En-HACN: Enhancing Hybrid Architecture With Fast Attention and Capsule Network for End-to-end Speech Recognition

Boyang Lyu, Chunxiao Fan, *Member, IEEE*, Yue Ming, *Member, IEEE*, Panzi Zhao, and Nannan Hu

***Abstract*—Automatic speech recognition (ASR) is a fundamental technology in the field of artificial intelligence. End-to-end (E2E) ASR is favored for its state-of-the-art performance. However, E2E speech recognition still faces speech spatial information loss and text local information loss, which results in the increase of deletion and substitution errors during inference. To overcome this challenge, we propose a novel Enhancing Hybrid Architecture with Fast Attention and Capsule Network (termed En-HACN), which can model the position relationships between different acoustic unit features to improve the discriminability of speech features while providing the text local information during inference. Firstly, a new CNN-Capsule Network (CNN-Caps) module is proposed to capture the spatial information in the spectrogram through the capsule output and dynamic routing mechanism. Then, we design a novel hybrid structure of LocalGRU Augmented Decoder (LA-decoder) that generates text hidden representations to obtain text local information of the target sequences. Finally, we introduce fast attention instead of self-attention in En-HACN, which improves the generalization ability and efficiency of the model in long utterances. Experiments on corpora Aishell-1 and Librispeech demonstrate that our En-HACN has achieved the state-of-the-art compared with existing works. Besides, experiments on the long utterances dataset based on Aishell-1-long show that our model has a high generalization ability and efficiency.**

***Index Terms*—Automatic speech recognition, capsule network, conformer, end-to-end, fast attention mechanism.**

I. Introduction

Automatic speech recognition (ASR) technology has wide applications in autonomous driving, smart homes, mobile phones, and so on. With the development of deep learning technology, E2E models are favored in ASR due to their simplified structure and advantaged recognition accuracy. Several E2E models have been proposed, such as connectionist temporal classification (CTC) [1], [2], attention-based encoder-decoder models [3], [4], recurrent neural network transducer (RNN-T) [5], [6], hybrid CTC/attention model [7], [8], and Transformer/Conformer [9], [10]. Currently, the Transformer/Conformer [11], [12] based on self-attention has achieved state-of-the-art performance due to it can capture global information without considering the distance among speech frames.

To capture the rich raw speech information contained in the spectrogram, the self-attention-based E2E models utilize convolution for temporal subsampling at the input layer has become a common technique [9], [10], [11], [12], [14], [15], [16]. Huang et al. [14] added interleaved convolution before the unidirectional Transformer to reduce the speech frame-rate, where adjacent speech frames can be formed into a chunk to represent more meaningful units such as phonemes. Recently, Wang et al. [15], [16] utilized a VGG-like convolutional block instead of convolutional subsampling to reduce the frame-rate of speech and obtain better recognition results. The input layer utilizes convolutional subsampling to capture unit information like phonemes in the spectrogram. While convolutional subsampling can capture useful unit information, CNN operations ignore the spatial information between features in the spectrogram, which contains position relationships of low-level features like pitch and formant frequency. As illustrated in Fig. 1, the lack of spatial information between features will reduce the distinguishability of speech features, resulting in the occurrence of "in it" substitution errors for similar pronunciations.

On the other hand, while self-attention-based models are good at capturing global information, the weighted average operation in self-attention disperses the attention distribution, ignoring the relationships between neighboring signals [13]. To combat this issue, various works [10], [17], [18], [19] proposed the hybrid structure by combining CNNs/RNNs with self-attention, which is an effective approach to complement the local information of the sequence. However, the above methods usually only focus on the speech local features while ignoring the role of text local information during the inference process. As illustrated in Fig. 1, the loss of the text local information will aggravate the skipping of words during the inference, which leads to "the" deletion errors. In addition, the computational complexity of self-attention in the self-attention-based E2E models increases quadratically as the length of paired speech and text data [20], [21], [22], [23]. Therefore, the existing self-attention-based E2E

Manuscript received 21 July 2022; revised 23 November 2022, 3 January 2023, and 7 February 2023; accepted 7 February 2023. Date of publication 15 February 2023; date of current version 24 February 2023. This work was supported in part by the Beijing Natural Science Foundation of China under Grant L201023, and in part by Natural Science Foundation of China under Grant 62076030. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rohit Prabhavalkar. *(Corresponding author: Yue Ming.)*

The authors are with the Key Laboratory of Work Safety Intelligent Monitoring, the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lvboyang@bupt.edu.cn; fcxg100@163.com; myname35875235@126.com; zpz@bupt.edu.cn; hunan246@bupt.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3245407

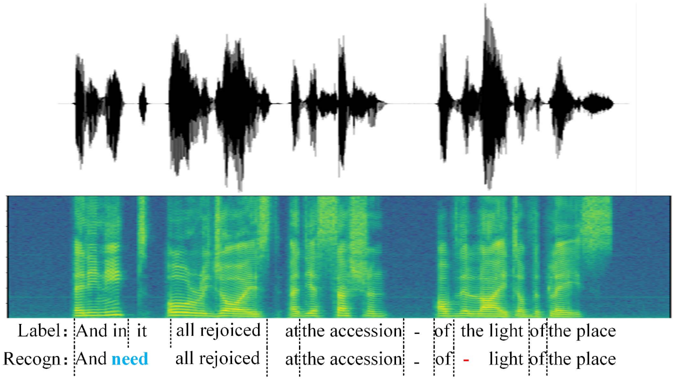


Fig. 1. An example of semantic force-alignment. We use the Montreal Forced Aligner forced alignment between the acoustic signals and the transcriptions to obtain the word-level timing information. The label is “in it” and the recognition output is “need”. The pronunciation of “in it” is similar to “need” when it is pronounced continuously. The distinguishability of speech features can be improved by modeling the position relationships between different acoustic unit features in the spectrogram, which reduces substitution errors caused by similar pronunciation. Meanwhile, the text local information is missing, which will lead to the deletion errors increasing during the decoding process.

models not only ignore the text local information but also the self-attention mechanism in the model has high computational complexity.

To solve the above issues, we propose a novel Enhancing Hybrid Architecture with Fast Attention and Capsule Network (termed En-HACN) for E2E speech recognition, as shown in Fig. 2. Specifically, we propose a novel CNN-Capsule network (CNN-Caps) module before the encoder, which consists of a capsule network and convolutional subsampling in parallel. It uses the capsule output and dynamic routing mechanism to capture the speech spatial information in the spectrogram, which can better improve the distinguishability of speech features. Furthermore, in order to effectively leverage the text local information to improve the correlation between adjacent words, we design a new LocalGRU Augmented Decoder (LA-decoder), in which LocalGRU can model local relationships of the target sequences. Beyond this, we introduce fast attention mechanism into the En-HACN to reduce the computational complexity to $O(L)$ concerning the sequence length L. The results show that our method can not only achieve state-of-the-art accuracy on Aishell-1 and Librispeech but also obtain comparable accuracy on long utterances with higher efficiency.

The main contributions are summarized as follows:

- We propose a novel CNN-Caps module to capture the spatial information between different acoustic unit features in the spectrogram, which can effectively complement the missing position and relationship information of low-level features such as pitch and formant frequency, in E2E speech recognition.
- We propose a novel LA-Decoder, which can effectively provide not only global information but also local information of the text during inference. Besides, we introduce fast attention into the En-HACN, which accelerates inference speed compared with the state-of-the-art self-attention-based methods.
- Experiments on the test of Aishell-1 and Librispeech achieve the recognition accuracy of 5.08% and 3.48%/8.21%, which outperforms existing approaches. Besides, experiments on Aishell-1-long demonstrate the effectiveness of En-HACN on long utterances.

We organize this paper as follows: The related work is reviewed in Section II. Section III details an En-HACN for an E2E speech recognition framework. We describe the Datasets, implementation details, experimental results, and analysis in Section IV, and the conclusions are given in Section V.

II. Related Works

In this section, we first introduce the research status of E2E speech recognition. To address the speech spatial information loss, we introduce the capsule networks in detail. Then, we introduce the augmented attention mechanism to supplement sequence local information during inference. Finally, we review the most relevant work on efficient attention mechanisms that reduce the computational complexity of self-attention.

A. E2E Speech Recognition

The E2E speech recognition techniques [24], [25], [26], [27] have attracted extensive attention because of their simplified architecture and competitive performance. Representative methods include CTC [1], LAS [3], and RNN-T [5]. Recently, Transformer based on self-attention [11], [12] has been applied to speech recognition due to its advantages of modeling global information and high training efficiency. Dong et al. [9] proposed a 2D attention block to model the temporal and spectral dynamics in a spectrogram, which improves discriminated representations of the acoustic features. In addition, Conformer [10] and R-Transformer [10] combined CNNs/RNNs with self-attention to overcome the disadvantage of self-attention in capturing local information of the sequences, but self-attention still requires huge computing resources. Then, to solve the problem of high computational complexity in self-attention, Linear Attention [20], Efficient-Con [21], Quantized Transformer [22], and Emformer [43] reduced the computational complexity in the self-attention-based E2E models by adding constraints to the self-attention. However, the insufficient description of speech spatial information and the loss of text local information in most of the above methods lead to increased substitution and deletion during inference. In addition, the high computational complexity of the self-attention mechanism causes the inference time to increase as the sequence length increases.

