonably preserve the detailed structure information throughout the network via sequestered vectors, rather than being lost and then recovered [15]. Owing to this advantage, they are widely used in image recognition [16], and recently, they have been introduced into speech emotion recognition [17]. Alternatively, benefiting from direct connection of arbitrary pairs of positions and easily parallelized computation, self-attention network has demonstrated promising results in a variety of natural language processing tasks [18–20]. Especially, the transformer model, which leverages multi-head attention and interleaves with feedforward layers, has been used for speech recognition with notable exceptions in recent years [21–23]. However, models with self-attention are less capable of extracting fine-grained local feature patterns. Therefore, the conformer encoder architecture, as shown in Figure 1, was proposed by Google Inc. to achieve very high-performance speech recognition, and is considered to be a convolution-augmented transformer [24].

Figure 1. Conformer encoder model architecture [24].

As shown in Figure 1, in the conformer encoder model, a convolution subsampling network is commonly used to downsample or extract shallow features of speech signals for the conformer blocks. To improve the robustness in feature extraction that results from the local filtering and max-pooling techniques in the convolution network, a capsule network was introduced to replace the convolution subsampling network, and then, an end-to-end speech recognition model was proposed based on that architecture in this study. As compared with traditional speech recognition models, our proposed model is novel in
several aspects and provides a new model architecture for speech feature extraction and learning. First, the convolution subsampling network in the original conformer encoder architecture is replaced by a capsule network. Thus, the structural information in speech can be preserved and input to the conformer blocks via sequestered vectors. Second, to improve the training speed of the capsule network, a layer of the residual network was added to deal with the degradation problem resulting from the increase in network layers. Finally, experiments with different architectures of end-to-end speech recognition models were conducted, and the results demonstrated that the proposed model architecture could obtain lower word error rates (WERs) with high efficiency. Furthermore, because our model architecture is able to improve the accuracy and speed of speech feature extraction and learning based on the conformer model, it also has potential for other speech-related applications.

The contents of this study are organized as follows: Firstly, in Section 2, the structure and information exchange equations in capsule networks are analyzed; secondly, a new end-to-end speech recognition model is proposed in Section 3; subsequently, in Section 4, the experimental results and error analysis based on different speech recognition models are given; finally, in Section 5, we provide the conclusions of this study.

2. Capsule Networks

Capsule networks are commonly used in image recognition because they can effectively solve problems in CNNs, such as indeterminate relationships between images and semantics and the lack of spatial layering or inference capability. The capsule network includes four parts: convolutional, primary capsule, digital capsule, and final output layers, as shown in Figure 2. The convolutional layer, as the starting point of the entire network, is mainly used to extract the low-level features of a target for preprocessing. The primary capsule layer contains the capsule network, which is mainly used to store the vectors of low-level features, accept the basic features detected by the convolutional layer, and generate the combination feature matrices, which are then sent to the digital capsule layer for the acquisition of high-level features.

![Figure 2. Architecture of the capsule networks.](image)

The core of the digital capsule layer is the dynamic routing algorithm, which consists of four main steps:

(1) Multiply the input features by matrix $W$ to realize the affine transformation,

$$\mu_{ji} = W_{ij} \mu_i$$  \hspace{1cm} (1)

where $\mu_i$ represents the input vector of the $i$-th capsule in layer $l$, $W$ represents the relation weight matrix between layer $l$ and layer $(l + 1)$, and $\mu_{ji}$ represents the vector of the $j$-th capsule in layer $(l + 1)$ after affine transformation.

(2) Calculate the coupling coefficient vector $c$,

$$c'_{ij} = \frac{e^{b'_{ij}}}{\sum_k e^{b'_{ik}}}$$  \hspace{1cm} (2)
where $c'_{i,j}$ represents the coupling coefficient after the $r$-th iteration, $k$ is the number of captures in layer $(l + 1)$, $b'_{i,j}$ represents the aggregation result from the $i$-th capsule to the $j$-th capsule after the $r$-th iteration, and $b^0_{i,j}$ is zero.

(3) Calculate the weighted sum of the vectors after the $r$-th iteration,

$$s'_{j} = \sum_i c'_{i,j} \mu_{j|i}$$  \hspace{1cm} (3)

(4) Calculate the output vector of the $j$-th capsule in layer $(l + 1)$ using the squash nonlinear activation function,

$$v'_{j} = \frac{||s'_{j}||^2}{1+||s'_{j}||^2} \times \frac{s'_{j}}{||s'_{j}||}$$  \hspace{1cm} (4)

If the termination condition of the dynamic routing algorithm (normally the iteration number is set to three) is satisfied, the iteration stops and the feature extraction completes; $v_j$ is the output of the $j$-th capsule in layer $(l + 1)$, and the magnitude of the output vector represents the probability of belonging to a class. Otherwise, the weight parameter $b$ is updated and return to Step 2 for the next iteration,

$$b'_{ij} = b_{ij} + \mu_{j|i}\nu_j$$  \hspace{1cm} (5)

