Hiformer: Sequence Modeling Networks with Hierarchical Attention Mechanisms

Xixin Wu, Hui Lu, Kun Li, Zhiyong Wu, Xunying Liu, Helen Meng, *Fellow, IEEE*

***Abstract*—The attention-based encoder-decoder structure, such as the Transformer, has achieved state-of-the-art performance on various sequence modeling tasks, *e.g.*, machine translation (MT) and automatic speech recognition (ASR), benefited from the superior capability of layer-wise self-attention mechanism in the encoder/decoder to access long-distance contextual information. Recently, analysis on the Transformer layers has shown that different levels of information, *e.g.*, phoneme level, word level and semantic level, are represented at different layers. Effectively integrating information from various levels is important for structured prediction. However, the self-attention in the conventional Transformer structure only focuses on intra-layer integration, and does not explicitly model inter-layer information relationships. Also, attention across the encoder and decoder (cross-coder) only focuses on the top encoder layer but ignores the intermediate layers. In this paper, we propose a sequence modeling structure equipped with a hierarchical attention mechanism, named Hiformer, that can consider the inter-layer and cross-coder hierarchical information to improve structured prediction performance. Extensive experiments conducted on both MT and ASR tasks demonstrate the effectiveness of the proposed Hiformer model.**

***Index Terms*—hierarchical attention mechanism, Transformer, automatic speech recognition, neural machine translation.**

I. Introduction

LEARNING structural information in sequential data plays an important role in many tasks, *e.g.*, machine translation (MT) and automatic speech recognition (ASR). Recently, attention-based encoder-decoder (AED) models have demonstrated significant successes in such sequence modeling tasks [1], [2]. In the AED models, the encoder encodes an input sequence to a hidden representation sequence, and the decoder predicts the outputs based on the encoded representation. An attention module across the encoder and decoder (cross-coder) is utilized to determine which parts of the representation sequence should be attended to and summarize the attended parts to a vector for each decoding step. Using the cross-coder attention, the decoder is enabled to flexibly access various parts of the encoded representation sequence.

The Transformer model is among the most promising AED structures [3]. In the Transformer structure, the self-attention mechanism in the encoder/decoder shows superior performance in modeling long-distance contextual information, compared to the conventional recurrent connections. Recent analysis on the Transformer structure has shown that different levels of information in input sequences, *e.g.*, phoneme level, word level and semantic level, are represented at different layers in the Transformer [4], [5]. The multi-level hierarchical information is important for structured prediction, *e.g.*, the prediction of a translated word needs to consider not only the corresponding source word, but also the related phrases and the semantic meaning of the sentence. Establishing interactions across different layers is desirable for utilizing such hierarchical information in the input sequences. Serban *et al.* [6] utilize hierarchical connections between layers of recurrent neural networks (RNNs) to enable the flowing of knowledge of textual hierarchical boundaries. However, the self-attention mechanism in the Transformer structure only focuses on the intra-layer information by retrieving the keys and values in the same layer and lacks explicit connections with previous layers. Also, the cross-coder attention only considers the encoded representation sequence at the top encoder layer and lacks connections with the lower intermediate layers, which also hinders the hierarchical information from flowing to the decoder.

In order to consider the hierarchical information, we propose a new AED structure improved from the Transformer model, named *Hiformer*, by introducing a novel hierarchical attention (hi-attention) mechanism. The hi-attention mechanism collects hierarchical information from previous layers by calculating the attention allocated to keys in previous layers using the queries in the current layer, and combining the corresponding values from previous layers according to the attention weights. The hi-attention is task-agnostic and can be used to improve the attention mechanisms for different tasks in various parts of the AED structure, *i.e.*, the encoder, the decoder and the cross-coder attention. We conduct thorough experiments on two representative sequence modeling tasks, MT and ASR to validate the effectiveness of the Hiformer model. The experimental results on both tasks consistently demonstrate the superiority of the Hiformer over the Transformer.

The rest of this paper is organized as follows: Section II reviews the previous research on sequence modeling with hierarchical connections. Recent related developments in MT and ASR areas are also introduced. The Transformer structure with the standard self-attention mechanism is introduced in Section III. Section IV describes the proposed hierarchical attention (hi-attention) mechanism. Section V illustrates the Hiformer structure with the introduced hi-attention. Experimental results on MT and ASR are described in Section VI. Analysis and conclusions are presented in Section VII and Section VIII.

Xixin Wu, Hui Lu, Kun Li, Xunying Liu and Helen Meng are with the Department of Systems Engineering and Engineering Mangement, The Chinese University of Hong Kong, Hong Kong SAR, China (e-mail: {wuxx,luhui,kunli,xyliu,hmmeng}@se.cuhk.edu.hk.