B. Capsule Networks

The capsule networks are applied to extract spatial information of features due to their ability to model the positional relationships between input features [28], [29]. Peng et al. [31] used capsule networks to overcome the limitations of information loss in the CNN pooling process. Gu et al. [30] utilized capsule networks to fuse the spatial information of different self-attention heads in machine translation, which reduces the translation difficulty caused by semantic overlap. Subsequently,

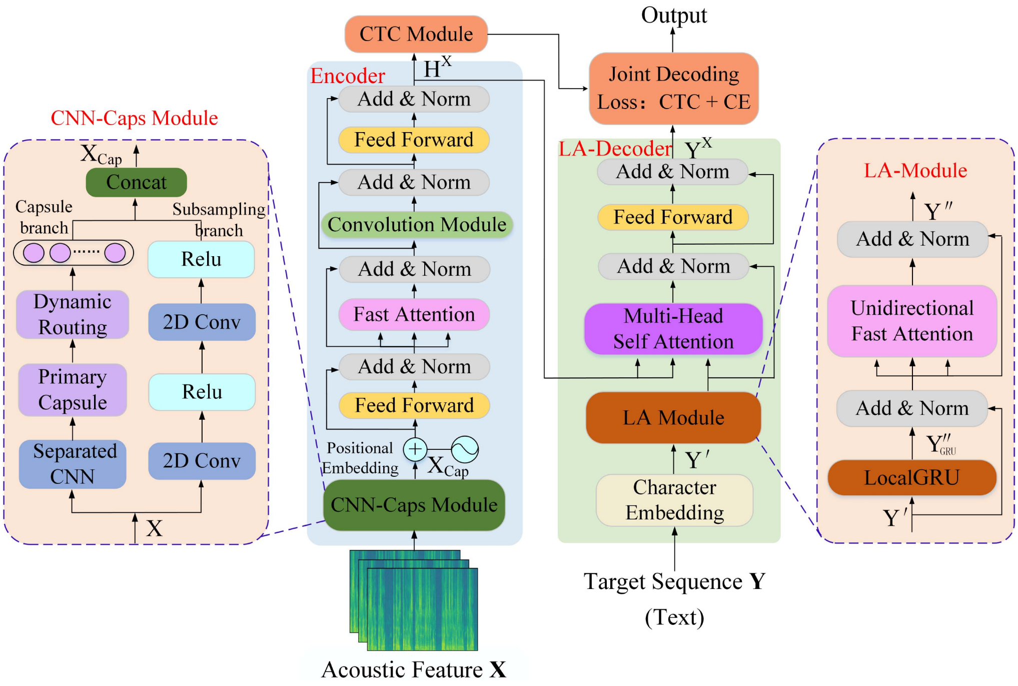


Fig. 2. The overall architecture of Enhancing Hybrid Architecture with Fast Attention and Capsule Network (termed En-HACN) for E2E speech recognition. For better leveraging the spatial information in spectrogram, we devise the CNN-Caps Module in Encoder, which enhances the distinguishability of speech features. Then the LA-Module is proposed to capture global and local information of the target sequences. Finally, the fast attention replaces the standard multi-head self-attention in En-HACN to reduce the computational complexity to $O(L)$ concerning the sequence length L.

Bae et al. [32] pointed out that the spatial information in the spectrogram contains the position and relationship for low-level speech features like pitch and formant frequency, which are essential for speech analysis [33]. Bhanusree et al. [36] leveraged capsule networks with dynamic routing to extract the position and relationship information of pitch and formant features by modeling the position relationships in the spectrogram [34], [35]. Similarly, the existing self-attention-based ASR models use convolutional subsampling in the input layer, which also suffers from the loss of spatial information in the spectrogram. In this paper, we attempt to utilize the capsule networks to supplement the missing spatial information in the spectrogram during convolution subsampling, which improves the distinguishability of speech features in E2E ASR.

C. Augmented Attention Mechanism

Although the attention mechanism has demonstrated extreme effectiveness in capturing global context information, it lacks the ability to obtain local information. Wang et al. [17] proposed the R-Transformer, which captures local and sequential information in sequences without position embeddings. Yang et al. [19] proposed a hybrid self-attention structure to reduce the repetition or skipping of words caused by missing local features in speech synthesis. More recently, some methods [18], [37], [38] added convolution before the transformer, which effectively enhances the ability to extract local speech features. Conformer [10] combined convolution and self-attention to model both local and global relationships of the speech, and the results significantly outperform previous Transformer and CNN models. However, existing methods usually only focus on the speech local relationships and ignore the text local information during inference, which exacerbates deletion or skipping errors. In this paper, we combine the benefits of the attention mechanism and Recurrent RNN/LSTM/GRU to supplement the missing text local information during inference.

D. Efficient Attention Mechanism

As we all know, the computational complexity of self-attention mechanism increases in quadratics with the length of the sequence. In response to this shortcoming, researchers proposed a variety of model variants to reduce the computational complexity of self-attention [39], [40], [41], [42], [43], [44], [45]. Wang et al. [42] proposed the Linformer, which reduces the computational complexity and improves the efficiency of the attention mechanism by reducing the rank of the self-attention matrix. Subsequently, some works [23] have been devoted to reducing the computational complexity of self-attention in speech recognition models due to the duration of audio varying from a few seconds to several minutes. Wang et al. [21] proposed the Conformer with prob-sparse attention to reduce computational complexity. Alex et al. [22] reduced the complexity of the model by simplifying the Transformer architectures. However, the above methods may lose attention information by adding additional constraints or using sparse attention to replace self-attention [46], which can reduce speech recognition accuracy. In this paper, we introduce a fast attention mechanism into the En-HACN, which is different from other low-rank approximate attention matrices (Linformer [42]) in utilizing the kernel-based self-attention formulation and correlation of matrix products. This makes the fast attention reduce the complexity of self-attention from quadratic to linear while reducing attentional information loss.

III. Proposed Method

In this section, we describe the proposed the overview of En-HACN in Section III-A. In Section III-B and Section III-C, we introduce the CNN-Capsule Network module and LocalGRU Augmented Decoder. Finally, Fast Attention Mechanism of En-HACN is presented in Section III-D.

A. Overview

In this paper, we propose a novel Enhancing Hybrid Architecture with Fast Attention and Capsule Network (termed En-HACN) for E2E ASR, as shown in Fig. 2. Firstly, the acoustic features $\mathbf{X} = (x_1, x_2, \ldots, x_L)$ are fed into the CNN-Capsule Network (CNN-Caps) Module to obtain the speech features $\mathbf{X}_{Cap}$. The CNN-Caps module combines the outputs of the capsule network and subsampling to extract the spatial information between different acoustic unit features in the spectrogram.

$$\mathbf{X}_{Cap} = CCN\ Module(\mathbf{X}) \tag{1}$$

where, $CCN\ Module(\cdot)$ refers to the CNN-Capsule Network Module.

Then, the outputs of the CNN-Caps module $\mathbf{X}_{Cap}$ are fed into the Fast Attention and Convolution Module to get the encoded representations $\mathbf{H}^X$, in which the Fast Attention extracts global information of the speech while can reduce the computational complexity to $O(L)$ concerning the sequence length L. The above operations obtained encoded representations $\mathbf{H}^X$ are defined by the following formulas:

$$\mathbf{H} = \mathbf{X_{Cap}} + \frac{1}{2}FFN(\mathbf{X_{Cap}}) \tag{2}$$

$$\mathbf{H}' = \mathbf{H} + FA(\mathbf{H}) \tag{3}$$

$$\mathbf{H}'' = \mathbf{H}' + Conv\,(\mathbf{H}') \tag{4}$$

$$\mathbf{H^X} = \mathbf{H}'' + \frac{1}{2}FFN\,(\mathbf{H}'') \tag{5}$$

where, $FA(\cdot)$ refers to the Fast Attention, $Conv(\cdot)$ represents the Convolution Module [10], $FFN(\cdot)$ refers to the Feed-Forward layers.

