Multi-Granularity Sequence Alignment Mapping for Encoder-Decoder Based End-to-End ASR

Jian Tang, Jie Zhang, *Member, IEEE*, Yan Song, Ian McLoughlin, *Senior Member, IEEE*, and Li-Rong Dai, *Member, IEEE*

***Abstract*—Encoder-decoder based automatic speech recognition (ASR) methods are increasingly popular due to their simplified processing stages and low reliance on prior knowledge. Conventional encoder-decoder based approaches usually learn a sequence-to-sequence mapping function from the source speech to target units (e.g., subwords, characters) in an end-to-end manner. However, it is still unclear how to choose the optimal target unit, or granularity of multiple units. In general, as increasing the information available for learning sequence-to-sequence mapping functions can improve modeling effectiveness, we therefore propose a multi-granularity sequence alignment (MGSA) approach. This aims to enhance cross-sequence interactions between different granularity units for both modeling and inference stages in the encoder-decoder based ASR. Specifically, a decoder module is designed to generate multi-granularity sequence predictions. We then exploit the latent alignment mapping among units having different levels of granularity, by utilizing the decoded multi-level sequences as input for model prediction. The cross-sequence interaction can also be employed to re-calibrate output probabilities in the proposed post-inference algorithm. Experimental results on both WSJ-80 hrs and Switchboard-300 hrs datasets show the superiority of the proposed method compared to traditional multi-task methods as well as to single granularity baseline systems.**

***Index Terms*—Multi-granularity, sequence alignment, end-to-end ASR, encoder-decoder, post-inference, deep learning.**

I. Introduction

Automatic speech recognition (ASR) has improved tremendously in recent years thanks to advanced deep learning (DL) techniques. Traditional DL-based methods are mostly based on a hybrid architecture, which consists of several separately trained components using conditional independent approximations [1]. End-to-end based methods have been proposed recently to learn sequence-to-sequence mappings from source speech to target units. For example, in connectionist temporal classification (CTC) [2], recurrent neural network (RNN) transducer [3], segmental conditional random fields (SCRFs) [4], attention-based encoder decoder (AED) [5] and transformer methods [6]. These have achieved comparable or even better performance than traditional hybrid systems [7] due to the reduced reliance on prior information and benefit from simplified processing stages. Performance can be further improved by fusing different architectures under a framework such as multi-task learning (MTL) [8]. As a representative end-to-end model [9], we will adopt the AED as the basis for the derivation of the proposed multi-granularity sequence alignment method.

Sequence-to-sequence learning approaches involve mapping input acoustic frames to target units. These units can have different granularities, such as words [10], characters [5]. Intuitively, word-based targets are more natural and have been shown to be simpler and faster for decoding [11]. However, the large number of possible words leads to a large model size and high computational complexity in implementation. Moreover, the word-level modeling requires a large amount of training data. By contrast, character-level targets enjoy smaller model size and less extensive training, yet fail to exploit long-term context information effectively. Thus intermediate units (e.g., subwords) were proposed [12] to trade-off between model complexity and capability. However, the optimality of the target unit(s) size, and the corresponding granularity for end-to-end based ASR is still questionable.[1]

As more information is likely to be captured from multiple targets of different granularity, this can potentially improve the modeling capability. The MTL method was thus proposed to learn multiple sequence-to-sequence mappings jointly in [13]–[15]. Meanwhile, a multi-stage pre-training based method [16], [17] was proposed to improve the training efficiency. To combine results obtained from different granularities, we can simply use score fusion [18]–[20]. Although this improves the ASR performance compared to single-granularity approaches, few of them take account of relationships between sequences of different granularity. Actually, there exists a latent alignment mapping between two sequences, as shown in Fig. 1. For instance, the text “of course not” can be represented as a subword sequence

Manuscript received November 26, 2020; revised March 17, 2021; accepted July 13, 2021. Date of publication August 6, 2021; date of current version September 2, 2021. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant WK2100000016, in part by the National Key R&D Program under Grant 2017YFB1002202, in part by the National Natural Science Foundation of China under Grants U1613211 and U1836219, and in part by the Key Science and Technology Project of Anhui Province under Grant 18030901016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lei Xie. *(Corresponding author: Jie Zhang.)*

The authors are with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China (USTC), Hefei 230026, China (e-mail: enjtang@mail.ustc.edu.cn; jzhang6@ustc.edu.cn; songy@ustc.edu.cn; ian.mcloughlin@singaporetech.edu.sg; lrdai@ustc.edu.cn).

Digital Object Identifier 10.1109/TASLP.2021.3101921

[1] Consider a simple example to illustrate the impact of different granularities. Given the same input speech, a character-level hypothesis might give a recognition result of “He adapt a dog” whereas word-level transcription yields “H E <space > A D O P T E D < space> A”. In this case neither are correct, but cross-verification would correctly yield “He adopted a dog.”

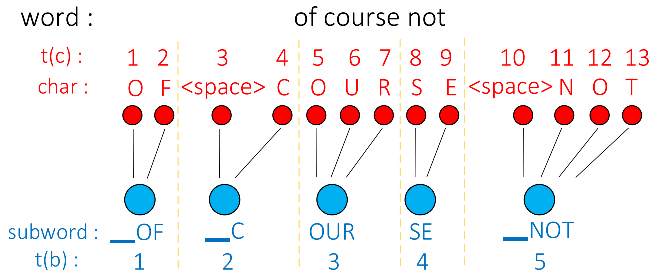
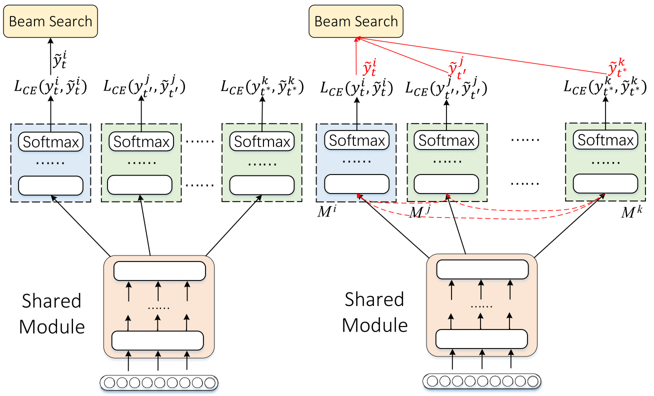


Fig. 1. An example of the alignment mapping relationship between multi-granularity sequences; subword unit “OUR” corresponds to the single character sub-sequence “O,” “U” and “R”. The units in multi-granularity sequences (e.g., “OUR“ or “O”) are referred to as “tokens” in this work.


Beam Search
$\tilde{y}_t^i$
$L_{CE}(y_t^i, \tilde{y}_t^i)$
$L_{CE}(y_{t'}^j, \tilde{y}_{t'}^j)$
$L_{CE}(y_{t^*}^k, \tilde{y}_{t^*}^k)$
Softmax
Softmax
Softmax
Shared Module
Beam Search
$\tilde{y}_t^i$
$\tilde{y}_{t'}^j$
$\tilde{y}_{t^*}^k$
$L_{CE}(y_t^i, \tilde{y}_t^i)$
$L_{CE}(y_{t'}^j, \tilde{y}_{t'}^j)$
$L_{CE}(y_{t^*}^k, \tilde{y}_{t^*}^k)$
Softmax
Softmax
Softmax
M^i
M^j
M^k
Shared Module

(a) the traditional multi-task method

(b) the proposed MGSA method

Fig. 2. The proposed MGSA approach for end-to-end ASR, where the alignment mapping information is applied for use in both architecture construction (the red dashed line) and output re-scoring (the solid red line).

“_OF _C OUR SE _NOT,” or as “O F <space > C O U R S E < space> N O T” at character-level. The mapping from subwords to characters, termed the *alignment mapping* in this work, can explicitly indicate the relationship between sequences.

Contributions: In this work, we propose a novel multi-granularity sequence alignment (MGSA) approach for the AED based ASR, which is based on the use of the alignment mapping information between multi-granularity (MG) units. The end-to-end ASR can be divided into training and inference stages, and the use of the alignment mapping is considered in *both* stages. The general framework of the proposed MGSA method is shown in Fig. 2(b). Compared to the commonly-used MTL-based multi-granularity end-to-end approach in Fig. 2(a), there are three main differences. Firstly, the alignment mapping information is estimated based on the joint optimization of the MG conditional posterior probabilities. Secondly, a new decoder module is used to merge the historical contents of the alignment mapping information. The resulting inherently MG information can then be used by the decoder to generate multiple predictions for MG units. As such, the interaction and fusion process of the MG units are exploited in the model architecture (the red dashed lines in Fig. 2). Finally, the MG information is also adopted by the end-to-end post-inference algorithm (the solid red lines in Fig. 2). For example, after the model M^i generates one token y_t^i, we can transform it to a new expression $y_{t'}^j$ using the obtained alignment mapping information. Using the model M^j to generate another sequence, the corresponding hypothesis score $S_{t'}^k$ is then used to verify and rectify the output prediction y_t^i.

The proposed MGSA method is evaluated on two ASR benchmark corpora (WSJ [21] and Switchboard [22]) and yields a performance improvement over both traditional single-granularity baseline and MTL approaches. On the WSJ corpus, the proposed method can reduce the word error rate (WER) by 2.2% (from 11.1% baseline to 8.9%). For eval2000 test set of the Switchboard corpus, the WER is reduced by 2.6% (from 17.3% to 14.7%). Although the proposed MGSA method is built on the original AED architecture, it is also compatible with other modified frameworks, such as transformer.