Zhiyong Wu is with Tsinghua Shenzhen International Graduate School of Tsinghua University, Shenzhen, China (e-mail: zywu@se.cuhk.edu.hk).

II. Related Work

A. Sequence Modeling with Neural Structures

Much effort has been devoted to designing effective neural structures for modeling sequential data. One research line is to capture temporal information in sequential data. Feedforward neural networks can only consider fixed time windows of input, even enhanced with sub-sampling as in time delay neural networks (TDNNs) [7], [8]. The introduction of recurrent connections effectively improves the temporal information capturing performance [9]. The long short-term memory (LSTM) RNNs demonstrate superior performance with the utilization of memory cells to store contextual information [10]. However, the sequential computation precludes parallelization within training examples and the direct access to long-distance context. The self-attention mechanism is introduced to address these problems by connecting different positions within a sequence to compute a representation for the current position [3]. Using the self-attention as a fundamental component, the Transformer structure has shown significant improvements over RNNs in various areas, *e.g.*, language modeling [11], MT [3] and ASR [12]. As recent studies suggest, different levels of information are learned at different layers in Transformer encoders and decoders [4], [5], [13]. In this work, we investigate the modeling of inter-layer hierarchical information by adding explicit connections between attention modules in the previous and the current layers to the original attention mechanism of the Transformer. The improvement is task-agnostic and can be integrated to the Transformer systems for various tasks, *e.g.*, MT and ASR, as demonstrated in this paper.

B. Hierarchical Connections for Sequence Modeling

Explicitly modeling sequential hierarchical information has been studied previously by enhancing inter-layer interactions, *e.g.*, residual connections [14], [15], highway connections [16], hierarchical connections [6] and cross-layer fusion [17], [18]. Our work is also related to hierarchical RNNs [19], which explicitly model information of different scales in a sequence by establishing connections between neighboring steps across different RNN layers to enable the flowing of knowledge of textual hierarchical boundaries. Our Hiformer structure is designed with the same aim to establish hierarchical connections. However, the Hiformer can access contextual information in longer distances with the attention mechanism.

Stacking multiple attention layers is another popular design choice for modeling complex contextual information in sequences [20]. Zhang *et al.* proposes to concatenate the context vectors from multiple attention layers for decoding [21]. Bertasius *et al.* adapts the Transformer to video data by staking self-attention layers for the time and the space dimensions, respectively [22]. Instead of simply stacking outputs of lower layers together, the FusionNet [18] uses low-level features from lower layers' outputs as part of keys and queries to compute attention for the question answering task. The proposed Hiformer structure enhances the inter-layer connections by retrieving the keys and values of attention modules in historical layers according to the queries in the current layer's attention module. Compared to FusionNet that focuses on specific question-context attention, the Hiformer structure improves encoder, decoder and cross-coder attention modules for sequence modeling tasks.

C. Neural Machine Translation

The Transformer [3] is a milestone in the area of neural machine translation (NMT) and has become the de facto benchmark structure. Attempts in improving Transformer-based MT systems has been made in various promising directions, *e.g.*, feature augmentation and structure optimization. Syntactic information [23] and pre-trained representation [24] are incorporated to provide translation models a syntactical or semantic prior. Back-translation is utilized to augment parallel data for NMT [25]. [26] and [2] equip basic models with translation memory components to cache training corpora. For structure optimization, more advanced NMT architectures based on the standard Transformer have been proposed. Transformer models with more stacked layers have shown superiority over the shallower ones [27]. However, training a deep Transformer is non-trivial and some strategies have been proposed, including training layers orderly from shallow to deep [27], parameters sharing among different layers [28] and creating residual connections between layers [15]. In the vanilla Transformer, the decoder only queries the representation of the topmost encoder layer through a cross-coder attention mechanism, which is considered insufficient for making use of source information from the lower encoder layers [29]. To this end, previous works propose deeper cross-coder attention mechanisms by, *e.g.*, aggregating the representations from multiple encoder layers for the decoder [17], [21], or using information from lower encoder/decoder layers (mean of the states in lower layers) to improve the attention allocation in the current layer [30], [31]. Zhang *et al.* introduce multiple parallel attention modules in the gated recurrent unit (GRU)-based decoder to attend to multiple encoder layers [20]. Wang *et al.* [32] propose to integrate tree structures of input text sequences into attention modules.

Our work shares the common objective of improving the Transformer structure. In contrast to the aforementioned works towards this objective, the proposed hi-attention can be applied to various parts of the AED structure. Moreover, the proposed attention mechanism emphasizes collection of hierarchical information from previous layers by dynamically allocating attention to previous layers, while the previous methods only focus on the attention allocation in the current layer by enhancing attention module inputs with deterministic connections from previous layers, *e.g.*, residual connections. Also, the collection reuses key and value vectors from previous layers without additional parameters, while the previous approaches require extra parameters to compute the key and value vectors for inter-layer attention.