Finally, The encoded representations $\mathbf{H}^X$ are transformed into the prediction $P(\hat{y}_n)$ through the CTC module and LocalGRU Augmented Decoder (LA-decoder). For the LA-decoder, we first employ Character Embedding to convert the target sequences $\mathbf{Y} = (y_1, y_2, \ldots, y_N)$ into real-valued vector $\mathbf{Y}'$. Then, the LocalGRU Augmented Module (LA-Module) encodes text embedded features $\mathbf{Y}'$ to obtain text representations $\mathbf{Y}''$, in which LocalGRU captures sequential and local information of embedded features $\mathbf{Y}'$ while extracting global dependencies by Unidirectional Fast Attention. Finally, the text representations $\mathbf{Y}''$ and encoded representations $\mathbf{H}^X$ are transformed into the prediction $P(\hat{y}_n)$. The above operations of LA-decoder to obtain $P(\hat{y}_n)$ are defined as following formulas:

$$\mathbf{Y}' = Character\ Embedding(\mathbf{Y}) \tag{6}$$

$$\mathbf{Y}'' = LA\ Module(\mathbf{Y}') \tag{7}$$

$$\mathbf{Y}''' = \mathbf{Y}'' + MHA(\mathbf{Y}'', \mathbf{H}^X) \tag{8}$$

$$\mathbf{Y}^X = \mathbf{Y}''' + FFN(\mathbf{Y}''') \tag{9}$$

$$\hat{y}_l = softmax(\mathbf{Y}^X) \tag{10}$$

where $Character\ Embedding(\cdot)$ can embed characters into a multidimensional feature matrix, where the value of each dimension represents the characteristics of this character in this dimension [9]. $LA\ module(\cdot)$ refers to the LocalGRU Augmented Module, $MHA(\cdot)$ stands for the Multi-Head Self-Attention [9], and $FFN(\cdot)$ refers to the Feed-Forward layers. For the CTC module, the blank label is inserted to solve the problem that the input length is much larger than the output length in speech recognition, where the target sequences $\mathbf{Y}$ can be extended to $\Omega(\mathbf{Y})$, a sequence set with length N'.

During model training, we predict the posterior distribution $P_{S2S}(Y|X)$ and $P_{CTC}(Y|X)$ frame by frame through the LA-decoder and CTC module, where X represents acoustic feature sequences, and Y represents ground-truth target sequences. The loss function consists of the weighted average of two negative log-likelihoods.

$$L_{ASR} = -\alpha \text{log} P_{S2S}(Y|X) - (1-\alpha)\text{log} P_{CTC}(Y|X) \tag{11}$$

where $\alpha \in [0, 1]$ represents the hyper-parameter to balance the weight of different loss functions. $P_{S2S}(Y|X) = \prod_{n=1}^{N} p(y_n|y_{(1:n-1)}, X)$ describes a cross-entropy (CE) loss, in which $y_{(1:l-1)}$ is previous tokens of y_l. $P_{CTC}(Y|X) = \sum_{\pi\in\Omega(Y)} \prod_{t=1}^{T} p(\pi_t|X)$ describes a CTC loss, in which $\Omega(Y)$ represents the set of all CTC paths π that can be mapped to ground-truth target sequences Y. In the inferencing stage, given the acoustic feature sequences X and the previously predicted token of target sequences as additional input, our En-HACN use beam search to predict the next token.

B. CNN-Capsule Network Module

To complement the spatial information in the spectrogram for enhancing the distinguishability of speech features, we propose the CNN-Capsule Network (CNN-Caps) module, which concatenates the output of the capsule branch and subsampling branch in parallel, As shown in Fig. 3. The capsule network models the position relationships of low-level features in spectrogram through capsule output and dynamic routing mechanism, in which the capsule structure utilizes a group of neurons in place of traditional neurons. We utilize the capsule output vector multiplied by the learnable transformation matrix $\mathbf{W}$ to encode the spatial information between low-level and high-level features, which can avoid local-to-global information loss.

$$\mathbf{X}_{Cap} = Concat(Capsule(\mathbf{X}); subsampling(\mathbf{X})) \tag{12}$$

where the $Capsule(\cdot)$ refers to the capsule network, and the $subsampling(\cdot)$ uses Convolution 2D (Conv2D) with two layers to reduce the frame-rate of acoustic features, where each layer

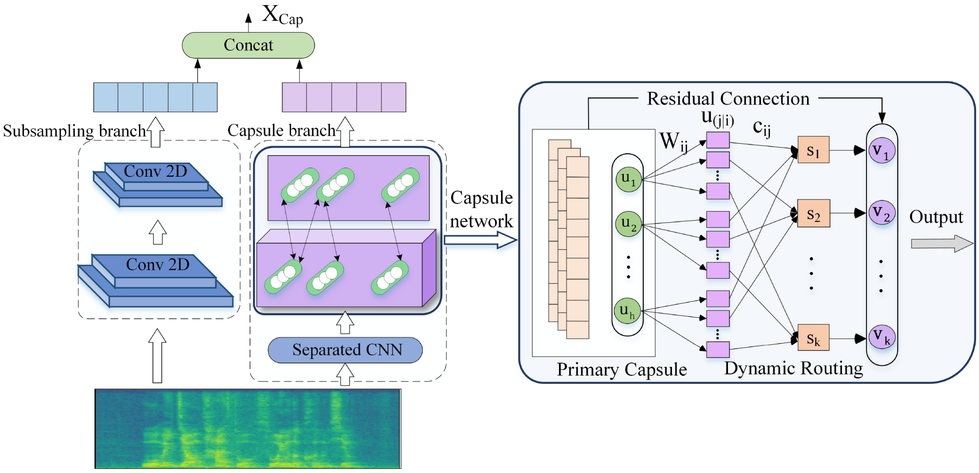


Fig. 3. The architecture of the CNN-Capsule Network Module, including the subsampling branch and the capsule branch. For the subsampling branch, the purpose is to reduce the frame-rate of the encoder input through convolution 2D. For the capsule branch, we utilize capsule output and dynamic routing algorithm to couple the capsules at various positions in the lower layer with the capsules in the upper layer, which captures the existence probability and spatial information.

employs 3×3 Convolution 2D. There are four subsampling times in total.

For the capsule branch, the parameters of the capsule network are proportional to the number of frames. Therefore, we split the input sequence into shorter windows, in which the window is set to a fixed length L and the fixed overlap is $D = L/2$. We chose 50% overlap, which makes any point of speech is covered by exactly two segments and the loss of information at segment boundaries can be recovered by referencing another overlapping segment. Then, we leverage two separated convolutions to capture the relationship information across temporal and spectral in each window. The output of separated convolutions is fed to the capsule network to capture the spatial information between the different acoustic unit features in spectrogram, in which the capsule network includes the Primary Capsule and Dynamic Routing mechanism. We utilize the shared primary capsule layer to obtain the low-level feature $\boldsymbol{u}_i$. The primary capsule layer is composed of convolution with a kernel size of 5×5 and stride 2. Subsequently, the low-level features $\boldsymbol{u}_i$ is routed to generate high-level feature representations $Capsule(\mathbf{X}) = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K]$.

Specifically, the acoustic features $\mathbf{X}$ are rearranged to calculate the low-level feature $\mathbf{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K)$ $\boldsymbol{u}_i \in R^d$. In order to enhance the representation ability of features, the capsule list output $\boldsymbol{u}_{(j|i)}$ is first calculated by multiplying $\boldsymbol{u}_i$ by a learned transformation matrix $\mathbf{W}_{ij}$: $\boldsymbol{u}_{(j|i)} = \mathbf{W}_{ij}\boldsymbol{u}_i$. All primary capsule vectors $\boldsymbol{u}_{(j|i)}$ are integrated to calculate the common output $\boldsymbol{s}_j$ by summing up all primary capsule vectors with weights c_{ij}. Then, the output capsule vectors $\boldsymbol{s}_j$ is normalized to obtain $\boldsymbol{v}_j$ as the output of the capsule network. The above operations are defined by the following formulas:

$$\boldsymbol{s}_j = \sum_i c_{ij} \cdot \boldsymbol{u}_{(j|i)} \tag{13}$$

$$\boldsymbol{v}_j = \frac{||\boldsymbol{s}_j||^2}{(1 + ||\boldsymbol{s}_j||^2)} \cdot \frac{\boldsymbol{s}_j}{||\boldsymbol{s}_j||} \tag{14}$$

where $c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}$ is the coupling coefficient, which represents the correlation between the high-level feature $\boldsymbol{v}_j$ and the lower-level feature $\boldsymbol{u}_i$.

During the above process, we use b_{ij} to update c_{ij}. The b_{ij} is a newly developed set of temporary variables and corresponds to c_{ij} one by one. It is defined as the possibility that the capsule $\boldsymbol{u}_i$ in the l layer is connected to $\boldsymbol{s}_j$ in the $l+1$ layer. And the softmax function is used to make c_{ij} uniformly distributed so that the lower feature $\boldsymbol{u}_i$ can be transmitted to the higher feature $\boldsymbol{v}_j$ with maximum uncertainty. The dynamic routing mechanism establishes a nonlinear mapping relationship from $\boldsymbol{u}_{(j|i)}$ to $\boldsymbol{v}_j$, where routing iteration is 3.

$$b_{ij} = b_{ij} + \boldsymbol{u}_{(j|i)} \cdot \boldsymbol{v}_j \tag{15}$$

Initially, b_{ij} is set to 0, and the initial coupling coefficient b_{ij} is iteratively refined by the point product consistency between the input capsule and each output capsule.