The remainder of this paper is organized as follows. Section II discusses related works. Section III explores the joint probability optimization, which is the theoretical foundation of the proposed MGSA model. In Section IV, the proposed MGSA approach is detailed, including the sequence alignment based encoder-decoder design and the post-inference algorithm. Experimental results are presented in Section V, and finally Section VI concludes this work.

II. Related Works

The end-to-end ASR makes use of a neural network to map an acoustic sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$ of length L to a text sequence $\boldsymbol{y} = (y_t \in \mathcal{U} | t = 1, \ldots, T)$ of length T, where $\mathcal{U}$ denotes the set of target units. Statistically, the objective of end-to-end modeling is to learn the conditional posterior probability distribution $P_\theta(\boldsymbol{y}|\boldsymbol{x})$,

$$P_\theta(\boldsymbol{y}|\boldsymbol{x}) = p(y_1, y_2, \ldots, y_T | x_1, x_2, \ldots, x_L), \tag{1}$$

where θ contains the model parameters. In general, the text sequence is categorized into basic units (e.g., phoneme, character, pinyin) [5], [23], [24], intermediate units (e.g., subword, wordpiece, phone-based subwords) [12], [20], [25] or word-level units [26]. Clearly, these three categories have different granularities. Such different granularity targets have been used in end-to-end ASR, e.g., phonemes, characters [2], [5], [27], subwords [28] and words [17], [26], [29], [30], [30].

A. Single Target Encoder-Decoder Architecture

As a typical end-to-end model, we consider the encoder-decoder architecture as an example to explain the training and inference stages in the single target ASR system. Given an input sequence $\boldsymbol{x}$, the conditional probability $P_\theta(\boldsymbol{y}|\boldsymbol{x})$ in the AED can be decomposed using the chain rule as,

$$P_\theta(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{T} P_\theta(y_t | y_{t-1}, s_{t-1}; \boldsymbol{x}), \tag{2}$$

where s_{t-1} denotes the state of the decoder at the previous step. In general, an AED encoder exploits input sequence $\boldsymbol{x}$ to produce a high-level representation, which will be encoded into the continuous vector $\bar{\boldsymbol{x}}$. Then the decoder iteratively generates the discrete target sequence $\boldsymbol{y}$ [27]. It is clear from (2) that, at each output position, the decoder predicts an output discrete token y_t based on the encoder representation $\bar{\boldsymbol{x}}$, the historical token y_{t-1} and the previous decoder state s_{t-1}. Note that the historical token y_{t-1} is not the same during model training and inference stages. During training, the decoder is conditioned on the true, known, prefix token y_{t-1}, whereas during inference, we can only use an assumed surrogate, say $\tilde{y}_{t-1}$. Thus, the posterior probability at the inference stage should be slightly modified based on (2). Given speech $\boldsymbol{x}$, the model searches the most likely token $\hat{\boldsymbol{Y}}$ using a beam-search algorithm at the inference stage, e.g. in [31],

$$\hat{\boldsymbol{Y}} = \arg\max_{\tilde{y}\in\mathcal{U}} \log p(\tilde{\boldsymbol{y}}|\boldsymbol{x}), \tag{3}$$

During beam-search, the model calculates the score of each hypothesis, which is defined as the logarithmic probability of the assumed token sequence. The number of remaining hypotheses is limited by a predefined number, i.e., the beam size, which dramatically affects the searching efficiency. The score for the hypothesis $\hat{\boldsymbol{Y}}$ can be recursively computed as

$$\begin{aligned}\hat{\boldsymbol{Y}} &= \arg\max_{\tilde{y}\in\mathcal{U}} \log \prod_t p(\tilde{y}_t|\tilde{y}_{t-1};\boldsymbol{x})\\ &= \arg\max_{\tilde{y}\in\mathcal{U}} \sum_t \log p(\tilde{y}_t|\tilde{y}_{t-1};\boldsymbol{x}).\end{aligned} \tag{4}$$

Instead of choosing a certain target unit, combining multiple target sequences can improve the modeling capability. As more multi-granularity units are considered, more information on the true audio content can be leveraged. Therefore, a variety of strategies to integrate multiple target units into such end-to-end ASR have been investigated, e.g., multi-task learning strategy, pre-training methods and output score fusion.

B. Multi-Granularity End-to-End Modeling

Existing multi-task methods can learn useful intermediate representations among all inputs, and those target units might be complementary to each other [13]–[15]. In [13], different training strategies were explored for building char-to-subword models one block at each time slot. In [14], an intermediate representation was used as an auxiliary supervision at lower levels to combine the advantages of end-to-end training and a traditional pipeline strategy. In addition, some multi-task models were presented in [15] for simultaneous signal-to-grapheme and signal-to-phoneme conversions, while sharing the encoder parameters.

The second integration category is to use the intermediate target to initialize or to assist the training process for the word-level target, which can reduce the dependence on the amount of the transcribed data [16], [17]. As the detection of subwords provides a robust starting point for detecting words, the subword-based model was used as the initialization of word-based modeling in [17]. In [16], a refined multi-stage multi-task training strategy was presented to improve the AED modeling performance. This used multiple encoder modules, corresponding to multiple target units, with each module exploring a different pre-training method for the encoder, including transfer learning from a different-level encoder. Though differing in implementation, the optimization objective functions in [16], [17] are similar. Taking a target unit pair, $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$, as an example, the objective function is

$$p(\boldsymbol{y}^i;\boldsymbol{y}^j|\boldsymbol{x}) = p(\boldsymbol{y}^i|\boldsymbol{x})p(\boldsymbol{y}^j|\boldsymbol{x}), \tag{5}$$

under the assumption that the two target units are independent. However, in practice this assumption on independence may not hold, resulting in a rather limited performance gain when applying such methods.

The third integration category is to use multi-level score fusion to integrate the scores obtained from different target units in end-to-end modeling [18]–[20]. Hori *et al.* suggested combining the predictions of a word-based language model (LM) with a character-based one at the inference stage, yielding a significant performance improvement over character-only methods [18]. Specifically, hypotheses are first scored using the character-based LM until a word boundary is encountered. Words that are already known are then re-scored using the word-based LM, while the character-based LM provides for the out-of-vocabulary score. This method can effectively exploit the benefits of character-based open vocabulary recognition and overcome the weak modeling of character-based LM using the word-based LM. However, an additional LM and some post-processing operations are required after building the end-to-end model. Another attempt is to directly combine the outputs from different targets. For instance, Wang *et al.* [20] developed a one-pass beam-search algorithm to efficiently combine predictions of both subword and phone-based targets. In this method, when a word boundary in the phone-based subword prediction is encountered, the token is decomposed into a subword sequence. This is used by an auxiliary system to validate or rectify the prediction. Clearly, this method only considered the correspondence between special tokens (e.g., word boundaries). However we note that much more correspondences may exist between other tokens in multiple target sequences. The utilization of multiple target sequences was thus not sufficient, since the information contained in one granularity, but not in other granularities, was not fully exploited. This might limit the ASR performance gain.

The MGSA framework proposed in this paper aims to better exploit correspondence between multiple target sequences.

III. Probability Optimization

Given three types of target sequence (e.g., basic units, intermediate units, word labels), smaller scaled units can be clustered to form larger scale units, which might correspond to one or more tokens in the former. In Fig. 3, we show these three target units in an example that illustrates how word token "COURSE" can correspond to a subword sub-sequence "_C OUR SE," while the subword token "OUR" also uniquely maps to sub-sequence "O U R". We can also observe a latent mapping relationship between the target sequences.

For two target sequences $\boldsymbol{y}^i = (y_1^i, y_2^i, \ldots, y_T^i)$ and $\boldsymbol{y}^j = (y_1^j, y_2^j, \ldots, y_N^j)$, where T and N denote the length of the two

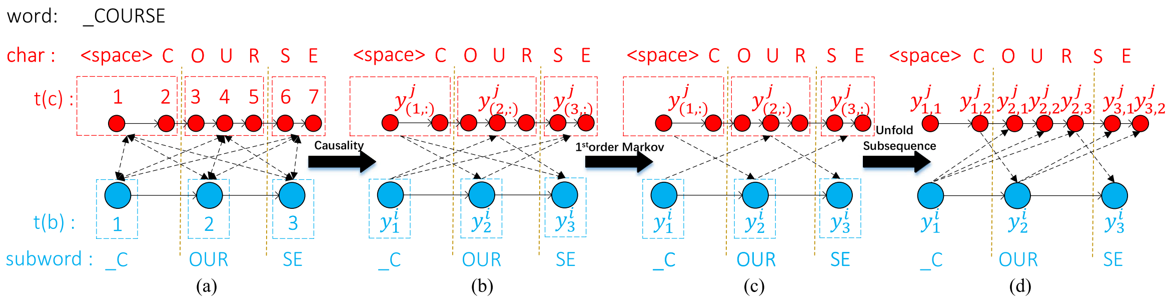

word: _COURSE
char : <space> C O U R S E
t(c) :
1 2 3 4 5 6 7
$y^j_{(1,:)}$ $y^j_{(2,:)}$ $y^j_{(3,:)}$
$y^j_{1,1}$ $y^j_{1,2}$ $y^j_{2,1}$ $y^j_{2,2}$ $y^j_{2,3}$ $y^j_{3,1}$ $y^j_{3,2}$
Causality
1storder Markov
Unfold Subsequence
t(b) :
1 2 3
y^i_1 y^i_2 y^i_3
subword : _C OUR SE
(a) (b) (c) (d)

Fig. 3. An illustration of the joint optimization probability in multi-granularity end-to-end modeling, where the sequences of character and subword are denoted by $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$, respectively: (a) The original graphical model, (b) the simplified model using causality, (c) the simplified model using the first-order Markov process, and (d) the final character subsequence unfolding based model.

sequences, respectively, each token y_t^i in the target sequence $\boldsymbol{y}^i$ might correspond to one or more tokens in the other target sequence $\boldsymbol{y}^j$. Let $y_{(t,:)}^j$ denote the sub-sequence corresponding to the t-th token in $\boldsymbol{y}^i$. Let the number of tokens contained in $y_{(t,:)}^j$ be denoted by k_t, and the u-th token in the sub-sequence $y_{(t,:)}^j$ by $y_{t,u}^j$, respectively. For example, y_2^i and $y_{(2,:)}^j$ in Fig. 1 represent the subword token "OUR" and character sub-sequence "O U R," respectively. Based on these definitions, the target sequence $\boldsymbol{y}^j$ can be equivalently rewritten as

$$\boldsymbol{y}^j \overset{(a)}{=} (y_1^j, y_2^j, \ldots, y_N^j) = (y_{(1,:)}^j, \ldots, y_{(t,:)}^j, \ldots, y_{(T,:)}^j)$$
$$\overset{(b)}{=} ([y_{1,1}^j \ldots, y_{1,k_1}^j], [y_{2,1}^j \ldots, y_{2,k_2}^j], \ldots, [y_{T,1}^j \ldots, y_{T,k_T}^j]), \tag{6}$$

where (b) shows the latent alignment mapping relationship between two target sequences (i.e., $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$).