D. Automatic Speech Recognition

The Transformer structure has been successfully applied to the ASR task and achieved outstanding performance [12], [33]. Though RNNs are more suitable for streaming ASR, the self-attention mechanism has the advantage of integrating information from longer-distance context, leading to superior performance than recurrent connections in RNNs. Many adaptations have been made to improve the Transformer structure for streaming, *e.g.*, restricting the attention computation to a fix-sized context window [12], [34]. [33] investigates the combination of self-attention and RNN transducers and shows that limiting the left attention context can make decoding computationally tractable for streamable speech recognition. Wang *et al.* [12] explore more encoding methods and show that 2D convolutional embeddings can implicitly model the positional information specifically for ASR. While Transformer blocks are good at capturing content-based global interactions, convolutional layers are better at exploiting local features. The convolution-augmented Transformer (Conformer) is proposed to combine the merits of both sides to model both local and global dependencies within an audio sequence, and has achieved state-of-the-art performance on several corpora [1].

This work improves the self-attention in the Transformer architecture to the hi-attention for integration of hierarchical information from multiple historical layers. The hi-attention inherits the key-value pairs from the self-attention in historical layers, therefore the improvement on the original Transformer to enhance the self-attention mechanism for streaming can be directly applied to the Hiformer model, *e.g.*, limiting attention windows.

III. Transformer with Self-Attention Mechanism

The general sequence modeling learns the mapping from a source sequence to another target sequence. Most competitive sequence modeling systems adopt the AED architecture [20], *e.g.*, the Transformer [1], [3]. In the AED architecture, the input sequence is first encoded into a hidden representation sequence, upon which the output sequence is decoded, in an autoregressive or non-autoregressive manner. Attention weights between the encoder and the decoder are utilized to combine the encoded representation sequence into a representation vector for each decoding step. We will illustrate these components in the Transformer structure in the following sections.

A. Encoder & Decoder

The encoder is composed of N identical layers, or called blocks, which are stacked one by one. Each layer has two sub-layers, a multi-head self-attention module and a position-wise feedforward network. The self-attention module, as introduced in the next section, attends to intra-layer context and aggregate information for each time step. The two sub-layers are surrounded by residual connections and followed by layer normalization [35].

The decoder also consists of N layers that contains three sub-layers, a multi-head self-attention module, a cross-coder multi-head attention module and a feedforward network. Similar to the encoder sub-layers, residual connections and layer normalization are employed for each sub-layer. The self-attention module focuses on intra-layer context weighting and aggregation. The cross-coder attention module determines which steps of encoder outputs to be focused on at the current decoding step. To prevent the decoder from attending to future positions, proper masks are applied to the decoder inputs. The decoder outputs are also offset by one position, such that the predictions for certain position only depend on the outputs at previous positions.

B. Self-attention

In the standard Transformer [3], the scaled dot-product self-attention is adopted to model the correlation between each step pairs in the same layer, as shown in Figure 1(a). The attention function in the l-th Transformer layer is computed based on a set of queries $\boldsymbol{Q}^{(l)}$ performed on the keys $\boldsymbol{K}^{(l)}$ and the corresponding values $\boldsymbol{V}^{(l)}$:

$$\boldsymbol{S}^{(l)} = \texttt{softmax}\big(d_k^{-1/2}\boldsymbol{Q}^{(l)}\boldsymbol{K}^{(l)^\top}\big)\boldsymbol{V}^{(l)}, \tag{1}$$

where $\boldsymbol{Q}^{(l)} \in \mathbb{R}^{T\times d_k}, \boldsymbol{K}^{(l)} \in \mathbb{R}^{T\times d_k}$ and $\boldsymbol{V}^{(l)} \in \mathbb{R}^{T\times d_v}$ are queries, keys and values derived from the input sequence. T, d_k and d_v are the sequence length and the hidden embedding dimensions of keys and values, respectively.

Note that the attention outputs $\boldsymbol{S}^{(l)}$ for different layers are calculated separately in Eq. (1), hence only the intra-layer information is integrated, and inter-layer connections are not established for integration of information from various layers.