C. LocalGRU Augmented Decoder

To effectively improve the ability to model local relationships on the target sequences, we design a novel hybrid structure LocalGRU Augmented Decoder (LA-decoder), which consists of LocalGRU Augmented Module (LA-module) and Multi-Head Self-Attention. As shown in Fig. 2, the LA-module consists of a LocalGRU and Unidirectional Fast Attention. It leverages the ability of LocalGRU to model local relationships of target sequences, which enhances the correlation between adjacent words. Firstly, we employ LocalGRU to capture local and sequential information of the target sequences without position embedding in our model. Then, we leverage the Unidirectional Fast Attention to capture global information of the target sequences. Finally, the projected 256-dimensional outputs $\mathbf{Y}''$ from the LA-module and encoder representation $\mathbf{H}^X$ as Multi-Head Self-Attention input to predict $P(\hat{y}_n)$.

In the LA-module, the target sequences $\mathbf{Y} = (y_1, y_2, \ldots, y_N)$ is embedded into the embedded feature $\mathbf{Y}'$ via a Character Embedding layer. Then, the embedded feature $\mathbf{Y}'$ is transmitted into the LA-module, which captures the local and global information of target sequences through integrating the LocalGRU and Unidirectional Fast Attention. This process could be described as:

$$\mathbf{Y}''_{GRU} = \mathbf{Y}' + LocalGRU(\mathbf{Y}') \tag{16}$$

$$\mathbf{Y}'' = \mathbf{Y}''_{GRU} + UnFA(\mathbf{Y}''_{GRU}) \tag{17}$$

where, the $UnFA$ refers to the Unidirectional Fast Attention mechanism.

For the LocalGRU, we divide the original sequence into many short sequences that only contain local information, and utilize a shared Gated Recurrent Unit (GRU) to process the different short sequences independently. Firstly, we construct a local window with length M. Secondly, each local window forms a short sequence $(y'_{t-M-1}, \ldots, y'_t)$, which contains local information. Finally, the latent representations of the local short sequence are combined to obtain the local and sequential information of the whole sequence. We refer to the shared GRU as LocalGRU. This process could be described as:

$$h_t = LocalGRU\left(y'_{t-M-1}, \ldots, y'_t\right) \tag{18}$$

$$h_1, \ldots, h_N = LocalGRU\left(y'_1, y'_2, \ldots, y'_N\right) \tag{19}$$

The entire embedded features are encoded as $Y''_{GRU} = Y' + LocalGRU(y'_1, y'_2, \ldots, y'_N)$.

The GRU usually has only two gates: a reset gate r and an update gate z, which obtains the information of r and z through current input y'_t and previously hidden state h:

$$r_t = \sigma\left(\mathbf{W_r} \cdot [h_{t-1}, y'_t]\right) \tag{20}$$

$$z_t = \sigma(\mathbf{W_z} \cdot [h_{t-1}, y'_t]) \tag{21}$$

After getting the gating information, the current input y'_t is spliced with the reset data. Then, the output of the currently hidden node is activated by tanh activation function. Finally, the output information of the hidden state h_t is computed as:

$$\tilde{h}_t = \tanh\left(\mathbf{W}\left[r_t h_{t-1}, y'_t\right]\right) \tag{22}$$

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \tag{23}$$

The LocalGRU Augmented module can model the local relationships of the target sequences, which supplements the missing text local information during the inference process, and further reduce the occurrence of deletion errors or word skipping.

D. Fast Attention Mechanism in En-HACN

We introduce fast attention mechanism into the En-HACN, which reduce the computational complexity to $O(L)$ concerning the paired speech and text data length L. Given the input matrix $\mathbf{X} \in \mathbb{R}^{L \times d_x}$ with sequence length L, where d_x is dimension of input. The projection matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ transform the $\mathbf{X}$ into query, key and value matrices respectively. The self-attention mechanism [9] in scaled dot-product form can be expressed by:

$$Att_{\leftrightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{QK}^T}{\sqrt{d}}\right)\mathbf{V} \tag{24}$$

where $\mathbf{Q} = \mathbf{XW}_Q$, $\mathbf{K} = \mathbf{XW}_K$, $\mathbf{V} = \mathbf{XW}_V$.

In order to describe fast attention [46], (24) will be reformulated to obtain the following formula:

$$Att_{\leftrightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{QK}^T}{\sqrt{d}}\right)\mathbf{V} = \mathbf{D}^{-1}\mathbf{AV} \tag{25}$$

where, $\mathbf{A} = exp(\frac{\mathbf{QK}^T}{\sqrt{d}})$, $\mathbf{D} = diag(\mathbf{A1}_L)$, $\mathbf{1}_L = (1)_{L \times L}$. The $exp(\cdot)$ is applied elementwise and $diag(\cdot)$ is a diagonal matrix with the input vector. It can be seen from the above formula that the purpose of the modified $\mathbf{D}^{-1}\mathbf{AV}$ is to split the softmax function of the self-attention into the multiplication of two matrices, and the computational complexity does not change.

The core of the fast attention mechanism is to represent the attention matrix A through the kernel function as: $A(i, j) = K(q_i^T, k_j^T)$. The matrix $\mathbf{Q}$ and $\mathbf{K}$ are mapped to the new matrix $\mathbf{Q}'$ and $\mathbf{K}'$ through the mapping relationship $\phi(x)$ of kernel function. The attention matrix $\mathbf{A}$ is approximate to the product of $\mathbf{Q}'$ and $\mathbf{K}'$, that is:

$$\mathbf{A} = exp\left(\frac{\mathbf{QK}^T}{\sqrt{d}}\right) \approx \phi(\mathbf{Q})\phi(\mathbf{K})^T = \mathbf{Q}^{'}(\mathbf{K}')^T \tag{26}$$

where $\mathbf{Q}^{'}, \mathbf{K}' \in \mathbb{R}^{L \times r}$. Mapping function $\phi(x)$ in the above formula acting on a matrix is actually mapping the row vector of the matrix. The rows of matrices $\mathbf{Q}^{'}, \mathbf{K}' \in \mathbb{R}^{L \times r}$ can be represented as $\phi(q_i)$ and $\phi(k_j)$ respectively. The function $\phi(x)$ is defined as:

$$\phi(x) = \frac{h(x)}{\sqrt{M}}\left(f(\omega_1^T x), f(\omega_2^T x), \ldots, f(\omega_M^T x)\right) \tag{27}$$

where $h(x) > 0$ is a constant, $\mathbf{W} = (\omega_1, \omega_2, \ldots, \omega_m)$ is a random feature matrix, and M represents the number of random features. The number of random features M is independent of the sequence length L and related to the feature dimension d.

Through the transformation of (26), the attention matrix $\mathbf{A}$ is expressed as the product of two low-rank matrices. Then, the matrix multiplication is rearranged to approximate the result of the self-attention mechanism. Firstly, we calculate the product of $\mathbf{K}'$ and $\mathbf{V}$. Secondly, multiplying the results of the first step with $\mathbf{Q}'$ can reduce the computational complexity into linear respectively. (25) will be reformulated follows:

$$\widehat{Att_{\leftrightarrow}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \hat{\mathbf{D}}^{-1}(\mathbf{Q}'((\mathbf{K}')^T\mathbf{V})) \tag{28}$$

where $\widehat{\mathbf{D}} = diag(\mathbf{Q}^{'}((\mathbf{K}^{'})^T 1_L))$, $\widehat{Att_{\leftrightarrow}}$ stands for approximate orginal attention. The specific process is shown in Fig. 4, where the dashed blocks indicate the order of computations.

The unidirectional fast attention is defined as following formulas:

$$Att_{\leftrightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}\mathbf{V} = \widetilde{\mathbf{D}}^{-1}tril(\mathbf{Q}^{'}(\mathbf{K}^{'})^T\mathbf{C}) \tag{29}$$

where, $\widetilde{\mathbf{D}} = diag(\widetilde{\mathbf{A}}\mathbf{1}_L)$, $\widetilde{\mathbf{A}} = tril(\mathbf{A})$, and the $tril(\cdot)$ stands for the lower-triangular portion of the matrix, including the diagonal. The $\mathbf{Q}^{'}$ and $\mathbf{K}^{'}$ were computed as described in (27).