Given an acoustic sequence $\boldsymbol{x}$, our goal is to model the joint conditional distribution $P_\theta(\boldsymbol{y}^i; \boldsymbol{y}^j|\boldsymbol{x})$. For the target sequence $\boldsymbol{y}^j$ given in (6), each sub-sequence $y_{(t,:)}^j$ maps to one token in sequence $\boldsymbol{y}^i$. For simplicity, let the token-pair of y_t^i and the corresponding substring $y_{(t,:)}^j$ be denoted by u_t^p. Thus, the joint conditional distribution can be expressed as

$$\begin{aligned} P_\theta(\boldsymbol{y}^i; \boldsymbol{y}^j|\boldsymbol{x}) &= p(y_1^i, \ldots, y_T^i; y_1^j, \ldots, y_N^j|\boldsymbol{x}) \\ &= p(y_1^i, y_{(1,:)}^j, \ldots, y_T^i, y_{(T,:)}^j|\boldsymbol{x}) \\ &= p(u_1^p, u_2^p, \ldots, u_i^p, \ldots, u_T^p|\boldsymbol{x}). \end{aligned} \tag{7}$$

Due to the correlation between multiple target sequences, the tokens in this joint conditional probability can influence each other. Fig. 3(a) shows a graphic representation of (7).

Considering the causality between the target sequences $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$, two types of interactions in Fig. 3(a) should be avoided. One is that the current tokens in one granularity target sequence should be independent on future tokens in the other sequence, since $y_{(t,:)}^j$ does not affect the prediction of y_{t-1}^i. This is called causality across tokens. The other is that for each token-pair $u_t^p = [y_t^i, y_{(t,:)}^j]$, one token should have no effects on the prediction of the other, e.g., token-pair, $y_{(2,:)}^j$ and y_2^i in Fig. 3 are of different granularities, but describe the same token "OUR". As the interaction between tokens inside a token-pair is directly related to optimizing the probability $p(y_t^i|y_{(t,:)}^j)$, in case of modeling the dependency inside the token pair, the neural network would directly copy the output from other granularity in the training process. The resulting model would totally depend on the text information of other granularities, the encoder module and the attention module will not be well-trained. Thus, the correlation in token-pair, like ["OUR"–"O U R"], should not give any output prediction for each token (e.g., "OUR") within this token-pair, and we can thus omit the dependency insider a token-pair. In order to avoid such situations, we re-write $P_\theta(\boldsymbol{y}^i; \boldsymbol{y}^j|\boldsymbol{x})$ as

$$\begin{aligned} P_\theta &= p(u_1^p) \ldots p(u_T^p|u_1^p \ldots u_{T-1}^p) \\ &= p(u_1^p) \ldots p(y_T^i|u_1^p \ldots u_{T-1}^p) p(y_{(T,:)}^j|u_1^p \ldots u_{T-1}^p), \end{aligned} \tag{8}$$

where $(\boldsymbol{y}^i; \boldsymbol{y}^j|\boldsymbol{x})$ is omitted for notational brevity. Considering the causality and based on (8), we can thus simplify the graphical model in Fig. 3(a) to Fig. 3(b).

Assuming that the output variables follow the first-order Markov random process (i.e., the current prediction is only affected by the latest token-pair), the joint conditional probability in (8) can be further simplified as

$$\begin{aligned} P_\theta &= p(y_1^i) p(y_{(1,:)}^j) \ldots p(y_{(T,:)}^j|u_{T-1}^p) p(y_T^i|u_{T-1}^p) \\ &= p(y_1^i) \prod_{t=2}^{T} p(y_t^i|y_{t-1}^i; y_{(t-1,:)}^j) p(y_{(t-1,:)}^j|y_{t-2}^i; y_{(t-2,:)}^j), \end{aligned} \tag{9}$$

where $y_{(t,:)}^j$ consists of k_t tokens. Similarly, (9) can be graphically illustrated using Fig. 3(c). Based on this, the prediction process for sub-sequence $p(y_{(t-1,:)}^j|y_{t-2}^i; y_{(t-2,:)}^j)$ can be unfolded via the chain rule as

$$P_\theta = \prod_{t=1}^{T} \{p(y_t^i|y_{t-1}^i; y_{(t-1,-1)}^j) V(y_{(t-1,:)}^j)\}, \tag{10}$$

where $y_{(t-1,-1)}^j$ represents the last token in target subsequence $y_{(t-1,:)}^j$, and the transition function $V(y_{(t-1,:)}^j)$ is given by

$$V(y_{(t-1,:)}^j) = \prod_{u=1}^{k_{t-1}} p(y_{t-1,u}^j|y_{t-2}^i; y_{t-1,u-1}^j). \tag{11}$$

The unfolding process is graphically shown in Fig. 3(d). This implies that the joint optimization of two target sequences should take both history information y^i_{t-1} and $y^j_{(t-1,:)}$ into account. In (10) and (11), in case $t-1=0$ or $u-1=0$, both y^s_0 and y^c_0 will be set to be 'sos' as the traditional end-to-end ASR models. Note that although (10) is built on the basis of two target sequences, the extension to three or more categories is straightforward. From the probability analysis above, we can conclude that there are conditions that need to be satisfied to enable multi-granularity end-to-end modeling:

- **Mapping relation**: A strict one-to-many mapping relationship between MG target sequences is the basis of the joint optimization.
- **Independence**: For each item of history information $y^v_{t-1}, v \in \{i, j\}$, its historical modeling ability should be guaranteed, and the influence of history tokens from other granularity targets should be avoided to ensure independence between the historical states.
- **Interaction prediction**: For each target sequence, the information from other granularities should directly affect the output prediction.

In practice, it might be effective to ignore the information transmission in one direction, namely from a subword to characters. In this case, (11) can be simplified as

$$V(y^j_{(t-1,:)}) = \prod_{u=1}^{k_{t-1}} p(y^j_{t-1,u} | y^j_{t-1,u-1}). \tag{12}$$

Note that when applying (11), the alignment mapping information is taken into account in the calculation of sub-sequences $y^j_{(t-1,:)}$, and the interaction between two different granularity target sequences is bi-lateral (e.g., see Fig. 3(d)). However, the simplification in (12) makes that interaction uni-lateral, e.g., only from a lower granularity $\boldsymbol{y}^j$ to a larger one $\boldsymbol{y}^i$ Both the bi-lateral and uni-lateral interactions are considered in this work and experimentally compared in Section V-A. Note that in principle there exists a potential interaction from a larger granularity to a lower one, while it was shown that in general for the large vocabulary continuous speech recognition, a larger granularity unit is more robust than the lower one, and thus more suitable for unit modeling [32]. Therefore, in the considered uni-lateral interaction, we chose y^i as the main granularity unit.

IV. The Proposed MGSA Framework

Based on the traditional AED architecture and the theoretical analysis in Section III, we now present the proposed MGSA framework, which consists of an alignment attention based encoder-decoder design and a post-inference process.

A. Attention Based Encoder-Decoder Design

For brevity, we choose two categories, the subword $\boldsymbol{y}^b$ and the character $\boldsymbol{y}^c$, corresponding to the target sequences $\boldsymbol{y}^i$ and $\boldsymbol{y}^j$, respectively, to introduce the proposed MGSA method. For a set of speech utterances parameterized into feature vector $\boldsymbol{x}$, we use $\boldsymbol{y}^b$ and $\boldsymbol{y}^c$ to represent the true subword and true character label sequence, respectively. The proposed MGSA framework is shown in Fig. 4. The encoder produces a high-level representation encoded in the continuous vector $\bar{\boldsymbol{x}}$, and the decoder generates subword predictions y^b_t by choosing the relevant elements of hidden state at the t-th time step. To explain this, we take the generation of the subword token (e.g., "SE" in Fig. 4) at the t-th step as an example. To calculate the subword prediction, the character sub-sequences $y^c_{(t-1,:)} = (y^c_{t-1,1}, \ldots, y^c_{t-1,k_{t-1}})$, which correspond to the subword at the previous step, have to be provided. In Fig. 4, the character predictions at step $(t-1)$ consist of {"O," "U," "R"}.