Multi-head attention is shown to improve the self-attention performance by adopting parallel projections to obtain the queries, keys and values and obtain attention outputs separately as Eq. (1) [3]. The outputs of multiple heads are then concatenated and projected back to a shared space again. The multi-head self-attention outputs $\boldsymbol{S}^{(l)}_{\texttt{mh}}$ can be calculated with the single-head outputs $\boldsymbol{S}^{(l)}_{*}$ from Eq. (1) as:

$$\boldsymbol{S}^{(l)}_{\texttt{mh}} = \texttt{concat}(\boldsymbol{S}^{(l)}_1, ..., \boldsymbol{S}^{(l)}_m)\boldsymbol{W}^S, \tag{2}$$

where $\boldsymbol{S}^{(l)}_j$ is the j-th head attention outputs and $\boldsymbol{W}^S \in \mathbb{R}^{md_v\times d}$ is a trainable matrix. m and d are the head number and the model hidden dimension, respectively.

C. Positional Encoding

To enable the Transformer blocks to consider position information, positional encodings are added to the input of the encoder and decoder stacks. There are different choices of such positional encodings. One option is each dimension of the encodings is based on sine and cosine functions [3]:

$$\texttt{PE}(p, 2j) = \sin(p/10000^{2j/d}), \tag{3}$$

$$\texttt{PE}(p, 2j+1) = \cos(p/10000^{2j/d}), \tag{4}$$

where p is the position and j is the dimension. With these functions, $\texttt{PE}(p+k,*)$ can be represented as a linear function of $\texttt{PE}(p,*)$, so that relative positions can be learnt by encoders and decoders.

IV. Hierarchical Attention Mechanism

To enable the inter-layer hierarchical information integration, we propose a hierarchical attention, called *hi-attention*, by introducing connections between the current layer and historical layers, *i.e.*, lower layers. We will explain the improvement of hi-attention over self-attention in the following sections.

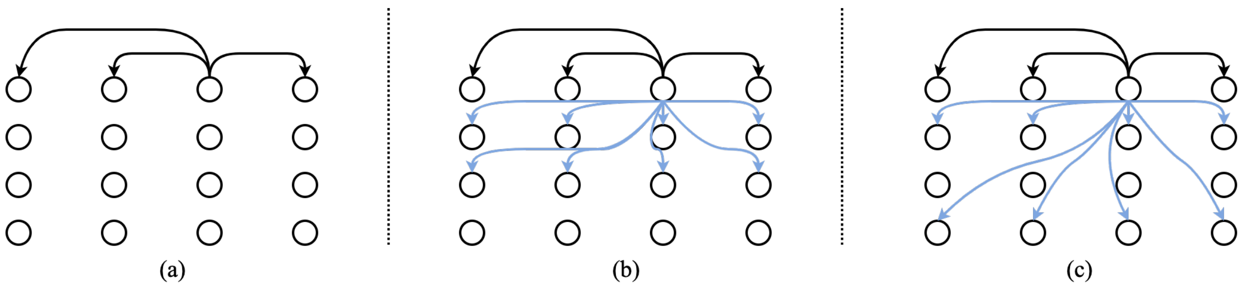

Fig. 1. Comparison of self-attention and hi-attention. (a) Self-attention; (b) Hi-attention that considers two historical layers; (c) Hi-attention that considers two historical layers with a dilation factor of 2.

A. Hi-attention

Inspired by the observation [4], [5] that different Transformer layers in pre-trained models, like wav2vec [36], represent different levels of information, we propose to consider the multi-layer/level information when calculating the attention outputs, as shown in Figure 1(b). Given the keys and values calculated in the historical layers $\boldsymbol{K}^{(i)}, \boldsymbol{V}^{(i)}, i = 1, 2, ..., l-1$, the cross-layer hi-attention can be calculated as:

$$\begin{aligned} \boldsymbol{H}^{(l,i)} =& \texttt{softmax}\big(d_k^{-1/2}\boldsymbol{Q}^{(l)}\boldsymbol{K}^{(i)}{}^{\top}\big)\boldsymbol{V}^{(i)}, && (5)\\ \boldsymbol{H}^{(l)} =& \boldsymbol{S}^{(l)} + \texttt{concat}(\boldsymbol{H}^{(l,l-1)}, ..., \boldsymbol{H}^{(l,l-n)})\boldsymbol{W}^{H}, && (6) \end{aligned}$$

where $\boldsymbol{W}^H \in \mathbb{R}^{nd_v \times d_v}$ is a trainable matrix and n is a hyperparameter of number of historical layers considered in the hi-attention. `concat` is a function to concatenate the input matrices along the last dimension. When $n = 0$, the hi-attention backs off to the original self-attention, which does not consider historical layer information. Note that since the key-value pairs $\boldsymbol{K}^{(i)}$ and $\boldsymbol{V}^{(i)}$ are directly borrowed from the historical layers, the hi-attention structure only requires additional parameters $\boldsymbol{W}^H$ compared with the self-attention structure. Compared to previous works using another set of attention modules [30], our hi-attention has better parameter efficiency by reusing the keys and values in historical layers.