From Figure 1, it can be observed that capsule networks have a shallow CNN layer and novel digital capsule and main capsule layers inside, so that the CNN layer can effectively extract the spatial and directional information of a speech signal based on the shallow features, such as the ability to obtain the location information of the pitch on the time and frequency axes, as well as the deep semantic logic information and the directional information of the bidirectional encoding [17]. Furthermore, capsule networks can use vectors to represent the features of speech signals, which can better classify the features and make parameter updating more accurate, therefore, they require less training data to achieve better training results as compared with other network structures [25]. However, current capsule networks have a small number of layers, the number of local parameters in each layer is very large, which easily causes the classification weights to be updated faster in the layers near the output and slower in the layers near the input in the back propagation, thus, the training speed of the entire network will be affected. Therefore, how to improve the training efficiency is a difficult problem in the applications of capsule networks. In the following section, we propose a method to solve this problem when we use the capsule network in speech recognition.

3. Improved Conformer-Based Model Architecture Based on the Capsule Network

In this study, the convolution subsampling network, shown in Figure 1, was replaced by a capsule network to downsample or extract shallow features of speech signals for the conformer blocks, and therefore, improve the accuracy of speech feature extraction. Then, based on this new model architecture that combined a capsule network and a conformer encoder, an end-to-end speech recognition model was proposed, which is shown in Figure 3. The model includes three core modules: the CapsNet blocks, conformer blocks, and bi-transformer decoder. The following describes our model according to its workflow.
Figure 3. Architecture of our proposed speech recognition model.

3.1. Encoder

The speech features are extracted as input for our model. After data enhancement using SpecAugment [26], the shallow features are extracted by the basic CNN blocks as input to the CapsNet blocks in the capsule network. In this study, the CapsNet blocks consist of two internal layers: primary and digital capsule layers. The primary capsule layer spreads the speech data into multiple capsules, and then integrates them into a matrix, which is sent to the digital capsule layer for the acquisition of high-level features. The digital capsule layer follows the dynamic routing algorithm in Equations (1)–(5), and the magnitude of the output vector is the probability that the speech signal belongs to a category.

First, to improve the training speed of the capsule network, we added a layer of residual network to the CapsNet blocks, as shown in Figure 4, because a residual network can deal with the degradation problem resulting from the increase in network layers [26]. After the introduction of a shortcut connection in the original network, our model allows data to flow across layers; thus, the initial learning goal of the network is changed, and the learning difficulty is reduced.
Figure 4. Combination of the capsule and residual networks.

Assuming the input and output dimensions of the nonlinear unit in the neural network are the same, the function $Z$ to be fitted within the neural network unit can be decomposed into two parts, given as:

$$Z(x) = F(x) + x$$

where $F$ represents the model function of the CapsNet blocks and $x$ represents the system input.

Inside the network, the constant mapping $Z(x) → x$ is learned, that is, the residual part $F(x)$ converges to 0. This is equivalent to replacing the learning target with $Z(x)-x$, leading the whole structure to converge in the direction of identity mapping. This transformation can speed up the training of the capsule network and simplify the parameter optimization process. Then, $Z$ is linearized to integrate the features and change the dimension, and the vector $x$ is processed in the dropout layer to avoid over fitting:

$$x = \text{Dropout} (\text{Linear}(Z))$$

Finally, in this study, inputs, i.e., $x$, to the conformer module enhance the performance of the encoder. Four modules are included in the conformer blocks: the feedforward, multi-head self-attention [27], convolution, and layer normalization modules; the specific structure is shown in Figure 5.

Figure 5. Diagram of the conformer blocks.

After feeding $x$ into the conformer blocks, output $y$ is fed through the feedforward module FFN, the multi-head self-attentive module MHSA, the convolutional module Conv, and the layer normalization module Layernorm. The specific process is as follows:

$$x' = x + \frac{1}{2} \text{FFN}(x)$$

$$x'' = x' + \text{MHSA}(x')$$

$$x''' = x'' + \text{Conv}(x'')$$

$$y = \text{Layernorm}(x''' + \frac{1}{2} \text{FFN}(x'''))$$

3.2. Decoder

Because a bi-transformer model can give output as a left-to-right target sequence and a right-to-left target sequence with the advantage of better attention and more effective
utilization of contextual information, our system uses the bi-transformer model as the decoder [28,29]. This decoder superimposes $N$ sublayers into a multi-head attention model and feedforward network, and adds residual connections between the inputs and outputs of the sublayers. The bi-transformer decoder comprises two single weight-sharing decoders: one decodes from left to right to generate the forward context, and the other decodes from right to left to generate the backward context. The input information of the bi-transformer is obtained from the mapping at the output end of the encoder. Suppose that the output of the conformer is $Y$, through matrix multiplication, it can be mapped into three matrices $K$, $Q$, and $V$. Then, the forward and backward vectors are spliced to obtain a two-way dot product attention layer, and the attention scores $H$ are calculated as:

$$H = \text{Attention}(\overleftarrow{Q}, \overleftarrow{K}, \overleftarrow{V})$$

Similarly, the multi-head attention value can be obtained by splicing:

$$\text{Multihead}\left(\begin{bmatrix} \overleftarrow{Q} \\ \overleftarrow{K} \\ \overleftarrow{V}\end{bmatrix}, \begin{bmatrix} \overrightarrow{Q} \\ \overrightarrow{K} \\ \overrightarrow{V}\end{bmatrix}\right) = \text{Concat}\left(\begin{bmatrix} \overleftarrow{H}_1 \\ \overrightarrow{H}_1\end{bmatrix}, \ldots, \begin{bmatrix} \overleftarrow{H}_h \\ \overrightarrow{H}_h\end{bmatrix}\right) W^O$$

where $W^O$ is the trainable weight matrix, Concat is the concatenate operation, $h$ is the number of heads.