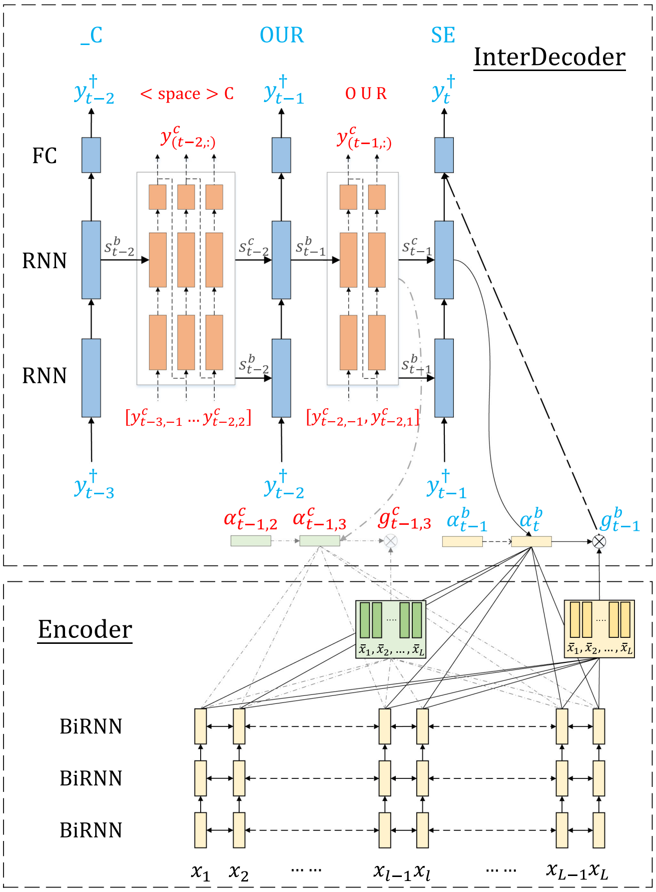


Fig. 4. The proposed MGSA framework for predicting the word "_COURSE," showing the encoder module, two decoder modules (one for subwords, and one for characters), and three attention mechanisms – where the interaction attention α^i_t is omitted for simplification. The decoder process for subwords and characters is alternately performed. FC and $\otimes$ represent a fully connected layer and element-wise multiplication respectively. Dash-dotted lines denote a copy-and-paste operation.

For the u-th token in the character sub-sequence $y^c_{(t-1,:)}$, first we perform the state update and attention alignment. In detail, the decoder updates the current state $s^c_{t-1,u}$ based on the output from the previous character step using

$$s^c_{t-1,u} \leftarrow \mathrm{RNN}(s^c_{t-1,u-1}, y^c_{t-1,u-1}), \tag{13}$$

where RNN represents a recurrent neural network (RNN) layer. The decoder state $s^c_{t-1,u}$ together with $\alpha^c_{t-1,u-1}$ are then provided for calculating the current alignment score $\alpha^c_{t-1,u}$. Due to the monotonicity of alignment in ASR, we use location-based attention in this work [5],

$$(g^c_{t-1,u}, \alpha^c_{t-1,u}) \leftarrow \mathrm{Attend}(s^c_{t-1,u}, \alpha^c_{t-1,u-1}, \bar{\boldsymbol{x}}), \tag{14}$$

where the Attend module returns the most generic attention.

For output character prediction, the decoder re-updates its state based on the obtained glimpse vector $g^c_{t-1,u}$ using

$$s^c_{t-1,u} \leftarrow \mathrm{RNN}(s^c_{t-1,u}, g^c_{t-1,u}). \tag{15}$$

The obtained character state $s^c_{t-1,u}$ is then applied to produce the prediction of $y^c_{t-1,u}$, which is given by

$$p(y^c_{t-1,u}|s^c_{t-1,u}, s^b_{t-2}) = \mathrm{softmax}(W^c[s^c_{t-1,u}; s^b_{t-2}]), \tag{16}$$

when the transition function in (11) is used, or given by

$$p(y^c_{t-1,u}|s^c_{t-1,u}) = \mathrm{softmax}(W^c[s^c_{t-1,u}]), \tag{17}$$

when the simplified transition function in (12) is used. Note that the matrix W^c can be trained in practice, and the bias variable in the fully connection (FC) layer is omitted for simplicity. This iterative procedure will be terminated when the predictions for all tokens in $y^c_{(t-1,:)}$ are obtained.

Given the prediction $y^c_{(t-1,:)}$, the subword decoder is therefore triggered to predict the t-th subword. Let the decoder state of the latest character be denoted by $s^c_{t-1,k_{t-1}}$, which contains the history token of the character sequence s^c_{t-1}. For the subword prediction, it consists of two steps: 1) updating the decoder state following

$$s^b_t \leftarrow \mathrm{RNN}(s^b_{t-1}, y^b_{t-1}), \tag{18}$$

and 2) calculating the attention vector using

$$(g^b_t, \alpha^b_t) \leftarrow \mathrm{Attend}(s^b_t, \alpha^b_{t-1}, \bar{\boldsymbol{x}}). \tag{19}$$

The difference from the character attention lies in that the attention weight and glimpse vector included in the subword attention are denoted by α^b_t and g^b_t, respectively.

Unlike the traditional AED structure, in order to model the effects of the character state s^c_{t-1} on the subword prediction, we use an interaction module, which consists of one attention mechanism and two RNN layers. With respect to the interaction module, combining the previous subword token y^b_{t-1} and the character decoder state s^c_{t-1} results in the interactive state s^i_t, which is given by

$$s^i_t \leftarrow \mathrm{RNN}(s^c_{t-1}, y^b_{t-1}), \tag{20}$$

as the output of an RNN layer. The state s^i_t is then used to calculate the interaction attention as

$$(g^i_t, \alpha^i_t) \leftarrow \mathrm{Attend}(s^i_t, \alpha^i_{t-1}, \bar{\boldsymbol{x}}). \tag{21}$$

Using s^i_t as a query vector, the interaction attention enables to extract the interactive glimpse g^i_t, which might contain a certain amount of complementary information with respect to g^b_t. By including the interaction module, the proposed model contains three attention mechanisms: character (14), subword (19) and interaction (21) attention.

The subword state s^b_t is the combination of the previous subword state s^b_{t-1} and the history output token y^b_{t-1}, while the character state s^c_t is the integration of character state s^c_{t-1} and y^b_{t-1}. Due to the fact that s^b_t and s^c_t occur at the same time step, the final prediction can be refined through state fusion, e.g., the gated linear unit (GLU) [33] as

$$f^i_t = \sigma(\mathrm{FC}(s^i_t)) + s^b_t, \tag{22}$$

where $\sigma(\cdot)$ represents a sigmoid activation. Next, the interactive state is updated by

$$s^i_t \leftarrow \mathrm{RNN}(f^i_t, g^i_t), \tag{23}$$

Given the fusion variables f^i_t, we can further apply an RNN layer to update the interdecoder state using the interactive glimpse vector g^i_t.

Finally, the interactive decoder state s^i_t is used to estimate the primary subword output $p(\tilde{y}^\dagger_t|s^i_t)$ of the t-th time step, which is given by

$$p(\tilde{y}^\dagger_t|s^i_t) = \mathrm{softmax}(W^i s^i_t), \tag{24}$$

Apart from the primary output, s^b_t can be applied to simultaneously obtain a secondary subword output $p(\tilde{y}^b_t|s^b_t)$, which is given by

$$p(\tilde{y}^b_t|s^b_t) = \mathrm{softmax}(W^b s^b_t). \tag{25}$$

Note that both the matrices W^i and W^b can be trained in practice. The operations in (18)–(25) constitute the complete decoding process for the t-th subword token. In combination with the calculation of character sub-sequences, which corresponds to the subword at time step $(t-1)$, we can obtain the whole decoding process for time step t. The proposed decoder module i.e., (13)–(25), is named by *interdecoder*. When (16) is used for character classification, the decoder module is called *bi-interdecoder*; when the simplified version (17) is used, it is termed by *uni-interdecoder*. In both cases, the interdecoder can simultaneously generate three types of output: the character sub-sequence $\tilde{y}^c_{(t-1,:)}$, the primary subword $\tilde{y}^\dagger_t$ and the secondary subword $\tilde{y}^b_t$. Considering the prediction for one training sequence, the frame-level cross entropy (CE) based loss function can therefore be formulated as a summation of three components, i.e.,

$$\begin{aligned}\mathrm{Loss} = &\sum_{t=1}^{T}\sum_{u=1}^{k_t} L_{\mathrm{CE}}(y^c_{t,u}, \tilde{y}^c_{t,u}) \\ &+ \sum_{t=1}^{T} L_{\mathrm{CE}}(y^b_t, \tilde{y}^\dagger_t) + \lambda\sum_{t=1}^{T} L_{\mathrm{CE}}(y^b_t, \tilde{y}^b_t),\end{aligned} \tag{26}$$

where $\lambda \in [0,1]$ is a balancing hyper parameter, and the first two terms refer to the losses of character and subword targets, respectively, while the third term guides subword attention.

In summary, the proposed MGSA approach for the subword prediction at the t-th step can be structured into four parts: (a) a char block for generating the prediction of the character sub-sequence at step $(t-1)$, in which the character decoder state s^c_{t-1} is estimated, (b) a subword state block to update the hidden state s^b_t of the subword decoder and calculate the corresponding attention vector α^b_t, (c) an interaction block to fuse the decoding states s^b_t and s^c_{t-1} by a GLU and to calculate attention score α^i_t and content vector g^i_t, and (d) a subword classification block for predicting the subword under the utilization of the interactive

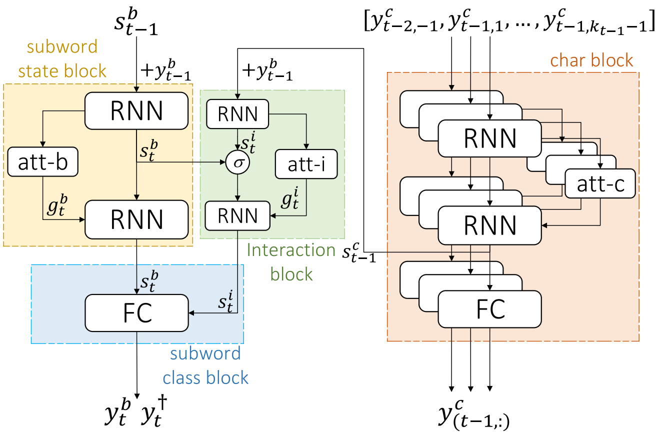


Fig. 5. An example of the MGSA subword token prediction at time step t, consisting of four blocks highlighted using different colours.