TABLE I
DETAILS INFORMATION OF CORPUS

Dataset	language	type	Total (hours)	Min Segment (sec.)	Max Segment (sec.)
Aishell-1	Chinese	monolog	178	1.0	12.0
aidatatang_200zh	Chinese	monolog	200	1.0	10.0
Thchs-30	Chinese	monolog	30	4.0	16.0
HKUST	Chinese	dialog	200	2.0	13.0
Librispeech	English	monolog	960	1.0	17.0
Aishell-1-long	Chinese	monolog	178	17.0	46.0

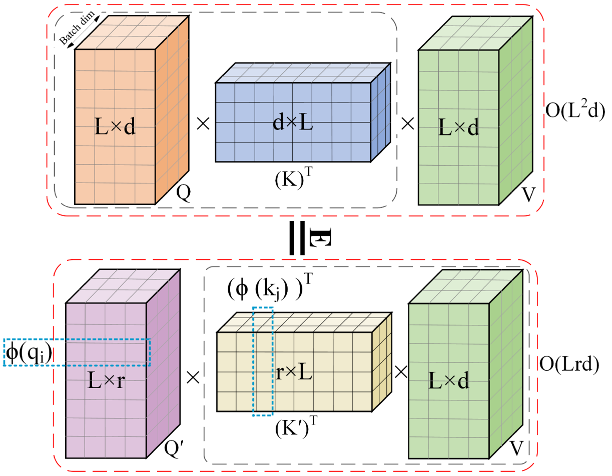


Fig. 4. Approximation of the regular attention mechanism by fast attention, where $\mathbf{E}$ represents the approximately equal. Dashed-blocks indicate order of computation.

We computer $tril(\mathbf{Q}^{'}(\mathbf{K}^{'})^T\mathbf{C})$ without constructing and storing the $L \times L$ sized matrix $tril(\mathbf{Q}^{'}(\mathbf{K}^{'})^T)$ explicitly, where $\mathbf{C} = [\mathbf{V}\ \mathbf{1}_L] \in \mathbb{R}^{L\times d+1}$.

IV. Experiments and Results

In this section, we first evaluate the performance of the proposed En-HACN on five different corpora. Secondly, we designed ablation experiments to verify the effectiveness of our proposed sub-module. Finally, we conduct experiments on different length utterances to analyze the generalization performance of our model.

A. Datasets

The effectiveness of the proposed model was evaluated on five different corpora, which are Aishell-1, aidatatang_200zh, Thchs-30, HKUST, and Librispeech respectively. Table I summarizes the details of datasets. The Aishell-1 contains about 178 hours of speech recordings. The aidatatang_200zh is part of the aidatatang_1505zh mandarin, containing 200 hours of speech data. The Thchs-30 is an open speech database published by the Center for Speech and Technologies (CSLT) of Tsinghua University, containing 30 hours of speech data. The HKUST contains about 200 hours of telephone recording dataset. The LibriSpeech consists of 960 hours of speech recordings from audiobooks.

In the above corpora, Aishell-1, aidatatang_200zh, Thchs-30, HKUST, and Librispeech are mainly composed of utterances less than 17 seconds. To validate the generalization ability of our model in long-form speech recognition scenarios. We use the Sound eXchange (Sox) [47] toolkit to process the Aishell-1 dataset into the Aishell-1-long corpus. Specifically, we utilize the Sox sound processing program to merge the utterances of the same speaker from the original corpus into new utterances. The Aishell-1-long is synthesized by Aishell-1, which consists dev set of 3 k utterances, and the test set consists of 1515 utterances. The average length of utterances in the Aishell-1-long is 25 seconds.

B. Experiment Setup

We use the WeNet toolkit [49] for training and decoding the model. The input features are 80-dimensional filterbanks features extracted with a window size of 25 ms and shifted every 10 ms. We apply the default setting of SpecAugmentation [48], and the speed perturbation on the training set by factors 0.9, 1.0, and 1.1. During the experiment, we explored different hyper-parameter combinations for the En-HACN. Our En-HACN contains of 12-layer encoder and 6-layer decoder. The model dimension $d_{model} = 256$, the dimensionality of the inner-layer in FFN $d_{ff} = 2048$ and the number of attention heads $h = 4$ are kept unchanged. The En-HACN (large) has 16-layer encoder, 1-layer decoder, and the number of attention heads $h = 8$, in which the model dimension and dimensionality of the inner-layer in FFN are the same as En-HACN. The local window length $M = 3$ of LocalGRU in our experiments. For decoding, we set the CTC weight and language model (LM) weight to 0.5 and 0.4 respectively. The Adam optimizer and the learning rate schedule with 25000 warm-up steps are used to train our models. The peak learning rate is 0.002. Our model was trained on 4 GTX 1080Ti GPU, and all experiments used the same random seed. We trained our models for 120 epochs and averaged the best ten checkpoints as the final model.

C. Compare With the State-of-the-art Methods on Different Corpora

Table II shows the experimental results of different methods on Aishell-1, where the Conformer-CTC [49] is the baseline model and is trained from scratch for comparison. It can be seen that our En-HACN achieves 4.64%/5.08% character error rate (CER) on the Aishell-1 corpus. Compared with the Conformer-CTC (Wenet), the CER of En-HACN is relatively

TABLE II
EXPERIMENTAL RESULTS (CER) ON AISHELL-1 DATASET WITH DIFFERENT MODELS

Model	Params	Aishell-1	
		DEV	TEST
HS-DACS Transformer [54]	-	6.20	6.80
CASS-NAT [50]	-	5.30	5.80
Improved CASS-NAT [52]	-	4.90	5.40
Transformer [55]	31M	6.00	6.70
Efficient-Con [21]	61M	5.90	6.50
LDSA [60]	53M	5.79	6.46
Stochastic Attention Head Removal [53]	31M	5.20	5.80
NAR-Transformer [51]	30M	5.20	5.70
Conformer-CTC [49]	49M	5.18	6.06
Ours: En-HACN	46M	**4.64**	**5.08**

The best performance is shown in bold, and the second-best result is underlined.

reduced by 10.4%/16.1%, which shows that our En-HACN can effectively reduce the CER on the Aishell-1 corpus. Compared with the Efficient-Con [21] and LDSA [60], the fast attention in En-HACN can get better performance because it does not add external constraints. Other works [50], [51], [52] focus on building non-autoregressive (NAR) transformer models, which lead to accuracy degradation due to their inability to provide effective target-side information. On the Aishell-1 dataset, the proposed En-HACN achieves state-of-the-art results compared with existing works.

Table III shows the word error rate (WER) results on the Librispeech corpus, where the Transformer (Espnet) [55] and Conformer-CTC [49] are trained from scratch for comparison. Without LM, the WER of En-HACN is 3.48%/8.21% on the test-clean/test-other. The LA-decoder of En-HACN combines the advantages of the RNN-T prediction network [61] and Transformer decoder [55], while the CNN-Caps module supplements missing spatial information in the spectrogram during CNN subsampling. Our En-HACN (large) obtains 2.66%/5.93% WER and achieves suboptimal performance compared with existing works, in which the hyper-parameter of En-HACN (large) is similar to Conformer [10], i.e., 16-layer encoder, 1-layer decoder, the attention heads $h = 8$, and $d_{model} = 256$. After fusing external LSTM LM, the WER of En-HACN (large) is reduced by 0.46%/1.03% on the test-clean/test-other dataset. Furthermore, it can be seen that our En-HACN (large) has achieved state-of-the-art performance compared to existing works with the fusion of LSTM LM. It is obvious that the results of our models achieve competitive performance.

To obtain the results of unified experimental conditions, we retrain Transformer/Conformer (Espnet) and Transformer/Confirmer CTC (Wenet) from scratch on six datasets for comparison, where Espnet uses RNN LM and Wenet uses N-gram LM during decoding. Table IV shows the CER/WER results for different corpus. It can be seen that the performance of the Transformer/Conformer (Conformer-CTC) in the Wenet toolkit and Espnet toolkit is better than the TDNN in the Kaldi toolkit. All results on corpora show that the performance of the Conformer/Conformer-CTC is better than the Transformer. Our En-HACN achieves suboptimal recognition results on Librispeech. Furthermore, the experimental results on the test dataset of aidattang_200zh, Aishell-1, Thchs-30, and HKUST corpora show that the CER/WER of our En-HACN model is relatively reduced by 5.29%/16.1%/5.75%4.76% compared with the Conformer-CTC (Wenet) model. The proposed En-HACN model achieves 7.01%/7.45% WER on Aishell-1-long and has higher robustness in long utterances scenarios.

D. Ablation Study of En-HACN Model

We conduct different ablation experiments on Aishell-1 and Aishell-1-long to demonstrate the effectiveness of each proposed module. One is to verify the impact of adding the CNN-Caps module and LA-decoder to the Conformer-CTC on the recognition performance. Another is to analyze the impact of different parameter settings of fast attention on performance. Firstly, we add the CNN-Caps module and LA-decoder in En-HACN-A and En-HACN-B based on Conformer-CTC respectively, which explore their roles. In addition, the En-HACN-C with CNN-Caps module and LA-decoder enhances the expressive power of paired speech and text data features, thereby improving recognition accuracy, as shown in Table V. Then, we explore the impact of different hyper-parameter combinations for fast attention and fast attention on model performance, as shown in Table VI, and Table VII.