1) Dilation: To enable the hi-attention structure to consider more distant layer information, while at the same time restrain the increase of parameter size, we introduce the dilated connections to the hi-attention structure. Using a dilation factor of f, the indices of the considered n historical layers at the l-th layer are $\{l-1, l-f-1, ..., l-(n-1)f-1\}$, as illustrated in Figure 1(c). The concatenation in Eq. (6) can be improved to

$$\boldsymbol{H}^{(l)} = \boldsymbol{S}^{(l)} + \texttt{concat}(\boldsymbol{H}^{(l,l-1)}, ..., \boldsymbol{H}^{(l,l-(n-1)f-1)})\boldsymbol{W}^{H}. \quad (7)$$

With the dilated connections, the hi-attention can consider a wider window of historical layers.

2) Multi-head Attention: Similarly, the hi-attention can be improved to have multiple heads:

$$\begin{aligned} \boldsymbol{H}^{(l)}_{\texttt{mh}} =& \boldsymbol{S}^{(l)}_{\texttt{mh}} + \texttt{concat}(\boldsymbol{H}^{(l,l-1)}_{\texttt{mh}}, ..., \boldsymbol{H}^{(l,l-n)}_{\texttt{mh}})\hat{\boldsymbol{W}}^{H}, && (8)\\ \boldsymbol{H}^{(l,i)}_{\texttt{mh}} =& \texttt{concat}(\boldsymbol{H}^{(l,i)}_{1}, ..., \boldsymbol{H}^{(l,i)}_{m})\boldsymbol{W}^{C}, \end{aligned}$$

where $\boldsymbol{H}^{(l,i)}_k$ is the k-th head outputs between the l-th and the i-th layer, as in Eq. (5). $\boldsymbol{W}^C \in \mathbb{R}^{md_v \times d}$ is a trainable matrix to project the concatenated multi-head outputs to a shared hidden space, and $\hat{\boldsymbol{W}}^H \in \mathbb{R}^{nd \times d}$ is another trainable matrix to project the concatenation of multi-head outputs across layers back to the model's hidden space. In practice, we can merge the two projections of $\boldsymbol{W}^C$ and $\hat{\boldsymbol{W}}^H$ into one single projection with a trainable matrix of $\boldsymbol{W}^F \in \mathbb{R}^{nmd_v \times d}$:

$$\begin{aligned} \boldsymbol{H}^{(l)}_{\texttt{mh}} =& \boldsymbol{S}^{(l)}_{\texttt{mh}} + \texttt{concat}(\boldsymbol{H}^{(l,l-1)}_{1:m}, ..., \boldsymbol{H}^{(l,l-n)}_{1:m})\boldsymbol{W}^{F}, && (9)\\ \boldsymbol{H}^{(l,i)}_{1:m} =& \texttt{concat}(\boldsymbol{H}^{(l,i)}_{1}, ..., \boldsymbol{H}^{(l,i)}_{m}). \end{aligned}$$

The parameter size of hi-attention increases with only nd^2 for each layer that introduces the hi-attention structure.

We also explore another option to combine hi-attention and self-attention, such that the model parameter size is not increased. Eq. (9) can be changed to simply sum up all the intra-layer attention outputs $\boldsymbol{S}^{(l)}_m$ and the inter-layer attention outputs $\boldsymbol{H}^{(l,i)}_m$ from historical layers:

$$\begin{aligned} \bar{\boldsymbol{H}}^{(l)}_{\texttt{mh}} = \texttt{concat}(&\boldsymbol{S}^{(l)}_1 + \sum_{i=l-1}^{l-n}\boldsymbol{H}^{(l,i)}_1, ...,\\ &\boldsymbol{S}^{(l)}_m + \sum_{i=l-1}^{l-n}\boldsymbol{H}^{(l,i)}_m)\boldsymbol{W}^{S}. \quad (10) \end{aligned}$$

Though directly summing attention outputs as Eq. (10) does not require extra parameters, analysis in Section VII shows that concatenating the outputs as Eq. (9) outperforms summing the outputs. Hence, in our experiments, we use the concatenating option.

B. Positional Encoding & Masking

The hi-attention structure directly reuses the queries, keys and values of the self-attention module, hence the positional encodings [3], [37] applied to queries, keys and values are inherited from the self-attention. Also, the masks applied to the key-value pairs for different layers can be reused conveniently. The hierarchical dilation can also be implemented with masks.

V. Hiformer Architecture

Compared with the Transformer, the Hiformer structure improves the self-attention to the hi-attention in the encoder, decoder and cross-coder attention modules, as shown in Figure 2. In the following, we will introduce these three parts of the Hiformer structure.