Because the decoder is bidirectional, the decoding process is performed in both directions using the magnitude of the beam search value, and it stops when a termination symbol is predicted. The search process is as follows: At each time step, the alive hypothesis statement consists of two candidate statements guided by the special character $<L2R>$, where two candidate statements guided by the special character $<R2L>$, and the score of the completed sentence determines the best generated sequence. If the highest score is guided by the right-to-left sequence, the final predicted sequence must be reversed. This search method effectively solves the problem that the traditional transformer model can only decode in one direction, therefore, the consistency between the different directional hypotheses is improved. The loss function of the bi-transformer is defined as:

$$L = \alpha \cdot L_{l2r} + (1 - \alpha) \cdot L_{r2l}$$

where $\alpha$ is the inversion coefficient to balance the weight of the decoding order, and $L_{l2r}$ and $L_{r2l}$ represent the scores of the left-to-right and right-to-left decoding, respectively. In our system $\alpha = 0.5$.

3.3. Decoding

The purpose of this study was to design an offline speech recognition model architecture that does not contain a language model. Therefore, there is a high requirement for accuracy and stability, as well as trainability and decoding performance. The decoding framework in this study uses the U2 model with a two-channel joint connectionist temporal classification/attention-based encoder–decoder (CTC/AED) [30]; its structure is shown in Figure 6. The framework is derived from the traditional joint CTC-attention framework and includes three modules: CTC, attention, and shared encoders, where the connection between the CTC decoder and attention decoder is strengthened by the rescoring method. The CTC decoder consists of a linear layer that is used to transform the intermediate output of the encoder into the final output and complete the decoding using the attention rescoring method. Specifically, the CTC prefix beam search algorithm generates the $N$-best intermediate candidates, and then the attention decoder is used to jointly rescore these candidates with a shared encoder to obtain a more accurate final output, as shown in Figure 7. In addition, the U2 model uses the chunk training method to limit attention modeling. The principle is that the distribution of chunk sizes dynamically changes during the training process, and the attention model captures different information of various chunk sizes and
learns how to make accurate predictions in different limited contexts to achieve a trade-off between decoding speed and accuracy.

The basic problem of speech recognition is to map a speech feature sequence to a word sequence, which can be solved by the Bayesian decision theory and probability function estimation theory [31,32]. First, the word sequence with the highest probability is estimated as follows:

$$Z^* = \arg\max_{Z \in V^*} (p(Z|X))$$

where $Z^* = \{z_n \in V | n = 1, \ldots, N\}$ is the estimated word sequence with the highest probability in all the possible word sequences, $X = \{x_t \in \mathbb{R}^D | t = 1, \ldots, T\}$ is the input speech feature sequence, $x_t$ is the $D$ dimensional speech feature vector in the $t$-th frame, and $z_n$ the word at position $n$ of the vocabulary $V$.

According the Bayesian theory, $p(Z|X)$ is transformed into:

$$p(Z|X) = \frac{p(X|Z) \cdot p(Z)}{p(X)}$$

Because in the decoding process, the input $X$ does not change, $p(X)$ is neglected. Therefore, Equation (16) is simplified as follows:

$$Z^* = \arg\max_{Z \in V^*} (p(X|Z) \cdot p(Z))$$

where $p(Z)$ is the probability of the output word sequence, which is described by the language model and $p(Z|X)$ is the likelihood probability, characterized by the acoustic model.

Because the input audio frame is only about 20–30 ms and its information cannot cover a word or even a character, the general output modeling unit is a phoneme. The CTC method defines a path sequence $C = \{c_t | t = 1, \ldots, T\}$, where both the non-blank label and blank label exist, and the probability of the whole path sequence is composed of the label probability of each frame:

$$p(C|X) = \prod_{t=1}^{T} y_{ci}$$

where $y_{ci}$ denotes the posterior probability of path $c_i$ in the $t$-th frame.
Define $L$ as a label element table, the augmented letter sequence is defined with the blank symbol $b$ as follows:

$$L' = L \cup \{b\}$$ (19)

Define $\Phi: L \leq T \rightarrow L' T$, and map the label sequence $Z$ to the path sequence $C$. Then, the probability of label $Z$ is:

$$p(Z|X) = \sum_{C \in \Phi(Z)} p(C|X)$$ (20)

The factorization of the posterior probability $p(C|X)$ is:

$$p(C|X) = \sum_z p(C|Z, X) p(Z|X) \approx \sum_z p(C|Z) p(Z|X)$$ (21)

In the CTC method, $p(C|X)$ is obtained using the conditional independence hypothesis, which can simplify the dependence of acoustic model and alphabetic model.