1) Effectiveness of CNN-Caps Module: To validate the effectiveness of the CNN-Caps module, we compare our method with Transformer [55] and Conformer-CTC [49] models on the Aishell-1 and Aishell-1-long. Table V shows that the CER of En-HACN-A achieves 4.75%/5.68% on the dev/test set of Aishell-1, and the CER is relatively decreased by 8.3%/6.2% compared with Conformer-CTC. Compared with Transformer, the CER is reduced by 20.8%/15.2%. Meanwhile, on the dev/test set of Aishell-1-long, the CER relative to Conformer-CTC is reduced by 0.34%/0.45% respectively. These results demonstrate that the CNN-Caps module can model the position relationships between different acoustic unit features in the spectrogram, which complements the missing spatial information in speech analysis and improves the discriminability of speech features. Hence, in speech recognition, we will insert the CNN-Caps module for better accuracy.

In the CNN-Caps module, dynamic routing is the critical process for ensuring the grouping of lower-layer to upper-layer capsules, in which a small iterations number may lead to insufficient capsule grouping and a large number of iterations can bring a huge computational burden. We further study the impact of routing iterations on recognition performance. As shown in Fig. 5, the comparison of 1-5 routing iterations is performed in Aishell-1 and Aishell-1-long. We found that the recognition performance on the two different datasets reached the best when the iteration was set to 3.

2) Effectiveness of LA-decoder: We concatenate the Local-GRU and unidirectional fast attention to catch the salient benefits of both. Table V shows that the CER of the En-HACN-B model is lower than the Conformer-CTC/Transformer model.

TABLE III
EXPERIMENTAL RESULTS (WER) ON LIBRISPEECH DATASET WITH DIFFERENT MODELS

Model	Params	Language Model (LM)	Dev		Test	
			clean	other	clean	other
Without LM						
RNN-T [61]	-	-	3.30	9.70	3.60	9.50
CASS-NAT [50]	-	-	3.70	9.20	3.80	9.10
ContextNet-M + Fast Emit [57]	-	-	-	-	3.50	8.60
Simplified Fully Quantized Transformer [22]	51M	-	5.40	14.5	5.50	15.2
Efficient-Con [21]	61M	-	4.20	10.9	4.40	11.0
Transformer (Espnet) [55]	31M	-	3.70	9.80	4.00	10.0
CONV2D4+TR2 [16]	75M	-	3.70	8.20	3.50	8.50
Conformer-CTC (Wenet) [49]	49M	-	3.50	8.82	3.63	8.72
Conformer [10]	31M	-	-	-	**2.30**	**5.00**
Transformers with convolutional context [38]	223M	-	4.80	12.7	4.70	12.9
Ours: En-HACN	46M	-	3.43	8.77	3.48	8.21
En-HACN (large)	121M	-	2.54	6.06	2.66	5.93
With LM						
HS-DACS Transformer [54]	-	LSTM LM	2.40	6.50	2.70	6.60
TCPGen [62]	-	LSTM LM	-	-	2.59	7.13
GTC-T [56]	-	LSTM LM	2.40	6.10	2.70	6.20
Quantization with conformer [63]	494 M	4-gram LM	2.47	6.04	2.78	6.19
CTC-Transformer [58]	32 M	LSTM LM	3.80	8.60	4.10	8.70
MEL-t-Fusion-Late [59]	140M	LSTM LM	3.05	6.63	3.34	7.15
Ours: En-HACN	46M	4-gram LM	3.20	7.56	3.32	8.08
En-HACN		LSTM LM	3.10	6.45	3.30	7.10
En-HACN (large)	121M	LSTM LM	**2.10**	**4.70**	**2.20**	**4.90**

The best performance is shown in bold, and the second-best result is underlined.

TABLE IV
COMPARISON THE RESULTS ON VARIOUS OPEN SOURCE ASR CORPORA

Dataset	CER	Kaldi (Chain+TDNN)	Transformer (Espnet)	Conformer (Espnet)	Transformer (Wenet)	Conformer-CTC (Wenet)	Ours: En-HACN
aidatatang_200zh	CER: Dev/test	(† ⋇) -/ 7.21	(†∗) 5.90/6.70	(†∗) 5.40/6.10	(†⋇) 5.83/6.54	(†⋇) **4.72**/5.29	(†∗) 5.21/**5.01**
Aishell-1	CER: Dev/test	(†⋇) -/8.64	(†∗) 6.00/6.70	(†∗) 4.90/5.40	(†⋇) 5.97/6.85	(†⋇) 5.18/6.06	(†∗) **4.64/ 5.08**
Thchs-30	CER: Dev/test	(†⋇) -/23.30	-	-	(†⋇) 14.38/15.63	(†⋇) **12.72**/13.90	(†∗) 12.73/**13.10**
Librispeech	WER: test_clean/ test_other	(†⋇) 5.72/13.72	(†∗) 4.00/10.0	(†∗) **3.10/6.90**	-	(†⋇) 3.51/8.34	(†∗) 3.30/7.10
HKUST	CER: Dev	(†⋇) 29.44	(†∗) 23.5	(†∗) 21.09	(†⋇) 22.47	(†⋇) 21.42	(†∗) **20.40**
Aishell-1-long	CER: Dev/test	-	(†∗) 12.53/13.20	(†∗) 10.25/11.10	(†⋇) 11.05/12.54	(†⋇) 9.05/9.24	(†∗) **7.01/7.45**

† indicate w/ speed perturbation, ∗ and ⋇ denote w/ RNN LM or N_gram LM, respectively.

TABLE V
RESULTS (CER) OF EN-HACN MODEL ABLATION EXPERIMENT, WHERE ✓ INDICATES THAT THE MODULE EXISTS

Model	CNN-Caps module	LA-decoder	Aishell-1		Aishell-1-long	
			DEV	TEST	DEV	TEST
Transformer [55]	-	-	6.00	6.70	12.73	14.20
Conformer-CTC [49]	-	-	5.18	6.06	9.13	9.71
En-HACN-A	✓	-	4.75	5.68	8.79	9.26
En-HACN-B	-	✓	4.95	5.97	8.95	9.34
En-HACN-C	✓	✓	**4.50**	**5.34**	**8.52**	**8.84**

The best performance is shown in bold, and the second-best result is underlined.

TABLE VI
COMPARISON THE RESULTS (CER) ON AISHELL-1 AND AISHELL-1-LONG WITH DIFFERENT f AND W

Model	$f(\cdot)$	W	Aishell-1		Aishell-1-long	
			DEV	TEST	DEV	TEST
Conformer-CTC	f=exp	Givens matrices	5.43	5.95	9.02	9.68
		Random Hadamard matrices	5.74	6.31	9.11	9.65
		Gaussian orthogonal matrices	**5.21**	**6.26**	**8.65**	**9.01**
	f=Relu		5.32	6.11	8.84	9.31

TABLE VII
COMPARISON OF RESULTS (CER) OF INTRODUCING FAST ATTENTION MECHANISM IN DIFFERENT MODELS

Model	Params	Aishell-1		Aishell-1-long	
		DEV	TEST	DEV	TEST
Transformer [55]	33M	6.00	6.70	12.73	14.20
+ fast attention	29M	5.99	6.94	11.34	12.67
Conformer-CTC [49]	49M	5.18	6.06	9.13	9.71
+ fast attention	43M	5.21	6.62	8.65	9.01
En-HACN	53M	**4.50**	5.34	8.52	8.84
+ fast attention	46M	4.64	**5.08**	**7.01**	**7.45**

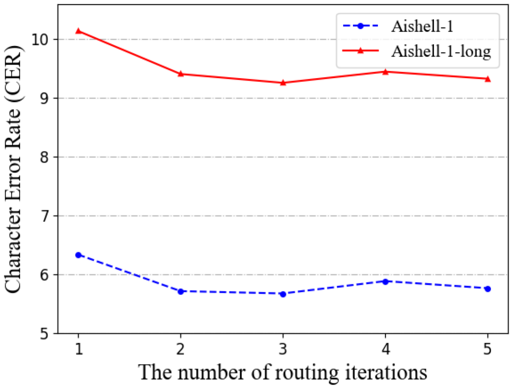


Fig. 5. Performances of the CNN-Caps module with different routing iterations.

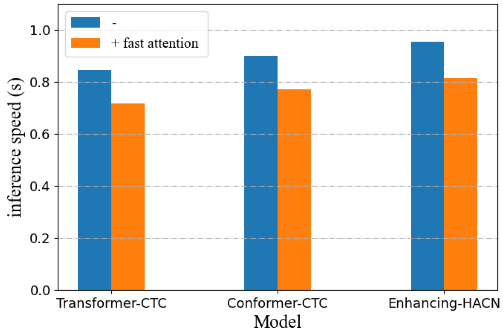


Fig. 6. The inference speed of different models during the decoding process for utterances with the same length (30 seconds).