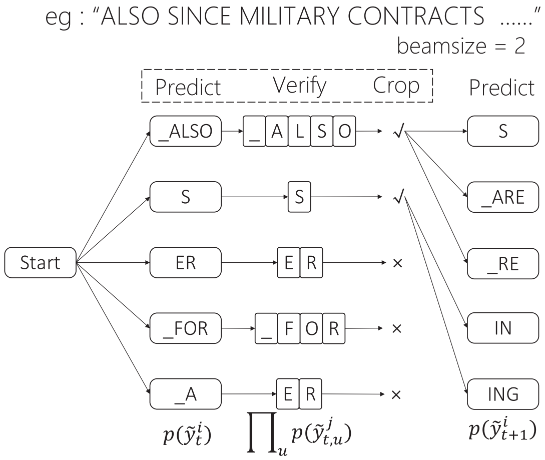


Fig. 6. The proposed post-inference algorithm, consisting of Predict, Verify and Crop blocks.

states s^i_t. The complete structure, including interconnection between blocks, is depicted in Fig. 5.

B. Post-Inference

From the description of the proposed MGSA framework, it is clear that the interdecoder module can utilize both the subword and character-level historical tokens for prediction, while the traditional decoder only uses subword history information. The proposed interdecoder is capable of alignment mapping, which can also be exploited for the end-to-end inference stage. Based on this alignment mapping information, we propose a post-inference algorithm for the final inference. For the inference at the t-th step, in case a candidate output $\tilde{y}^b_t$ is obtained, the corresponding sub-sequence $\tilde{y}^c_{(t,:)}$ can be determined. For instance, given a subword candidate output "SE," the character sub-sequence will be "S E". The sub-sequence can then be used to cross verify the candidate output.

The inference in the end-to-end ASR is performed by synchronous output-label decoding using beam search [18]. The decoder computes the score for each remaining hypothesis, which is defined as the logarithmic probability, given by

$$\hat{\boldsymbol{Y}}^i = \arg\max_{\tilde{y}^i \in \mathcal{U}} \log p(\tilde{\boldsymbol{y}}^i | \boldsymbol{x})$$

$$= \arg\max_{\tilde{y}^i \in \mathcal{U}} \sum_t \log p(\tilde{y}^i_t | \tilde{y}^i_{t-1}, \boldsymbol{x}) \tag{27}$$

$$= \arg\max_{\tilde{y}^i \in \mathcal{U}} \frac{1}{2} \sum_t \log p(\tilde{y}^i_t | \tilde{y}^i_{t-1}) + \log p(\tilde{y}^i_t | \tilde{y}^i_{t-1}) \tag{28}$$

$$= \arg\max_{\tilde{y}^i \in \mathcal{U}} \frac{1}{2} \sum_t \log p(\tilde{y}^i_t | \tilde{y}^i_{t-1}) + \log p(\tilde{y}^j_{(t,:)} | \tilde{y}^j_{(t-1,:)}) \tag{29}$$

$$= \arg\max_{\tilde{y}^i \in \mathcal{U}} \frac{1}{2} \sum_t \log p(\tilde{y}^i_t | \tilde{y}^i_{t-1}) + \log \Big(\prod_u p(\tilde{y}^j_{t,u} | \tilde{y}^j_{t,u-1}) \Big) \tag{30}$$

where the variable $\boldsymbol{x}$ is omitted for clarity in (28, 29, 30). Note that (28) is obtained by dividing each element in the summation of (27) into two equal parts, and (29) is obtained by introducing the score of the character subsequence into the inference (if two subword units obtain a comparable score, their character subsequences are more likely to be different). Finally, (30) is obtained by expanding the probability function $p(\tilde{y}^j_{(t,:)} | \tilde{y}^j_{(t-1,:)})$.

Moreover, the proposed post-inference algorithm can be easily generalized into other end-to-end models, since only multi-granularity prediction probabilities are required. An illustrative example of predicting the subword and character, $\tilde{y}^b_t$ and $\tilde{y}^c_{(t,:)}$ is shown in Fig. 6, which consists of three blocks: Predict, Verify and Crop. In the Predict block, the subword decoder calculates the candidate output prediction $\tilde{y}^b_t$ at time step t. In Verify, the candidate subword $\tilde{y}^b_t$ is uniquely matched to one character sub-sequence $\tilde{y}^c_{(t,:)} = (\tilde{y}^c_{t,1}, \dots, \tilde{y}^c_{t,k_t})$, and the probability of generating sub-sequence $\tilde{y}^c_{(t,:)}$ is calculated. As such, the candidate hypotheses of subwords can be verified and rectified instead of generating new hypotheses. The Crop block refines the likelihood score by excluding outliers that have a much lower score.

Note that both post-inference and the interdecoder module use the alignment mapping information, but at different phases. The difference in the context of decoding y^i_t is illustratively explained in Fig. 7. For post-inference, the subsequence $y^j_{(t,:)}$ can be further applied to verify and rectify the predicted output in Fig. 7(a), while the history output token of time step $(t-1)$ is used in the interdecoder module in Fig. 7(b). It is clear that the alignment mapping is exploited at different time steps. Therefore, the proposed MGSA end-to-end model, by using the post-inference algorithm during the inference stage, exploits alignment mapping information from both the current and previous time steps.

V. Performance Evaluation

In order to validate the effectiveness of the proposed interdecoder module and post-inference algorithm, we evaluate the ASR performance in terms of the word error rate (WER) on WSJ-80 hrs and Switchboard-300 hrs for various systems.

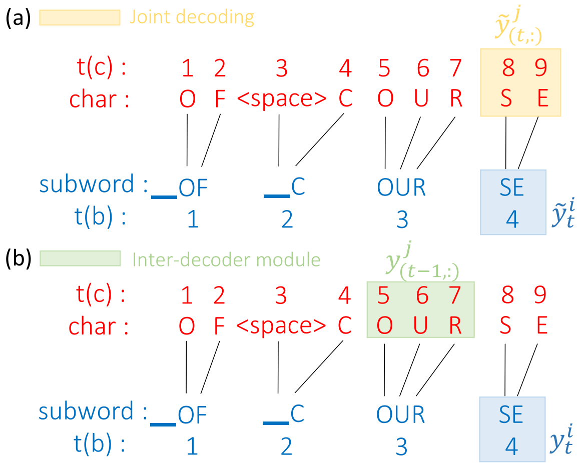


Fig. 7. The use of multi-granularity target information in (a) post-inference using $\tilde{y}^{j}_{(t,:)}$ or inference (the yellow font). (b) the interdecoder, utilizing only the past sub-sequence $y^{j}_{(t-1,:)}$ for modeling (green font).

The WSJ database contains 80 hours of transcribed speech. In this work, we follow the standard division, i.e., si284 for training, dev93 for validation and eval92 for evaluating the WER. The Switchboard corpus consists of a large amount of English language telephone speech. We choose the 300 h subset LDC97S62 for training, reserving 10% for cross validation. The Hub5 eval2000 (i.e., LDC2002S09) is chosen for performance evaluation, consisting of two subsets: 1) Switchboard (similar in style to the training set) and 2) CallHome, collected from conversations between friends and within families. The complete Hub5 eval2000, the subsets Switchboard and CallHome are denoted "Full," "SWD" and "CHE," respectively, For completeness, we also evaluate the ASR performance on the RT03 Switchboard test set (i.e., LDC2007S10).

The encoder used for both corpora has two convolutive layers, which down-sample the sequence in time, with 3×3 filters and 32 channels, followed by 6 layers of bi-directional long short-term memory (LSTM) with a cell size of 800. The default decoder is a two-layer uni-directional LSTM with 800 cells. We use 80-dimensional log-mel filterbank coefficients, three pitch coefficients and the normalized mean and variance as the input features. The character target in experiments is a set of 51 characters, which contain English letters, numbers, punctuation and special transcribed notations for WSJ, and 46 characters for Switchboard. For the subword target, we perform segmentation using SentencePiece,[2] which is based on the byte pair encoder algorithm. Based on [34] and the default setting in ESPnet, we use a vocabulary of size around 500 and 2000 for WSJ and Switchboard, respectively. ESPnet [35] and Pytorch [36] are used throughout experiments. As spec-augmentation and label-smoothing are included in ESPnet, they are applied in this work. For fair comparison, language model re-scoring, auxiliary output in the process of decoding or pre-training strategy is not applied.