As compared with the CTC, the attention method estimates $p(C|X)$ based on the rules of probability chain without any hypothesis [33] as follows:

$$p(C|X) = \prod_{l=1}^{L} p(c_l|c_1, \ldots, c_{l-1}, X) \triangleq p_{at}(C|X)$$ (22)

where $p_{at}(C|X)$ is an object function based on attention, $p_{at}(c_l|c_1, \ldots, c_{l-1}, X)$ is calculated as follows:

First, the input speech feature is encoded as:

$$h_t = \text{Encoder}(X)$$ (23)

The weight of attention is given as:

$$a_{lt} = \begin{cases} 
\text{Contentattention}(q_{l-1}, h_l) \\
\text{Locationattention}\{a_{l-1}\}_{T-1}^{T}, q_{l-1}, h_l
\end{cases}$$ (24)

where $a_{lt}$ is the weight of attention, which denotes the soft connection $h_l$ of hidden vectors; Contentattention () and Locationattention () denote the content-based attention mechanism with and without convolution features, respectively.

For each output, the hidden vector of the lexical sequence is calculated as:

$$r_1 = \sum_{l=1}^{T} a_{lt} h_l$$ (25)

Then, the decoder network including a recurrent network based on the prior output $c_{l-1}$ and hidden vector $q_{l-1}$, except for the hidden vector $r_1$, is used to calculate $p(t|c_1, \ldots, c_{l-1}, X)$ as follows:

$$p(t|c_1, \ldots, c_{l-1}, X) = \text{Decoder}(r_l, q_{l-1}, c_{l-1})$$ (26)

In our training process, the CTC is combined with attention-based cross entropy [34,35], and a multi-objective learning framework is adopted to improve robustness. The loss function is denoted as:

$$L(S) = \lambda L_{CTC}(S) + (1 - \lambda) L_{AED}(S)$$ (27)

where $L_{CTC}$ and $L_{AED}$ are the loss functions of the CTC and attention decoders, respectively; $\lambda$ is an adjustable parameter, which satisfies $0 \leq \lambda \leq 1$; $S$ denotes the training collection.
The $L_{\text{CTC}}$ is calculated as:

$$L_{\text{CTC}} = -\ln \prod_{(X,Z^*) \in S} p(Z^*|X)$$

(28)

where $Z^*$ represents the correct label value, $p(Z^*|X)$ denotes the probability of obtaining the correct label corresponding to given $X$, which can be calculated using the forward–backward algorithm as follows:

$$p(Z^*|X) = \sum_{u=1}^{|y'|} \alpha_t(u) \frac{\beta_t(u)}{q_t(c_u')}$$

(29)

where $c_u'$ represents all the possible prefix paths ending with $u$, $\alpha_t(u)$ represents the total probability of all possible prefix paths ending with the $u$-th label, $\beta_t(u)$ is the total probability of all possible suffixes starting from the $u$-th label, and $q_t(c_u')$ represents the output value of $c_u'$ at time $t$ after activation by the softmax function.

The $L_{\text{AED}}$ is calculated as:

$$L_{\text{AED}} = -\sum_u \ln p(Z_u^*|X,Z_{1:u-1}^*)$$

(30)

where $Z_{1:u-1}^*$ represents the correct label before the current character $u$.

In summary, the flexible alignment of CTC accelerates the training process of the network, and the fusion of the CTC and attention decoder loss functions significantly simplifies the training channels and speeds up the model training process; therefore, training of the multiple branches of the encoder can be realized in speech recognition.

4. Experiment
4.1. Experimental Setup

To validate the model proposed in this study, a series of experiments were conducted on the open-source dataset AISHELL-1 which is an open-source Mandarin Chinese corpus comprising a 150-h training set, 10-h validation set, and a 5-h test set [36]. Furthermore, it has 4235 output units, including 4230 Chinese characters and five additional tokens [37]. In our experiments, the 80-dimensional Meier filter bank was used to compute the real-time acoustic features via Torchaudio, where the window length was 25 ms and the frame shift was 10 ms. In addition, two masks with the maximum frequency ($F = 10$) and maximum time ($T = 50$) were applied to effectively mitigate overfitting in SpecAugment.

For the model parameters, we used a 12-layer conformer and transformer for encoding, where the feedforward position encoding dimension, model dimension, and number of attention heads were 2048, 256, and 4, respectively. The decoding side used a six-layer transformer and bi-transformer for decoding. For the optimizer, the Adam optimizer and the warmup strategy were introduced. The learning rate was set to 25,000, and the feedforward dropout was set to 0.1 to prevent overfitting. The batch size was set to 15, and the most suitable model was searched for through 150 rounds of training. In addition, to ensure the robustness of our model, the top five best models with a low loss in the development set during training were averaged to achieve the final model. The experimental system was Centos 7.9 (64 bits), the deep learning neural network framework training and testing software was Pytorch, and the GPU device model was Tesla T4 with a running memory of 15 GB.