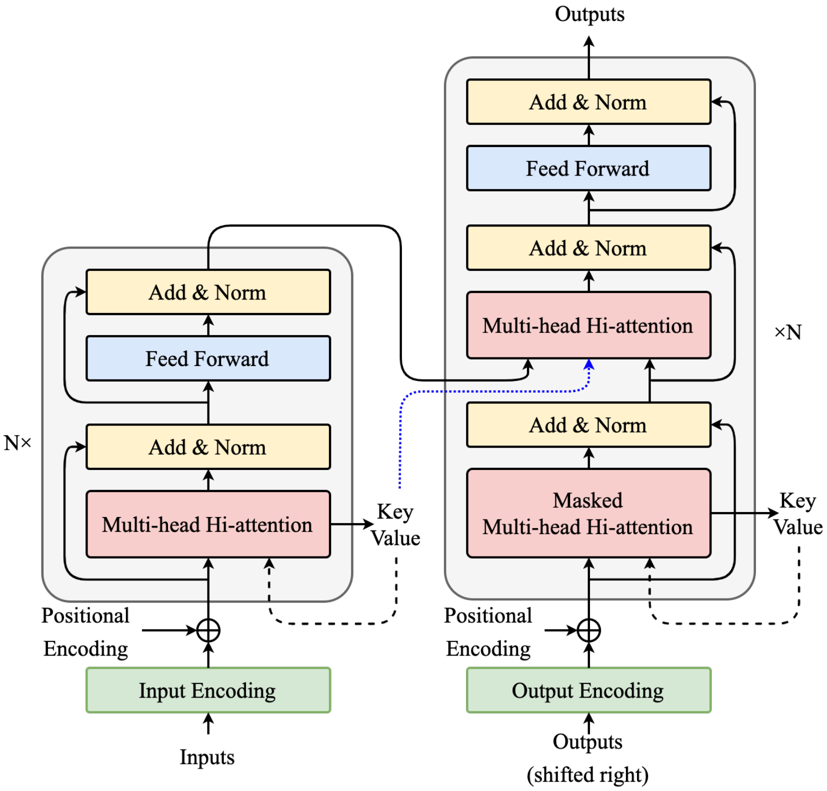

Fig. 2. Hiformer architecture. Compared to traditional Transformer architecture, the hi-attention can access to information in historical layers (dashed arrows), and propagate multi-layer hierarchical information across the encoder and the decoder (blue arrow).

TABLE I
STATISTICS OF THE JRC-ACQUIS EXPERIMENTAL DATA.

Directions	#Train Pairs	#Dev Pairs	#Test Pairs
Es⇔En	679,088	2,533	2,596
De⇔En	699,569	2,454	2,483

A. Encoder

The encoder consists of N layers equipped with the hi-attention. In each of these layers, we follow the previous standard configuration of Transformer [3]. Each layer is composed of a multi-head hi-attention sub-layer and a feedforward sub-layer. Residual connections with layer normalization are employed around the two sub-layers.

B. Decoder

Similar to the encoder, N layers with hi-attention are stacked in the decoder. Each decoder layer consists of an intra-layer multi-head hi-attention, a cross-coder multi-head hi-attention and a fully connected feedforward sub-layer. These three sub-layers are surrounded by residual connections followed by layer normalization. The intra-layer attention is the same as that in the encoder, and the inter-layer attention matches the queries from the decoder and the keys from the encoder layers and aggregating the corresponding values according to the query-key matching. Attention masks are applied to the hi-attention to prevent dependence on future outputs.

C. Attention

The hi-attention mechanism in the Hiformer structure is employed to handle three types of information integration:

- Intra-layer integration–similar to the conventional self-attention, the hi-attention projects hidden representations to queries, keys and values and integrates the values in the same layer by matching the corresponding queries and keys.
- Inter-layer integration–based on the keys and values generated by historical layers, the hi-attention in the current layer integrates the historical-layer values by matching the current-layer queries with the corresponding historical-layer keys.
- Cross-coder integration–in the conventional Transformer, only the values from the topmost encoder layer are integrated. In the proposed Hiformer structure, values generated from not only the top encoder layer but also the lower encoder layers are integrated in the cross-coder attention. For the lower layers, the key-value pairs generated for the intra-layer attention are reused.

VI. Experiments

We evaluate the Hiformer structure on two sequence modeling tasks, MT and ASR, that tackle text and audio input sequences respectively. Both tasks predict discrete labels for the target sequences, which generally have different lengths from the input sequences.

TABLE II
BLEU scores on the JRC-Acquis corpus. "our impl." denotes our implementation of the Transformer model on this corpus. "concat" and "sum" denote the two combination option in the hi-attention modules. *denotes Hiformer outperforms our implemented Transformer significantly with $p < 0.05$, tested by bootstrap re-sampling [38].