TABLE I
WSJ Dataset WER When Considering Subword, Character, and/or Both as Labels

Model	Label Unit	dev93	eval92
Baseline	char	15.6	11.9
	subword	14.2	11.1
Baseline+	subword	14.1	10.9
MultiTask	both	13.8	10.6
MGSA$_{\mathrm{bi}}$	both	13.7	10.2
MGSA$_{\mathrm{uni}}$	both	**12.9**	**9.6**

During model learning, the CE is optimized using AdaDelta [37] with gradient clipping [38], where the hyper parameter λ is set to be 0.2. We also apply a uni-gram label smoothing technique [39] with a probability of $p = 0.05$ to avoid over-confident predictions. For the beam search algorithm, the beam size is set to be 20. We compare several variant systems, including:

- **Baseline**: following the commonly-used training criteria, which is the basis of all other extended systems.
- **Baseline+**: the baseline extended by cascading one additional bi-directional RNN layer for the encoder module to eliminate the effects of model size.
- **MultiTask**: using the multi-task learning strategy for multi-granularity modeling, which includes one shared encoder module and two separate decoder modules. Note that the MultiTask system ignores the interaction between target sequences.
- **MGSA$_{\mathbf{bi}}$**: uses the bi-interdecoder based on (10) and (11) to incorporate interaction between multi-granularity target units. This considers multi-granularity information for both character and subword predictions.
- **MGSA$_{\mathbf{uni}}$**: consists of a conventional encoder module and a uni-interdecoder module. It thus only incorporates multi-granularity information for subword prediction.

It is worth noting that in [32], by introducing multi-stage pre-training, speed perturbation, RNN-transducer the ASR performance is significantly improved, while it also requires more training time and makes the model more complicated. In general, the more strategies that are used, the more training time will be consumed and the more complicated the model. In order to focus on the impact of multi-granularity alignment mapping on the ASR performance, we therefore ignore similar potential strategies in the implementation of **MultiTask** and **Baseline** for fair comparison.

A. *Results and Discussions*

1) Evaluation of the Model Structure: In order to analyze the effect of the model structure on performance, we first consider the traditional beam search algorithm at the inference stage for all comparison methods. Table I lists WERs achieved on two validation sets. The WER for character and subword baselines in eval92 are 11.1% and 11.9%, respectively. Clearly, compared to Baseline, the MultiTask approach can improve the performance by 0.6%, and compared to Baseline+, a reduction in WER by

[2] [Online]. Available: https://github.com/google/sentencepiece

TABLE II
SWITCHBOARD-300 HRS DATASET WER WHEN CONSIDERING SUBWORD, CHARACTER, AND/OR BOTH AS LABELS

Model	Label Unit	Dev	eval2000			RT03
			SWB	CHE	Full	
Baseline	char	15.6	24.7	11.0	17.9	21.2
	subword	15.1	23.0	11.5	17.3	20.7
Baseline+	subword	15.9	23.0	12.4	17.7	21.0
MultiTask	both	14.4	21.9	11.3	16.8	19.4
$\mathrm{MGSA}_{\mathrm{bi}}$	both	14.5	21.3	11.0	16.2	19.2
$\mathrm{MGSA}_{\mathrm{uni}}$	both	**13.7**	**20.4**	**10.3**	**15.4**	**18.4**

0.3% is obtained. This implies that using multiple target information is more beneficial for improving the performance than considering more model parameters. Comparing $\mathrm{MGSA}_{\mathrm{uni}}$ and $\mathrm{MGSA}_{\mathrm{bi}}$ with MultiTask or the single-granularity baselines, we see that the alignment mapping in the decoder module is indeed helpful to improve the performance. As the performance of $\mathrm{MGSA}_{\mathrm{bi}}$ is worse than that of $\mathrm{MGSA}_{\mathrm{uni}}$ (e.g., 10.2% vs 9.6%), the bi-lateral transmission over multi-granularity units does not achieve a performance gain. Note that the major difference between these two models lies in the transition function (e.g., (11) for $\mathrm{MGSA}_{\mathrm{bi}}$ and (12) for $\mathrm{MGSA}_{\mathrm{uni}}$). The former has to further load the subword state for the prediction of character tokens, resulting in a more complicated structure. Due to the fact that the bi-lateral interaction increases the correlation between two sequences and reduces the fault tolerant ability during inference, the exposure bias problem becomes more serious as opposed to the uni-lateral counterpart or MTL. These lead to that $\mathrm{MGSA}_{\mathrm{bi}}$ cannot outperform $\mathrm{MGSA}_{\mathrm{uni}}$ in general.

Similarly, we next evaluate the ASR performance on the Switchboard-300 hrs corpus, which is much larger than WSJ. The results are shown in Table II. Clearly, the proposed $\mathrm{MGSA}_{\mathrm{uni}}$ reduces the WER by 1.4% and 1.9% compared to MultiTask and Baseline on the eval2000 dataset, respectively. For RT03, $\mathrm{MGSA}_{\mathrm{uni}}$ yields a reduction in WER of 1.0% and 1.7% compared to MultiTask and Baseline, respectively. In line with the WSJ results, $\mathrm{MGSA}_{\mathrm{bi}}$ cannot work better than $\mathrm{MGSA}_{\mathrm{uni}}$. We see that, for both WSJ and Switchboard corpora, $\mathrm{MGSA}_{\mathrm{uni}}$ outperforms its bi-lateral counterpart $\mathrm{MGSA}_{\mathrm{bi}}$. We will therefore only select the former for further comparisons. In fact $\mathrm{MGSA}_{\mathrm{uni}}$ has another advantage in that the prediction for all character sequences can be calculated simultaneously, and the parameters characters need to provide for the corresponding subword can be extracted all at once.

2) Evaluation of the Proposed Post-Inference: As the multi-granularity target not only affects the model structure, but also the inference, we therefore experimentally evaluate the impact of applying the proposed post-inference algorithm at the inference stage. For notational brevity, in the following the $\mathrm{MGSA}_{\mathrm{uni}}$ and MultiTask plus post-inference will be denoted by $\mathrm{MGSA}_{\mathrm{uni}}$+ and MultiTask+, respectively.

The performance on the WSJ dataset is shown in Table III. We see that, compared to $\mathrm{MGSA}_{\mathrm{uni}}$, $\mathrm{MGSA}_{\mathrm{uni}}$+ reduces the WER from 9.6% to 8.9% on eval92. $\mathrm{MGSA}_{\mathrm{uni}}$+ performs better than MultiTask+, so the proposed interdecoder is clearly beneficial.

TABLE III
THE WERS OF $\mathrm{MGSA}_{\mathrm{uni}}$ AND MultiTask WITH/WITHOUT POST-INFERENCE ON THE WSJ DATASET

Model	Label Unit	dev93	eval92
MultiTask	both	13.8	10.6
+ post-inference		13.4	10.1
$\mathrm{MGSA}_{\mathrm{uni}}$	both	12.9	9.6
+ post-inference		12.4	8.9

TABLE IV
THE WERS OF $\mathrm{MGSA}_{\mathrm{uni}}$ AND MultiTask WITH/WITHOUT POST-INFERENCE ON THE SWITCHBOARD DATASET

Model	Label Unit	Dev	eval2000			RT03
			SWB	CHE	Full	
MultiTask	both	14.4	21.9	11.3	16.8	19.4
+ post-inference		13.3	20.6	10.6	15.6	18.4
$\mathrm{MGSA}_{\mathrm{uni}}$	both	13.7	20.4	10.3	15.4	18.4
+ post-inference		12.8	19.3	10.1	14.7	17.6

Results on the Switchboard dataset are shown in Table IV. Similarly, the proposed $\mathrm{MGSA}_{\mathrm{uni}}$+ approach also further reduces the WER by 0.7% on eval2000 and by 0.8% on RT03.

As the application of the proposed post-inference is not restricted by the end-to-end structure, we therefore further show the performance of MultiTask+ on the WSJ dataset in Table III and on the Switchboard dataset in Table IV, respectively. Due to the use of the post-inference algorithm, the WER of MultiTask can be reduced by 0.5% on WSJ compared to the original MultiTask method, and the average reduction in WER turns out to be 1.2% on Switchboard. Hence, we conclude that the proposed post-inference is able to further improve the ASR performance. Notably, the improvement for $\mathrm{MGSA}_{\mathrm{uni}}$ is higher than for MultiTask. This is due to the fact that the alignment mapping information contained in the multiple granularities is taken into account in the former but not the latter. Since $\mathrm{MGSA}_{\mathrm{uni}}$+ achieves a performance gain with respect to $\mathrm{MGSA}_{\mathrm{uni}}$, which is sightly smaller than the improvement obtained by MultiTask+ over MultiTask, we can conclude that the performance gains obtained by separately using interdecoder and post-inference may be partly complementary.

B. visualization and Complexity Analysis

In this section, we will visualize the results of the comparison methods and compare the time complexity.