To measure the accuracy of different models, the WER was used as an evaluation index, which can be expressed as:

$$\text{WER} = (S + D + I)/N \cdot 100\%$$

(31)

where $S$ is character substitution, $D$ is the deletion error, $I$ is the insertion error, and $N$ is the total number of words in the corresponding text of the original speech.
4.2. Performance Comparison and Ablation Analysis

4.2.1. Comparisons of Different Encoders and Decoders

First, a series of cross-sectional comparison experiments were conducted to verify the accuracy of the proposed model. The models introduced for comparison included the following:

1. Baseline models, i.e., compute library for deep neural network (CLDNN) with the structure of CNN × 2 + ResconvLSTM × 8 + DNN + CTC in [7], and LAS in [6];
2. Models based on the capsule network combined with different encoders and decoders (Caps denotes the capsule network, C denotes the conformer, B denotes the bi-transformer, T denotes the transformer); i.e., Caps-CB, Caps-CT, Caps-TT, and Caps-TB;
3. Models with the original convolution subsampling network, i.e., CB, CT, TT, and TB.

In the experiments, the complete AISHELL-1 corpus was used for training, validation, and testing, and no pretrained model or language model was applied in the training. The WER and model parametric number were both calculated and compared, and the experimental results are shown in Table 1, from which the following conclusions can be obtained:

1. Without the capsule network, the WER of the CB system that combined the conformer encoder and bidirectional transformer decoder was the lowest (6.21%). The WER of the TT system using only the transformer model was 7.5%, while the WER of the CT model using the conformer encoder and transformer decoder was 6.82%. That means, the recognition accuracy improved after the conformer model was used because of the encoder’s ability to incorporate multi-head attention global modeling and convolutional small-range feature extraction. Similarly, the WER of the TB system using the bidirectional transformer decoder was 6.55%, which was 0.95% lower than that of the TT system using the unidirectional transformer decoder. This is because the bidirectional transformer can fully use the contextual information, and decoding in both directions extends the modeling possibilities.
2. After introducing the capsule and residual networks, the WER of the proposed Caps-CB model was the smallest (5.97%) under the same testing conditions; as compared with the models in [6] and [7], it decreased by 11.62% and 7.23%, respectively. The WERs of the Caps-TT, Caps-CT, and Caps-TB models were 7.31%, 6.36%, and 6.29%, respectively, which decreased by 0.19%, 0.46%, and 0.26%, respectively, as compared with the models without the capsule network. For our Caps-CB model, the WER decreased by 0.24% as compared with that of the CB model. This implies that the capsule network can enhance the feature extraction ability of these speech recognition models and can effectively extract deep features of speech signals, and also shows that the features in vector form are better than those in scalar form, which can enhance the performance of the encoder, and the capsule network can be well integrated with speech systems and applied to the field of speech recognition.
3. For the number of parameters, without the capsule network, the number of parameters of the CB system was the largest, with a size of 50.0 M, whereas that of the TB model was the smallest, which was 33.8 M. Therefore, we can conclude that the addition of the encoder improved the recognition accuracy of the model, but increased the parameter number. As a result, the parameter number of these various models with the capsule network increased by approximately 2 M relative to the corresponding models without the capsule network. The ability to further improve the performance of the speech recognition system despite the small increase in the number of participants also demonstrates the power of the capsule network.
Table 1. Comparison of speech recognition models with different encoder and decoders.

| Models                        | WER (%) | Parameter Quantity (M) |
|-------------------------------|---------|------------------------|
| CNN × 2 + ResconvLSTM × 8 + DNN + CTC | 17.59   | -                      |
| LAS                           | 13.20   | -                      |
| CB                            | 6.21    | 50.0                   |
| CT                            | 6.82    | 47.8                   |
| TT                            | 7.50    | 33.8                   |
| TB                            | 6.55    | 35.8                   |
| Caps-TB                       | 6.29    | 37.7                   |
| Caps-TT                       | 7.31    | 35.5                   |
| Caps-CT                       | 6.36    | 49.7                   |
| Caps-CB                       | 5.97    | 51.9                   |

Then, to observe the convergence state of model training, the loss function curves of the CT, Caps-CT, CB, and proposed Caps-CB models with an increasing number of iterations were drawn, and the results are shown in Figure 8, where the horizontal axis represents the number of iterations and the vertical axis represents the loss function value.

![Figure 8](image)

**Figure 8.** The loss function curves during training of different models.

As can be observed from Figure 8, the loss function values of all models decrease significantly with an increase in training time, and the convergence speed of the Caps-CB model proposed in this study is significantly faster when the iteration number is more than 40. When the iteration number reaches 150, the loss function value of the Caps-CB model is approximately 50% of that of the Caps-CT model, followed by the CB model whose loss function value is approximately 75%. This implies that the combination of the conformer encoder and bidirectional transformer decoder can effectively improve the convergence speed, and after the subsampling network in the conformer is replaced by the capsule network, the convergence speed can be greatly increased.