The CER of En-HACN-B is 4.95%/5.97% on the dev/test set of Aishell-1, with 3.8% and 1.4% relatively decrease in CER to the Conformer-CTC model. We also study the effects of different ways of combining the unidirectional fast attention with the LocalGRU. We try to transmit the embedded feature into a unidirectional fast attention branch and a LocalGRU branch, and then concatenated their output in parallel as suggested in [18]. We find that it degrades the results by 1.2%/0.9 on the test set of Aishell-1 and Aishell-1-long when compared to our proposed architecture. Thanks to the LA-decoder, our model is able to provide detailed text local information for inference, reducing deletion errors during the decoding process.

3) Effectiveness of Fast Attention Mechanism: We explore different hyper-parameter combinations for the fast attention, including the parameter $f(\cdot)$ of the $\phi(x)$, and the random feature matrix W. Choromanski et al. [46] points out that $\mathbf{W} = (\omega_1, \ldots, \omega_M)$ performs Gram-Schmidt operation can effectively reduce the variance of estimation and improve the average accuracy. Therefore, the Gram-Schmidt operation is performed on the random feature matrix $\mathbf{W}$ in the experiment. In order to verify, we make the parameter $\mathbf{W}$ obey three different distributions: Givens matrices, Random Hadamard matrices, and Gaussian orthogonal matrices. The experimental results of Aishell-1 and Aishell-1-long in Table VI shows that the model works best when $f = exp$ and $\mathbf{W} \sim N(0, 1_d)$. The CER reaches 6.26%/9.01% on the test set of the Aishell-1 and Aishell-1-long respectively.

With the input length increases, the self-attention mechanism will not only cause a significant increase in CER due to alignment errors but also increase the time consumption of the inference process. Thus, we add fast attention mechanism in Transformer, Conformer, and En-HACN to explore the performance of the model in Aishell-1-long. It can be seen from Table VII that the number of model parameters is reduced by about 12% after using the fast attention mechanism. Compared with the Conformer-CTC without updating the self-attention with fast attention, the number of model parameters is reduced by about 6.1%, while the recognition accuracy is guaranteed. Furthermore, we evaluate the inference speed of different models based on the fast attention mechanism on the same length utterance during decoding. As shown in Fig. 6, adding fast attention, the model achieves 15% to 20% inference speed-up in the same utterance lengths and achieves the best recognition performance on long utterances Aishell-1-long.

E. Comparison the Complexity and Inference Speed

This section compares our model's computational complexity and real-time factor (RTF) with other methods, as shown in Table VIII . Our model's computational complexity can be reduced from square to linear with sequence length L, which

TABLE VIII
COMPARISON THE CER AND RTF OF MODELS BASED ON DIFFERENT ATTENTION IN THE TEST SETS OF AISHELL-1 AND AISHELL-1-LONG, WHERE L IS THE LENGTH OF INPUT FEATURE AND C IS THE CONTEXT WIDTH

Model	Computational complexity	Aishell-1	Aishell-1 -long	RTF
DSA [60]	$O(L^2)$	7.26	16.50	0.457
Transformer [55]	$O(L^2)$	6.70	14.20	0.403
Conformer [10]	$O(L^2)$	5.18	11.84	0.543
Conformer-CTC [49]	$O(L^2)$	6.06	9.71	0.197
CTC-Enhanced [51]	$O(L^2)$	5.20	8.01	**0.171**
Linear Attention [20]	$O(L)$	5.54	9.37	0.357
Efficient-Con [21]	$O(L)$	6.50	8.54	0.237
LDSA [60]	$O(Lc)$	6.49	11.50	0.367
En-HACN	$O(L)$	**5.08**	**7.45**	0.183

The best performance is shown in bold, and the second-best result is underlined.

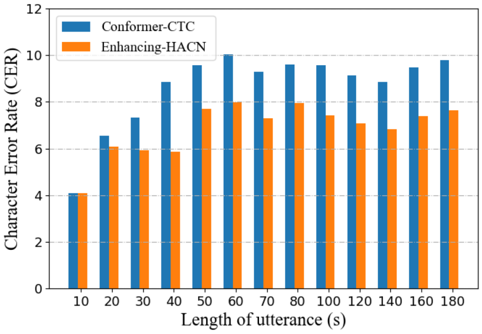


Fig. 7. Effect of different utterance length (s) on CER of Conformer-CTC and En-HACN.

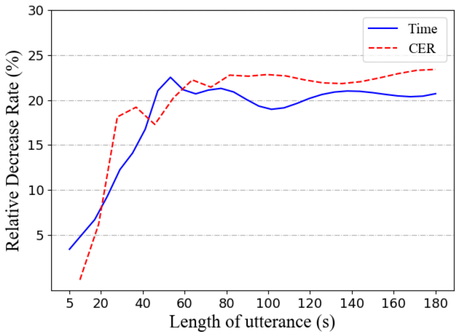


Fig. 8. Relative decrease rate of CER and inference speed with different utterance length (s).

is lower than DSA [60], Transformer [55], Conformer [10], and CTC-Enhanced [51]. In addition, we select the Aishell-1-long to test the RTF of different models, where the RTF was used to measure the inference speed on GPU (NVIDIA GTX 1080Ti). The results show that the RTF of our model is only 0.012 higher than the CTC-Enhanced [51] because CTC-Enhanced uses a non-autoregressive decoding method for faster inference. However, the non-autoregressive decoding method will reduce its recognition accuracy. The CER of our model on the Aishell-1 and Aishell-1-long datasets is 0.12%/0.56% lower than CTC-Enhanced. This shows that the fast attention mechanism reduces the computational complexity of the attention mechanism in the model to linear, enabling inference speedup while ensuring recognition performance.

F. Experiments on Different Length Utterances

To further investigate the generalization ability of the En-HACN to different length utterances, we use the Sox sound processing programs to synthesize the utterances of different segment lengths. The synthesized long utterances contain 5/10/20/30/40/50/60/70/80/90/100/120/140/160/180 seconds of utterance. We report the accuracy of utterances with different lengths, in which the model parameters are kept to be the same for all experiments. Fig. 7 shows the recognition results of Conformer-CTC and En-HACN on utterances of different segment lengths. As the speech length increases, the CER of both models increases. Compared with the Conformer-CTC model, our proposed model reduces the CER by 0.47% to 2.98% on different utterance lengths, which indicates the effectiveness of our En-HACN on long utterances.

Furthermore, the fast attention exhibits linear computational complexity, and the advantage of inference speed will become more obvious as the duration of the input speech increases. To verify the generalization ability and speed advantage of our model in long-form speech recognition, we evaluate the CER and inference speed of the En-HACN and Conformer-CTC models for utterances with different lengths. As shown in Fig. 8, compared with the Conformer-CTC, the En-HACN achieves 3% to 24% CER reduction in different utterance lengths, which proves the proposed En-HACN can improve the recognition accuracy by capturing the speech spatial information and text local information. The blue line in Fig. 8 shows that as the segment length of utterance becomes longer, En-HACN achieves 3.4% to 22% inference speedup compared to Conformer-CTC. We can see that introducing fast attention can solve the incompatibility between long speech sequences and the self-attention mechanism.

V. CONCLUSION

In this paper, we propose an Enhancing Hybrid Architecture with Fast Attention and Capsule Network (En-HACN) for E2E speech recognition. We add a novel CNN-Caps module before the encoder to complete the missing spatial information, such as the position relationships of pitch and formant frequency. Furthermore, we propose a hybrid structure LA-decoder to provide the local and global information of the target sequences during inference. Moreover, to better reduce the computational complexity to $O(L)$ concerning the sequence length L, we introduce fast attention into the En-HACN. As demonstrated on the aidatatang_200zh, Aishell-1, Thchs-30, HKUST, Librispeech, and Aishell-1-long. Our En-HACN has achieved state-of-the-art compared with existing works on Aishell-1 and Librispeech corpus, while having excellent generalization ability in long utterances. In the future, we will establish long utterances database to verify the generalization ability of our proposed method in real long utterance scenarios, which can avoid the problem of truncation or discontinuity in the synthesized long utterances.

REFERENCES

[1] A. Graves, "Connectionist temporal classification," in *Supervised Sequence Labelling With Recurrent Neural Networks*, Berlin, Germany: Springer, 2012, pp. 61–93.

[2] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 173–182.

[3] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4960–4964.

[5] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6645–6649.

[6] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[7] H. Miao, G. Cheng, P. Zhang, and Y. Yan, "Online hybrid CTC/attention end-to-end automatic speech recognition architecture," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1452–1465 2020.

[8] Q. Gao, H. Wu, Y. Sun, and Y. Duan, "An end-to-end speech accent recognition method based on hybrid CTC/Attention transformer ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 7253–7257.

[9] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5884–5888.

[10] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 5036–5040.

[11] T. Hori, N. Moritz, C. Hori, and J. L. Roux, "Transformer-based long-context end-to-end speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 5011–5015.

[12] T. Hori, N. Moritz, C. Hori, and J. L. Roux, "Advanced long-context end-to-end speech recognition using context-expanded transformers," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2097–2101.

[13] Y. Baosong et al., "Modeling localness for self-attention networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4449–4458.