#	Systems	Es⇒En		En⇒Es		De⇒En		En⇒De	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
1	RNNencdec [39]	63.97	64.30	61.50	61.56	60.10	60.26	55.54	55.14
2	Transformer [2]	64.25	64.07	62.27	61.54	59.82	60.76	55.01	54.90
3	Transformer (our impl.)	64.41	64.18	62.39	61.63	59.92	60.36	54.83	55.08
4	Hiformer (concat Eq. (9))	**65.11**	**64.91***	**62.71**	**62.07***	**60.91**	**61.52***	55.48	**56.17***
5	- Cross-coder Hi-attn	65.07	64.33	62.66	61.77	60.63	61.36	55.51	55.58
6	- Dec Hi-attn	65.08	64.75	62.55	61.86	60.54	61.40	55.30	55.60
7	Hiformer (sum Eq. (10))	64.73	64.47	62.34	62.01	60.38	61.02	55.37	55.42

TABLE III
Evaluation of translation performance on the WMT'14 English⇒German ("En⇒De") translation task. #Para. denotes number of parameters, "Train" and "Decode" respectively denote the training speed (steps/second) and decoding speed (sentences/second) on a Tesla V100 GPU.

Systems		#Para.	Train	Decode	FLOPs(10^{18})	BLEU
Transformer		66.48M	2.34	235.76	3.7	27.42
Transformer-7L		73.84M	1.62	212.39	5.7	27.64
Transformer-1.5W		132.75M	1.67	170.85	7.3	27.63
Transformer-7L-1.5W		149.29M	1.47	156.53	8.6	27.89
Hiformer	Concat Eq. (9)	66.70M	0.47	138.27	8.8	**28.24**
- Cross-coder Hi-attn		66.63M	0.52	185.06	7.6	27.99
- Dec Hi-attn		66.56M	0.58	229.76	6.9	27.67
Hiformer	Sum Eq. (10)	66.48M	0.48	140.14	8.3	**28.07**
- Cross-coder Hi-attn		66.48M	0.53	191.87	7.5	27.85
- Dec Hi-attn		66.48M	0.59	228.43	6.6	27.63

TABLE IV
Evaluation of various systems on the WMT'14 En⇒De task. #Para. denotes number of parameters.

Systems	#Para.	BLEU
Transformer [3]	65M	27.31
MSC [40]	73M	27.68
DLCL [41]	62M	27.60
Deep Representation [30]	111M	28.78
GTRANS [17]	225M	30.01
Hiformer	67M	28.24

TABLE V
WER (%) of various systems on the AMI IHM dataset. *and**indicate Chiformer significantly outperforms our implmented Conformer with $p < 0.05$ and $p < 0.005$ respectively. †denotes systems from ESPnet official repository.

Systems	no LM		with LM	
	Dev	Eval	Dev	Eval
Hybrid [46]	–	–	–	17.5
Transformer†	19.8	19.1	19.1	18.3
Conformer†	18.0	17.0	17.7	16.5
Conformer (our impl.)	18.2	17.1	18.1	16.9
Chiformer	18.0*	16.7**	17.8**	16.5**

A. Machine Translation

We conduct the MT experiments on the JRC-Acquis corpus [42], which contains the total body of European Union (EU) law applicable to the EU member states, and the standard WMT 2014 English-to-German dataset. On the JRC-Acquis corpus, we focus on the translation directions of Spanish⇒English (Es⇒En), En⇒Es, German⇔English (De⇒En) and En⇒De. The corresponding statistics is shown in Table I. We use the same dataset that is processed by [43] and followed by [2], hence our experimental results can be fairly compared with the results in [2], [43]. The WMT 2014 English-German training set consists of about 4.5M sentence pairs. We use the newstest2014 as test set. The sentences are tokenized by Moses [44] and byte pair encoding (BPE) [45] with a shared vocabulary of 44k symbols.

1) MT Models: The Transformer structure has the same configuration as the Transformer Base in [3], with 8 attention heads, 512 dimensional hidden states and 2048 dimensional feedforward states. The encoder and decoder both contain 6 Transformer blocks. We use the baseline Transformer implemented on the JRC-Acquis by [2][1] and the Transformer baseline from Fairseq[2] on the WMT'14 dataset. Based on these Transformer baselines, we build the Hiformer models by replacing self-attention modules with hi-attention modules. To ensure fair comparison, the model configuration and training settings of the Transformer and the Hiformer are kept the same, except the introduction of hi-attention in the Hiformer models. We also compare the Hiformer with the well-known attention-based encoder-decoder MT model based on recurrent neural networks, denoted as RNNencdec [39]. We follow the learning rate, dropout and label smoothing settings in [2], [3]. The Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$ is used to train all models up to 150K training steps. The models are trained on one Tesla V100 GPU. The max tokens are set to 16384 with an update frequency of 8. The dropout rate of Hiformer is increased to 0.2 from 0.1 used in Transformer training. A beam size of 5 is used in decoding.