4.2.2. Combination with the Performer Encoder

It is well known that the training speed of the performer model is faster as compared with that of the conventional transformer [38], because it uses the FAVOR+ mechanism to construct an unbiased estimator for the attention mechanism matrix, resulting in a linear
increase in resource requirements and a significant reduction in space and time complexity. In this experiment, we replaced the conformer encoder of our model with a performer, remaining the capsule network as the subsampling network, and the new model was called Caps-PB. Then, we tested the recognition performance and parameter quantity of the Caps-PB model and compared it with the PB, CB, and Caps-CB models. Table 2 presents the results.

Table 2. Performance comparisons of the recognition models based on the performer encoder.

| Model   | Enc-Dec-Dm-Head | WER (%) | Parameter Quantity (M) |
|---------|-----------------|---------|------------------------|
| Caps-PB | 12-6-256-4      | 8.01    | 36.1                   |
| Caps-CB | 12-6-256-4      | 5.97    | 51.9                   |
| PB      | 12-6-256-4      | 8.24    | 34.2                   |
| CB      | 12-6-256-4      | 6.21    | 50.0                   |

From Table 2, we can observe that the WER of the Caps-PB model decreases by 0.23% as compared with the PB system, which means that enrollment of the capsule network can improve the recognition performance. The numbers of parameters of Caps-PB and PB are both lower than those of our model and CB because of the advantage of the performer; however, the WER of the Caps-PB model increases by 2.04% and 1.8% as compared with that of the Caps-CB and CB models, respectively. The reason for performance degradation of these models with the performer may be that the performer model is more effective for handling longer sequences (length > 5000) but less effective when dealing with shorter sequences. In the future, we plan to continue to explore new datasets that are more suitable for the Caps-PB model and to explore the performance of the capsule network on different datasets.

4.3. Analysis of Some Important Parameters

4.3.1. Influence of Parameter Quantity

From Table 1, we can observe that the recognition WER of the Caps-CB model decreases by 0.24% relative to the CB model; however, the parameter quantity also increases by nearly 2 M. To explore whether the performance improvement was related to an increase in parameters, the intrinsic relationship between the number of parameters and performance was explored. First, we increased the number of parameters in the downsampling layer of the original CB model using layer superposition until it was as close as possible to that of the proposed Caps-CB model. Here, the enlarged CB model is referred to CB_enlarged. Then, we calculated the WERs of these models, and the experimental results are listed in Table 3. From this table, we can observe that the WER of the enlarged CB model is 0.03% lower than that of the original CB model, whereas it is still higher than that of our proposed Caps-CB model, although its parameter number increases by 1.95 M. These results show that the recognition ability of the convolutional network does not significantly improve after layer stacking, which does not change the working principle of the convolutional networks. This is also a good indication that the capsule network outperforms the CNNs correlation model with a novel change in terms of performance and is able to have good WER results with matching parameters.

Table 3. Performance improvement after parameter balance and analysis.

|                | CB     | CB_enlarged | Caps-CB |
|----------------|--------|-------------|---------|
| WER (%)        | 6.21   | 6.18        | 5.97    |
| Parameter quantity (M) | 50.00 | 51.95       | 51.94   |

4.3.2. Influence of Router Iterations

In this experiment, the number of dynamic routing iterations in a capsule network determines the effect on parameter updating, even when it is set to three in most image
recognition tasks using capsule networks; we tested the WER and the number of parameters of the Caps-CB model when the router iteration number gradually increased from two to six, and the results are shown in Table 4. As shown in Table 4, the number of parameters in our model does not change with an increase in the iteration number; however, there is a small decrease in the WER of the model. The WER is the lowest when the number of iterations is two. This is because the dynamic routing algorithm iterates only for parameter updating and optimization and does not expand the number of parameters. As the iteration number increases, the number of parameter updates, matrix multiplications, and complex operations during the training process increase, thus, slowing down the training rate of the capsule network and reducing the convergence degree of the network. Therefore, in practical speech recognitions, it is not necessary to increase the number of the dynamic router iterations more than two.

Table 4. Performance achieved with different router iterations in the capsule network.

|     | 2    | 3    | 4    | 5    | 6    |
|-----|------|------|------|------|------|
| WER (%) | 5.97 | 6.25 | 6.23 | 6.27 | 6.28 |
| Parameter quantity (M) | 51.9 | 51.9 | 51.9 | 51.9 | 51.9 |

4.3.3. Influence of the Beam Size

Because the attention rescoring method was adopted in our decoding process, to explore the model prediction performance, we calculated the WER curves of the Caps-CB, CB, Caps-CT and CT models when different beam sizes were chosen in the CTC search; the results are shown in Figure 9. It can be observed that the recognition performance of all models shows a significant improvement when the value of the beam size increases from 1 to 20, however, the WER curves tend to be flat when the beam size further increases, that is, the recognition performance gain of these models reaches a saturation state. This is because increasing the beam size can improve the diversity of the generated texts, but simultaneously, the similarity between the generated texts becomes higher. When the diversity of samples is not sufficiently high, the recognition accuracy of these model tends to be constant or even decrease. As compared with these models, the WER curve of the Caps-CB model proposed in this study decreased the fastest with increasing beam size, which was due to the fast convergence speed and good fitting effect of Caps-CB, and it did not decrease significantly even when the beam size was equal to 30, while the WER of the CT model decreased the slowest, followed by the Caps-CT and the CB models. This means that the bidirectional transformer can decode in both directions, which fully uses the contextual information and extends the modeling possibilities.