1) Visualization: The proposed $\mathrm{MGSA}_{\mathrm{uni}}$ method includes three attention modules: subword α_t^b, interaction α_t^i and character $\alpha_{t'}^c$. To analyze their functions, we visualize the alignment variables using the WSJ and Switchboard datasets in Fig. 8. The first three rows plot heatmaps of subword α_t^b, the interaction α_t^i and the character alignments $\alpha_{t'}^c$, with the plots in the left column being from WSJ and from Switchboard in the right column. The black dashed lines in Figs. 8(a) and (b) represent the estimated central position of the subword attention α_t^b, and similarly in Figs. 8(e) and (f). As with the representation of $y_{(t,:)}^c$, we convert the attention vector $\alpha_{t'}^c$ into $\alpha_{(t,:)}^c$, and plot the boundary position

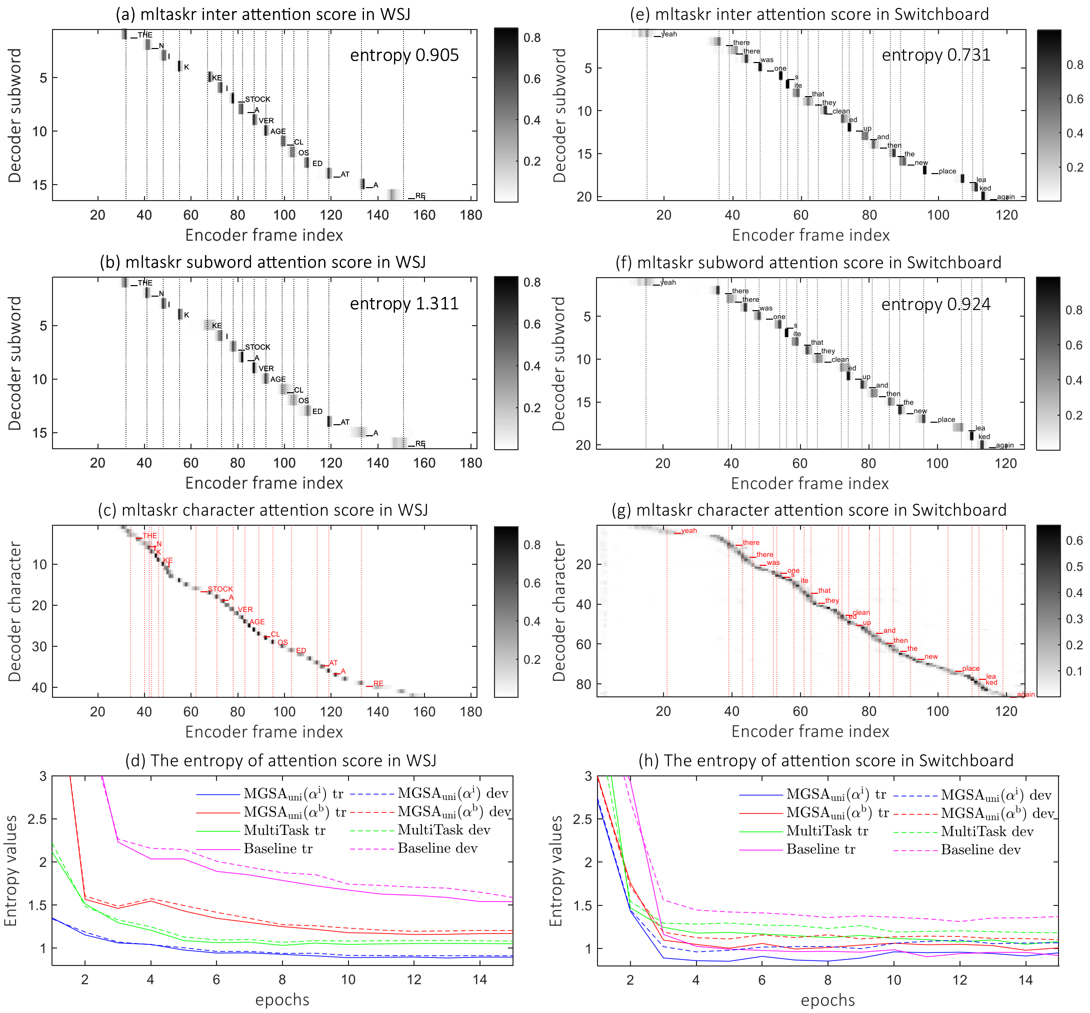


Fig. 8. Visualization and analysis of the attention for the AED-based approaches, where the subplots in the left column are obtained from the WSJ dataset and those on the right from Switchboard: (a) and (e) plot the alignment attention score, (b) and (f) plot the subword attention score, (c) and (g) plot the character attention score, and (d) and (h) plot the entropy of different attention weights across epochs. For simplicity, we use 'tr' and 'dev' to denote training and validation datasets, respectively.

using red dashed lines in Figs. 8(c) and (g). Comparing Fig. 8(b) to (c) (or Fig. 8(f) to (g)), it is obvious that the attention locations of the character $\alpha^c_{(t,:)}$ and the subword α^b_t are different. In Figs. 8(a) and (b) (or Figs. 8(e) and (f)), the central positions are also different. As the attention distributions of character and subword fluctuate around the ground truth boundary, the subword attention score and character attention score are not aligned in time domain[3]. Thus, it is reasonable to exploit more abundant intermediate representations from the encoder module for the prediction and fuse the subword and character vectors, say g^i_t, g^b_t and $g^c_{(t,:)}$, to complement the alignment. In Figs. 8(a) and (b), we can see that the entropy of the interaction alignment α^i_t is lower than that of the subword α^b_t (e.g., 0.905 vs 1.311). As a more comprehensive illustration, we plot the entropy in terms of epochs for the subword alignment scores of $\mathrm{Baseline}$, $\mathrm{MultiTask}$ and $\mathrm{MGSA_{uni}}$ in Fig. 8(d). We observe that the entropy of the interaction alignment score α^i_t is the lowest, which decreases rapidly and converges after several epochs. This is due to the fact that the interaction module contains history information from both subword and character. The more history information contained in α^i_t, the more accurate the prediction will be. Hence, we conclude that the combination of history subword and character is indeed helpful for optimizing the alignment vectors. Similar results can be seen for the Switchboard dataset in Fig. 8(h). Note that compared to WSJ, Switchboard entropies converge faster, due to the fact that it contains more training data, leading to more model updates possessed at each epoch.

[3] This misalignment might be caused by 1) the presented speech recognition systems, which are based on the attention mechanism, are not strictly aligned, and the alignment effect of the attention mechanism is different from real text boundaries; 2) the context modeling ability in the encoder module weakens the differences between adjacent frames, so a certain amount of effective information can also be obtained even if the alignment position is different from the ground truth; and 3) as the voicelessness or coarticulation is more obvious on character than on subword, the requirements of acoustic information for character classification and subword classification are different. Different requirements in acoustic information between character and subword may result in difference alignment distributions.

TABLE V
THE NORMALIZED TRAINING AND INFERENCE TIME CONSUMPTION OF THE COMPARISON METHODS ON THE WSJ AND SWITCHBOARD DATASETS WITH RESPECT TO THE BASELINE METHOD

Model	Training Time		Inference Time	
	WSJ	Switchboard	WSJ	Switchboard
CDS	2.5	3.6	2.5	3.6
MultiTask + post-inference	×1.44	×1.74	×1.00 ×1.80	×1.00 ×2.36
$\mathrm{MGSA_{uni}}$ + post-inference	×1.58	×1.97	×1.89 ×1.89	×2.45 ×2.45

2) Time Complexity: The normalized processing time for both training and inference stages of the $\mathrm{MGSA_{uni}}$ and MultiTask methods on the WSJ and Switchboard datasets with respect to Baseline are shown in Table V. It is clear that for both training and inference stages, both MultiTask and the proposed $\mathrm{MGSA_{uni}}$ method consume a longer time on both datasets compared to the Baseline method. Incorporating the post-inference algorithm for both MultiTask and $\mathrm{MGSA_{uni}}$ increases the time complexity. This is due to the fact that the post-inference requires an extra mapping transformation between multi-granularity units, resulting in more calculations for character sub-sequences. The decoding time of $\mathrm{MGSA_{uni}}$ and $\mathrm{MGSA_{uni}}$+ is the same, because the information required by the post-inference algorithm is already calculated by the interdecoder. In other words, the sequence alignment information and the corresponding character sub-sequence are the output of the char block of the interdecoder module, and the former can be directly applied to post-inference. The time consumption of the post-inference algorithm is thus negligible, effectively provided for free. In addition, the time complexity of $\mathrm{MGSA_{uni}}$+ is only slightly higher than that of MultiTask+. Therefore, we can conclude that the performance gain of the proposed $\mathrm{MGSA_{uni}}$+ method is improved at the cost of a small increase in time consumption. In Table V, we notice that for $\mathrm{MGSA_{uni}}$ and MultiTask+ methods, both the training time and inference time are different for the two datasets. This is caused by the segmentation fineness of subwords on the corpus. For analysis, we also give the average character density of subword (CDS) in the two corpora and use CDS to measure the segmentation fineness of subwords in Table V. It is obvious that the CDS differs significantly between the two datasets, and the time consumption is strongly dependent on the CDS.

C. Comparison to State-of-The-Art Systems

Finally, we compare the proposed MGSA method to state-of-the-art granularity-based end-to-end ASR systems. Note that in order to focus on the attention-based model without introducing complicated training strategies, e.g., CTC, RNN-tranducer, some results, such as [32], are excluded. Also, note that the provided results can be further improved as using a more complicated configuration in [32]. Table VI and Table VII show the performance and the granularity unit (e.g., character, subword, both) of different approaches using WSJ-80 hrs and Switchboard-300 hrs, respectively. Comparing with the optimal completion distillation (OCD) based Sabour method [40] which uses character units, or the Espresso baseline [41], we can conclude that the utilization of multi-granularity units and the proposed post-inference algorithm is more robust than optimizing exposure bias. For both datasets, the proposed $\mathrm{MGSA_{uni}}$+ method achieves the best performance. From both tables, it is obvious that multi-granularity based approaches (i.e., MultiTask, MultiTask+, $\mathrm{MGSA_{uni}}$ and $\mathrm{MGSA_{uni}}$+) outperform single-granularity based methods, implying that the utilization of multiple granularity information can improve the performance of end-to-end ASR systems. On the WSJ dataset, the character-based methods in general outperform subword-based approaches, while for Switchboard the latter work better. The choice of optimal single granularity for the design of ASR systems is thus dataset dependent.