[14] W. Huang, W. Hu, Y. T. Yeung, and X. Chen, "Conv-transformer transducer: Low latency, low frame rate, streamable end-to-end speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 5001–5005.

[15] C. Wang et al., "Semantic mask for transformer based end-to-end speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 971–975.

[16] M. A. Haidar, C. Xing, and M. Rezagholizadeh, "Transformer-based ASR incorporating time-reduction layer and fine-tuning with self-knowledge distillation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2102–2106.

[17] Z. Wang, Y. Ma, Z. Liu, and J. Tang, "R-transformer: Recurrent neural network enhanced transformer," 2020, *arXiv:1907.05572*.

[18] S. Li et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 5243–5253.

[19] S. Yang, H. Lu, S. Kang, L. Xie, and D. Yu, "Enhancing hybrid self-attention structure with relative-position-aware bias for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6910–6914.

[20] A. Katharopoulos, A. Vyas, and N. Pappas, "Transformers are RNNs: Fast autoregressive transformers with linear attention," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5156–5165.

[21] X. Wang, S. Sun, L. Xie, and L. Ma, "Efficient conformer with prob-sparse attention mechanism for end-to-end speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 4578–4582.

[22] A. Bie, B. Venkitesh, J. Monteiro, M.d. Akamal, and M. H. Rezagholizadeh, "A simplified fully quantized transformer for end-to-end speech recognition," 2020, *arXiv:1911.03604*.

[23] Y. Shi et al., "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6783–6787.

[24] T. G. Kang, H.-G. Kim, M.-J. Lee, J. Lee, and H. Lee, "Partially overlapped inference for long-form speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5989–5993.

[25] Z. Lu et al., "An empirical study of RNN-T and MWER training for long-form telephony speech recognition," 2021, *arXiv:2110.03841*.

[26] M. Zeineldeen et al., "Conformer-based hybrid ASR system for switchboard dataset," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7437–7441.

[27] C. -F. Zhang, Y. Liu, T. -H. Zhang, S. -L. Chen, F. Chen, and X. -C. Yin, "Non-autoregressive transformer with unified bidirectional decoder for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6527–6531.

[28] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6734–6738.

[29] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-CAPSNet: Capsule network with self-attention routing," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, 2021.

[30] S. Gu and Y. Feng, "Improving multi-head attention with capsule networks," in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.*, 2019, pp. 314–326.

[31] D. Peng, D. Zhang, C. Liu, and J. Lu, "BG-SAC: Entity relationship classification model based on self-attention supported capsule networks," *Appl. Soft Comput.*, vol. 91, 2020, Art. no. 106186.

[32] B. Jaesung and K. Dae-Shik, "End-to-end speech command recognition with capsule network," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, 2018, 776–780.

[33] X. Jie, L. Qijing, H. Kai, and Z. Mingying, "Classical and deep learning methods for speech command recognition," in *Proc. IEEE 9th Int. Conf. Inf., Commun. Netw.*, 2021, pp. 41–45.

[34] X. Wu et al., "Speech emotion recognition using capsule networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6695–6699.

[35] X. Wu et al., "Speech emotion recognition using sequential capsule networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3280–3291, 2021.

[36] Y. Bhanusree, T. V. V. Reddy, and S. K. Rao, "Capsule networks based acoustic emotion recognition using Mel cepstral features," in *Proc. IEEE Int. Conf. Innov. Comput., Intell. Commun. Smart Elect. Syst.*, 2021, pp. 1–7.

[37] K. J. Han, R. Prieto, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1D convolutions," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 54–61.

[38] A. Mohamed, D. Okhonko, and L. Zettlemoyer, "Transformers with convolutional context for ASR," 2019, *arXiv:1904.11660*.

[39] X. Tong, L. Yinqiao, Z. Jingbo, Y. Zhengtao, and T. Liu, "Sharing attention weights for fast transformer," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 5292–5298.

[40] J. W. Rae, P. Anna, S. M. Jayakumar, H. Chloe, and T. P. Lillicrap, "Compressive transformers for long-range sequence modelling," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–19.

[41] K. Nikita, K. Lukasz, and L. Anselm, "Reformer: The efficient transformer," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.

[42] W. Sinong, L. Z. Belinda, K. Madian, F. Han, and M. Hao, "Linformer: Self-attention with linear complexity," 2020, *arXiv:2006.04768*.

[43] B. Iz, E. P. Matthew, and C. Arman, "Longformer: The long-document transformer," 2020, *arXiv:2004.05150*.

[44] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. Assoc. Adv. Artif. Intell.*, 2021, pp. 11106–11115.

[45] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "FNet: Mixing tokens with fourier transforms," in *Proc. Int. Conf. Learn. Representations*, Seattle, USA, 2021, pp. 4296–4313.

[46] K. Choromanski et al., "Rethinking attention with performers," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–38.

[47] sox-web, "SoX - sound eXchange." 2012. [Online]. Available: http://sox.sourceforge.net

[48] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, and B. Zoph, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2613–2617.

[49] Z. Yao et al., "WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 4054–4058.

[50] R. Fan, W. Chu, P. Chang, and J. Xiao, "CASS-NAT: CTC alignment-based single step non-autoregressive transformer for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*2021, pp. 5889–5893.

[51] X. Song, Z. Wu, Y. Huang, C. Weng, D. Su, and H. Meng, "Non-autoregressive transformer ASR with CTC-Enhanced decoder input," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5894–5898. .

[52] R. Fan, W. Chu, P. Chang, J. Xiao, and A. Alwan, “An improved single step non-autoregressive transformer for automatic speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3715–3719.

[53] S. Zhang, E. Loweimi, P. Bell, and S. Renals, “Stochastic attention head removal: A simple and effective method for improving transformer based ASR models,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2541–2545.

[54] M. Li, C. Zorilă, and R. Doddipatla, “Head-synchronous decoding for transformer-based streaming ASR,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5909–5913.

[55] P. Guo et al., “Recent developments on ESPNet toolkit boosted by conformer,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5874–5878.

[56] N. Moritz, T. Hori, S. Watanabe, and R. J. Le, “Sequence transduction with graph-based supervision,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7212–7216.

[57] J. Yu et al., “FastEmit: Low-latency streaming ASR with sequence-level emission regularization,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6004–6008.

[58] N. Chen, P. Żelasko, J. Villalba, and N. Dehak, “Focus on the present: A regularization method for the ASR source-target attention layer,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5994–5998.

[59] T. Lohrenz, Z. Li, and T. Fingscheidt, “Multi-encoder learning and stream fusion for transformer-based end-to-end automatic speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2846–2850.

[60] M. Xu, S. Li, and X. -L. Zhang, “Transformer-based end-to-end speech recognition with local dense synthesizer attention,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 5899–5903.

[61] W. Zhou, Z. Zheng, R. Schlüter, and H. Ney, “On language model integration for RNN transducer based speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 8407–8411.

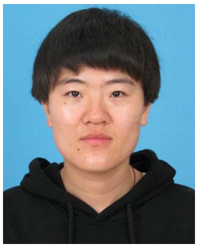
[62] G. Sun, C. Zhang, and P. C. Woodland, “Minimising biasing word errors for contextual ASR with the tree-constrained pointer generator,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 345–354, 2023.

[63] S. Kim et al., “Integer-only zero-shot quantization for efficient speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 4288–4292.

Boyang Lyu received the B.S. degree in electronic information engineering, and the M.Sc degree in electronic science and technology from Northwest Normal University, Lanzhou, China, in 2017 and 2020, respectively. He is currently working toward the Ph.D. degree in electronic science and technology with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include speech recognition and speech enhancement.

Chunxiao Fan (Member, IEEE) is currently a Professor and the Director of Center for Information Electronic and Intelligence System. She was a Member of Chinese Sensor Network Working Group. She was elevated to Evaluation Expert of Beijing Scientific and Technical Academy Awards. She has authored or coauthored more than 30 papers in international journals and conferences, authored and edited three books and authorized several invention patents. Her research interests include Heterogeneous media data analysis, Internet of Things, data mining, communication software.

Yue Ming (Member, IEEE) received the Ph.D. degree in signal and information processing from Beijing Jiaotong University, Beijing, China, in 2013. She was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA, between 2010 and 2011. Since 2013, she has been a Faculty Member with the Beijing University of Posts and Telecommunications, Beijing. She has authored more than 40 scientific papers. Her research interests include biometrics, computer vision, computer graphics, information retrieval, pattern recognition.



Panzi Zhao is currently working toward the Ph.D. degree with the Beijing Key Laboratory of Work Safety Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include computer vision, pattern recognition, and 3D face recognition.

Nannan Hu received the B.S. degree in communication engineering, and the M.Sc. degree in electronic science and technology from Shandong Normal University, Jinan, China, in 2016 and 2019. She is currently working toward the Ph.D. degree in electronic science and technology with the Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include object recognition, NLP and multi-modal information processing.