Therefore, when designing a speech-recognition model, it is necessary to select an appropriate beam size.

Finally, we fixed the value of the beam size to be 20, and compared the transcriptions obtained by different models with the same discourse in AISHELL-1. The results are shown in Table 5. All the models were trained using the full dataset, and the converged models were used for prediction. From Table 5, we can observe that the output of Caps-CB is more accurate and grammatically correct, whereas both the LAS and CB models have some error cases, such as the underlined parts in the table. These results prove that the Caps-CB model in this study has better generalization ability and practical performance. It also shows that the capsule network is more effective for extracting deep speech features and semantic logic, and therefore, to further exploit the location information of utterances. There are good results on smaller databases and the performance is able to outperform ordinary CNN layers.
Figure 9. WERs of different models with variable beam sizes.

Table 5. Transcription performance in Chinese with different models when the beam size was 20.

| Models       | Transcription                                           |
|--------------|---------------------------------------------------------|
| Truth        | 这令被贷款的员工们寝食难安                             |
| (a) LAS      | 这令被贷款的员工们寝室男安                             |
| (b) CB       | 这令被贷款的员工们请是男安                             |
| (c) Caps-CB  | 这令被贷款的员工们 寝食难安                             |
| Truth        | 按照扶贫扶大扶贫的原则                                |
| (a) LAS      | 按照福悠福大幅的扶贫的原则                             |
| (b) CB       | 按照富有扶大扶贫的原则                                |
| (c) Caps-CB  | 按照 扶贫 扶大扶贫 的 原则                              |

5. Conclusions
To address the problems of incomplete feature extraction and limited learning accuracy of existing speech processing models based on a conformer encoder, in this study, a method of using a capsule network to improve the accuracy of feature extraction based on a conformer encoder was studied, and then, an end-to-end speech recognition model was proposed. Our primary contribution is the introduction of the capsule network into the conformer model architecture to improve the speech feature learning accuracy with its powerful feature extraction ability. The second contribution is the simultaneous application of the residual network in the capsule blocks to reduce the training difficulty of the capsule network. Third, we built a series of similar architectures using the CTC/AED model, which significantly improved the training speed and stability, and conducted several comparison experiments on the AISHELL-1 dataset. The experimental results demonstrate that our Caps-CB model achieves a fast convergence speed and higher recognition accuracy. In addition, because our model does not require a language model, the modeling process is simple, and the error analysis results show that it can achieve speech recognition with high precision. Therefore, our model which combines a capsule network and a conformer encoder has the potential for other practical applications related to speech processing because of its high accuracy of speech extraction and learning.
Author Contributions: Conceptualization, M.L. and Y.W.; methodology, M.L.; software, M.L.; validation, M.L. and Y.W.; formal analysis, Y.W.; writing—original draft preparation, M.L.; writing—review and editing, Y.W.; funding acquisition, Y.W. 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, grant number 61973059.

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.

Acronyms

- AED: Attention-based Encoder-Decoder
- Bi-transformer: Bidirectional-Transformer
- CapsNet: Capsule Network
- CNN: Convolutional Neural Network
- CTC: Connectionist Temporal Classification
- LAS: Listen, Attend and Spell
- LSTM: Long Short-Term Memory
- MHSA: Multi-Head Self Attention Module
- RNN: Recurrent Neural Network
- VGGNet: Visual Geometry Group Network
- WERs: Word Error Rates