TABLE VI
COMPARISON TO OTHER END-TO-END CE-BASED ASR SYSTEMS (WITHOUT LANGUAGE MODEL RE-SCORING) ON WSJ-80 HRS

Model	Label Unit	dev93	eval92
LS [39]	char	13.7	10.6
OCD [40]	char	-	9.3
PAPB [42]	char	-	10.6
EPAM [34]	char	14.0	10.6
Espresso [41]	char	14.8	12.1
LSD [43]	subword	-	9.6
PASM [44]	subword	18.5	15.6
Baseline	both	14.2	11.1
MultiTask	both	13.8	10.6
MultiTask+	both	13.4	10.1
$\mathrm{MGSA_{uni}}$+	both	**12.4**	**8.9**

TABLE VII
COMPARISON TO OTHER END-TO-END CE-BASED ASR SYSTEMS ON SWITCHBOARD-300 HRS

Model	Label Unit	eval2000			RT03
		SWB	CHE	Full	
EPAM [34]	char	10.1	22.5	16.3	-
LAS-IF-sMBR [45]	char	12.2	23.3	17.7	-
Espresso [41]	subword	10.7	20.7	15.7	-
Pre-training [46]	subword	11.9	23.7	17.7	-
SpecAugm(w/o) [47]	subword	11.2	21.6	16.4	-
Baseline	both	11.5	23.0	17.3	20.7
MultiTask	both	11.3	21.9	16.8	19.4
MultiTask+	both	10.6	20.6	15.6	18.4
$\mathrm{MGSA_{uni}}$+	both	**10.1**	**19.3**	**14.7**	**17.6**

VI. CONCLUSION

In this work, we proposed a multi-granularity sequence alignment approach for the AED-based ASR, which exploits the alignment mapping between different granularity units for both modeling and inference stages. By leveraging the dependency and interaction between multi-granularity target sequences, the interdecoder based framework can improve the ASR performance. The proposed post-inference algorithm can improve the performance significantly at the cost of a small increase in the time consumption. We found that the one-way interaction in the interdecoder module works better than the bi-lateral counterpart. In general, the utilization of more intermediate speech representations and sequence alignment mapping information is beneficial for ASR. As only two target units (e.g., character and subword) are taken into account in this work, we will consider more granularities in the future. We will also optimize the combination of multiple granularities and explore the application of the proposed MGSA method to other end-to-end ASR frameworks, e.g., transformer. In the future, we will consider the generalization capability of the proposed method using a larger-scale dataset (e.g., with thousands hours of training data) and the application to other languages, e.g., Chinese.

Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for their constructive suggestions that helped to improve the presentation of this article.

References

[1] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[3] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 193–199.

[4] O. Abdel-Hamid, L. Deng, D. Yu, and H. Jiang, "Deep segmental neural networks for speech recognition," in *Proc. Int. Speech. Community Assoc, Interspeech*, vol. 36, 2013, pp. 70–74.

[5] J. K. 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.

[6] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," in *Proc. Int. Speech Community Assoc., Interspeech*, 2018, pp. 791–795.

[7] C. Kim, M. Shin, A. Garg, and D. Gowda, "Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system," in *Proc. Int. Speech Community Assoc. Interspeech*, 2019, pp. 739–743.

[8] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-Attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 4835–4839.

[9] R. Masumura, T. Tanaka, T. Moriya, Y. Shinohara, T. Oba, and Y. Aono, "Large context end-to-end automatic speech recognition via extension of hierarchical recurrent encoder-decoder models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5661–5665.

[10] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2015, pp. 1468–1472.

[11] R. Sanabria and F. Metze, "Hierarchical multitask learning with CTC," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 485–490.

[12] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.

[13] D. Gowda, A. Garg, K. Kim, M. Kumar, and C. Kim, "Multi-task multi-resolution Char-to-BPE cross-attention decoder for end-to-end speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2019, pp. 2783–2787.

[14] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2017, pp. 3532–3536.

[15] Y. Kubo and M. Bacchiani, "Joint phoneme-grapheme model for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6119–6123.

[16] A. Garg, D. Gowda, A. Kumar, K. Kim, M. Kumar, and C. Kim, "Improved multi-stage training of online attention-based encoder-decoder models," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 70–77.

[17] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2017, pp. 959–963.

[18] T. Hori, S. Watanabe, and J. R. Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 287–293.

[19] T. Hori, J. Cho, and S. Watanabe, "End-to-end speech recognition with word-based RNN language models," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 389–396.

[20] W. Wang, Y. Zhou, C. Xiong, and R. Socher, "An investigation of phone-based subword units for end-to-end speech recognition," in *ISCA Interspeech*, 2020, pp. 1778–1782.

[21] D. B. Paul and J. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop Speech Nat. Lang.*, 1992, pp. 357–362.

[22] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1992, pp. 517–520.

[23] 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.

[24] W. Chan and I. Lane, "On online attention-based speech recognition and joint mandarin Character-Pinyin training," in *Proc. Int. Speech Community Assoc., Interspeech*, 2016, pp. 3404–3408.

[25] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 5149–5152.

[26] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2016, pp. 3707–3711.

[27] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4945–4949.

[28] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2018, pp. 7–11.

[29] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5060–5064.

[30] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4759–4763.

[31] S. Abdou and M. S. Scordilis, "Beam search pruning in speech recognition using a posterior probability-based confidence measure," *Speech Commun.*, vol. 42, no. 3/4, pp. 409–428, 2004.

[32] M. Huang, Y. Lu, L. Wang, Y. Qian, and K. Yu, "Exploring model units and training strategies for end-to-end speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 524–531.

[33] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 933–941.

[34] J. Tang, J. Hou, Y. Song, L. Dai, and I. Mcloughlin, "Effective exploitation of posterior information for attention-based speech recognition," *IEEE Access*, vol. 8, pp. 108988–108999, Jun. 2020.

[35] S. Watanabe *et al.*, "ESPNet: End-to-end speech processing toolkit," in *Proc. Int. Speech Community Assoc., Interspeech*, 2018, pp. 2207–2211.

[36] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[37] M. D. Zeiler, "Adadelta: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[38] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.

[39] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. Int. Speech Community Assoc., Interspeech*, 2017, pp. 523–527.

[40] S. Sabour, W. Chan, and M. Norouzi, "Optimal completion distillation for sequence learning," in *Int. Conf. Learn. Representations (ICLR)*, New Orleans, LA, USA, May 2019.

[41] Y. Wang *et al.*, "Espresso: A fast end-to-end neural speech recognition toolkit," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 136–143.

[42] M. K. Baskar, L. Burget, S. Watanabe, M. Karafiat, T. Hori, and J. H. Cernocky, "Promising accurate prefix boosting for sequence-to-sequence ASR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5646–5650.

[43] W. Chan, Y. Zhang, Q. Le, and N. Jaitly, "Latent sequence decompositions," 2016, *arXiv:1610.03035*.

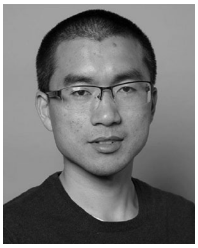
[44] H. Xu, S. Ding, and S. Watanabe, "Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 7110–7114.

[45] C. Weng *et al.*, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition." in *Proc. Int. Speech Community Assoc., Interspeech*, 2018, pp. 761–765.

[46] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A comprehensive analysis on attention models," in *Proc. IRASL Workshop, NeurIPS*, Montreal, Canada, Dec. 2018.

[47] D. S. Park *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Int. Speech Community Assoc., Interspeech*, 2019, pp. 2613–2617.

Jian Tang received the B.S. degree from the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China, in 2014. He is currently working toward the Ph.D. degree with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei, China. His current research interests include deep learning for speech recognition and acoustic modeling.



Jie Zhang (Member, IEEE) was born in Anhui Province, China, in 1990. He received the B.Sc. (Hons.) degree in electrical engineering from Yunnan University, Kunming, China, in 2012, the M.Sc. (Hons.) degree in electrical engineering from Peking University, Beijing, China, in 2015, and the Ph.D. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 2020. He is currently an Assistant Professor with the National Engineering Laboratory for Speech and Language Information Processing, Faculty of Information Science and Technology, University of Science and Technology of China, Hefei, China. His current research interests include multimicrophone speech enhancement, sound source localization, binaural auditory, speech recognition, and speech processing over wireless (acoustic) sensor networks. He was the recipient of the Best Student Paper Award for his publication at the 10th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2018) in Sheffield, U.K.

Yan Song received the B.Sc degree in electronic engineering from the University of Electronic Science and Technology of China, Hefei, China, in 1994, and the M.Sc. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 1997 and 2006, respectively. He is currently an Associate Professor with the National Engineering Laboratory for Speech and Language Information Processing, and has been a Faculty Member of the Department of Electronic Engineering and Information Science, University of Science and Technology of China since 2000. His research interests include multimedia information processing, automatic language identification, speaker diarization, and image classification.

Ian McLoughlin (Senior Member, IEEE) received the Ph.D. degree in electronic and electrical engineering from the University of Birmingham, Birmingham, U.K., in 1997. He worked for more than ten years in the R&D industry and about 15 years in academia, on three continents. He is currently a Professor with the Singapore Institute of Technology, Singapore. He has written many papers and several patents on speech analysis and communications, and is author of four books on speech processing and embedded computation. He is a fellow of the IET, a Chartered Engineer, was the recipient of the Chinese Academy of Sciences President's International Fellowship Award and Hundred Talent Program funding from Anhui, Province, China.

Li-Rong Dai (Member, IEEE) was born in China, in 1962. He received the B.S. degree in electrical engineering from Xidian University, Xi'an, China, in 1983, the M.S. degree from the Hefei University of Technology, Hefei, China, in 1986, and the Ph.D. degree in signal and information processing from the University of Science and Technology of China (USTC), Hefei, China, in 1997. In 1993, he joined USTC. He is currently a Professor with the School of Information Science and Technology, USTC. He has authored or coauthored more than 50 papers in his research field, which include speech synthesis, speaker and language recognition, speech recognition, digital signal processing, voice search technology, machine learning, and pattern recognition.