References

1. Michelsanti, D.; Tan, Z.H.; Zhang, S.X. An Overview of Deep-Learning-Based Audio-Visual Speech Enhancement and Separation. *IEEE Trans. Audio Speech Lang. Process.* 2021, 29, 1368–1396. [CrossRef]
2. Stoller, D.; Ewert, S.; Dixon, S. Wave-U-Net: A Multi-scale Neural Network for End-to-end Audio Source Separation. *arXiv* 2018, arXiv:1806.03185v1.
3. Luo, Y.; Mesgarani, N. Conv-TasNet: Surpassing Ideal Time–frequency Magnitude Masking for Speech Separation. *IEEE Trans. Audio Speech Lang. Process.* 2019, 27, 1256–1266. [CrossRef] [PubMed]
4. Li, C.; Ma, X.; Jiang, B. Deep Speaker: An End-to-End Neural Speaker Embedding System. *arXiv* 2017, arXiv:1705.02304.
5. Hanan, A.Y.; Mass, A.L.; Jurafsky, D.; Ng, A.Y. First-Pass Large Vocabulary Continuous Speech Recognition Using Bi-Directional Recurrent DNNs. *arXiv* 2014, arXiv:1408.2873.
6. Chan, W.; Jaitly, N.; Le, Q.; Vinayals, O. Listen, Attend and Spell: A Neural Network for Large Vocabularies. *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Shanghai, China, 20–25 March 2016; pp. 4960–4964.
7. Sun, S.N.; Zhang, B.B.; Xie, L. An Unsupervised Deep Domain Adaptation Approach for Robust Speech Recognition. *Neurocomputing* 2017, 257, 79–89. [CrossRef]
8. Abdel-Hamid, O.; Mohamed, A.R.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional Neural Networks for Speech Recognition. *IEEE Trans. Audio Speech Lang. Process.* 2014, 22, 1533–1545. [CrossRef]
9. Dahl, G.; Yu, D.; Deng, L.; Acero, A. Context-dependent Pre-trained Deep Neural Networks for Large-vocabulary Speech Recognition. *IEEE Trans. Audio Speech Lang. Process.* 2012, 20, 30–42. [CrossRef]
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 2017, 60, 84–90. [CrossRef]
11. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* 2014, arXiv:1409.1556.
12. Shan, C.; Zhang, J.; Wang, Y. Attention-Based End-to-End Speech Recognition on Voice Search. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4764–4768.
13. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic Routing Between Capsules. *arXiv* 2017, arXiv:1710.09829.
16. Chen, Y.; Zhao, J.; Qiu, Q. A Transformer-Based Capsule Network for 3D Part-Whole Relationship Learning. *Entropy* 2022, 24, 678. [CrossRef] [PubMed]

17. Wu, X.; Liu, S.; Cao, Y. Speech Emotion Recognition Using Capsule Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 13–18 May 2019; pp. 6695–6699.

18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7 June 2015; pp. 1–12.

19. Vaswani, A.; Shazeer, N.; Parmar, N. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Los Angeles, CA, USA, 4–7 December 2017; pp. 5998–6008.

20. Devlin, J.; Chang, M.-W.; Lee, K. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* 2018, arXiv:1810.04805.

21. Wu, Z.; Liu, Z.; Lin, J.; Lin, Y.; Han, S. Lite Transformer with Long-short Range Attention. In Proceedings of the International Conference on Learning Representations (ICLR), Online, 26–30 April 2020.

22. Lu, Y.; Li, Z.; He, D.K.; Sun, Z.; Dong, B.; Qin, T.; Wang, L.; Liu, T.-Y. Understanding and Improving Transformer from a Multi-Particle Dynamic System Point of View. In Proceedings of the World Conference on Artificial Intelligence (WAC), Shanghai, China, 29–31 August 2019.

23. 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 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019.

24. 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. In Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH), Shanghai, China, 25–29 October 2020; pp. 5036–5040.

25. Poncelet, J.; Hamme, H.V. Multitask Learning with Capsule Networks for Speech-to-Intent Applications. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 8494–8498.

26. Park, D.S.; Chan, W.; Zhang, Y. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH), Graz, Austria, 15–19 September 2019; pp. 2019–2680.

27. He, K.M.; Zhang, X.Y.; Ren, S.Q. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

28. Dai, Z.; Yang, Z.; Yang, Y. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *arXiv* 2019, arXiv:1901.02860v2.

29. Chen, X.; Zhang, S.; Song, D.; Ouyang, P.; Yin, S. Transformer with Bidirectional Decoder for Speech Recognition. *arXiv* 2020, arXiv:2008.04481.

30. Yao, Z.; Wu, D.; Wang, X. WeNet: Production Oriented Streaming and Non-streaming End-to-End Speech Recognition Toolkit. *arXiv* 2021, arXiv:2102.01547.

31. Graves, A.; Fernández, S.; Gomez, F. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with ReCurrent Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 369–376.

32. Helmenstine, A.M. *Bayes Theorem Definition and Examples*; ThoughtCo: New York, NY, USA, 2021.

33. Lafferty, J.; McCallum, A.; Pereira, F. Conditional random fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the International Conference on Machine Learning and Applications (ICML), Boca Raton, FL, USA, 16–19 December 2001; pp. 282–289.

34. Feng, J.; Wang, X.; Lu, D. *Probability Theory and Mathematical Statistics*; Higher Education Press: Beijing, China, 2012.

35. Rubinstein, R.Y.; Kroese, D.P. *The Cross-Entropy Method*; Springer: Berlin, Germany, 2004.

36. Chen, J.C. Design of zero reference codes using cross-entropy method. *Opt. Exp.* 2009, 17, 22163–22170. [CrossRef] [PubMed]

37. Bu, H.; Du, J.; Na, X. AISHELL-1: An Open-source Mandarin Speech Corpus and A Speech Recognition Baseline. In Proceedings of the Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases & Speech I/o Systems & Assessment (Oriental COCOSDA), Seoul, Korea, 1–3 November 2017.

38. Horomanski, K.; Likhoshchevstov, V.; Dohan, D. Rethinking Attention with Performers. *arXiv* 2020, arXiv:2009.14794.