[1] https://github.com/jcyk/copyisallyouneed
[2] https://ai.facebook.com/tools/fairseq

TABLE VI
WER (%) of various systems on the Librispeech corpus. *and** indicate Chiformer significantly outperforms Conformer with p <0.05 and p <0.005, respectively. †denotes systems from ESPnet official repository.

Systems	no LM				with LM			
	Dev		Test		Dev		Test	
	Clean	Other	Clean	Other	Clean	Other	Clean	Other
Transformer [47]	2.54	6.67	2.89	6.98	2.10	4.79	2.33	5.17
Conformer [1]	–	–	2.1	4.3	–	–	1.9	3.9
Conformer†	2.1	5.2	2.4	5.2	1.8	3.9	2.0	4.2
Conformer (our impl.)	2.4	6.7	2.8	6.5	1.9	4.9	2.1	4.9
Chiformer	2.4	**6.5**	**2.6****	**6.4**	1.9	**4.7***	2.2	**4.8**
- Cross-coder Hi-attn	2.5	6.6	2.7	6.5	1.9	4.8	2.1	4.9
- Dec Hi-attn	2.5	6.7	2.7	6.6	1.9	4.8	2.1	5.0

TABLE VII
WER (%) of the Conformer and Chiformer systems on the AISHELL-2 corpus. †denotes systems from ESPnet official repository.

Systems	no LM				with LM			
	dev_ios	test_android	test_ios	test_mic	dev_ios	test_android	test_ios	test_mic
Transformer†	–	–	–	–	8.9	7.5	8.6	8.3
Conformer†	5.4	6.1	5.7	6.1	5.2	6.0	5.5	5.8
Conformer (our impl.)	5.6	6.4	5.8	6.4	5.4	6.1	5.5	6.0
Chiformer	**4.9**	6.1	**5.2**	**5.9**	**4.8**	6.0	**5.1**	5.8

2) Results: The BLEU scores of the compared systems on the JRC-Acquis corpus are shown in Table II. It can be found that the Hiformer significantly outperforms the other baseline systems on all the four translation directions, with the significance tested by bootstrap re-sampling [38]. This demonstrates the superiority of the proposed hi-attention mechanism over the self-attention. We investigate the effect of the hi-attention in various parts of the Hiformer structure. It can be found that the cross-coder hi-attention plays an important role in the Hiformer model (by comparing line 4 and line 5 in Table II). Reverting the cross-coder hi-attention to self-attention degrades the performance in all four translation directions consistently, with a BLEU score reduction of 0.16–0.59 on the test sets. It can also be found that simply using hi-attention in the encoder can still provide performance gains (comparing line 3 and 6).

The experimental results on the WMT'14 English⇒German translation task also demonstrate the effectivenss of the Hiformer structure, as shown in Table III. The Hiformer structure, with either concatentating or summing combination in hi-attention (Eq. (9) or Eq. (10)), outperforms the Transformer baseline. Note that the Hiformer with summing combination does not require additional parameters in comparison to the Transformer. We also build three variants of the Transformer base structure, *i.e.*, Transformer-7L, Transformer-1.5W and Transformer-7L-1.5W. The Transformer-7L structure consists of 7 encoder layers and 7 decoder layers. The Transformer-1.5W has the same layer numbers as the Transformer base, but with 1.5 times wider layers. The Transformer-7L-1.5W combines these two improvements. All these three systems achieve better performance than the Transformer base model. The Hiformer outperforms all these three Transformer variants with less model parameters, which indicates the parameter efficiency of the Hiformer structure. We estimate the number of floating point operations used to train a model by multiplying the training time, number of used GPUs and an estimate of the sustained single-precision floating-point capacity of the used GPUs. We use the value of 14 TFLOPs for the Tesla V100 GPU. The Hiformer structure requires around 2.5 times the training cost of the Transformer structure, but achieves significantly better performance, improving from 27.42 to 28.24 BLEU. The Transformer-7L-1.5W requires comparable FLOPs with the Hiformer but has more parameters. However, the performance of Transformer-7L-1.5W is inferior to that of Hiformer with either concatenating or summing option, which indicates the training efficiency of the Hiformer structure. Table IV shows the comparison of the Hiformer with the latest systems that consider cross-layer information. The Hiformer outperforms the MultiScale Collaborative (MSC) nets [40] and the Transformer with Dynamic Linear Combination of Layers (DLCL) [41]. The Deep Representation [30] and GTRANS [17] achieve better performance but require significantly more model parameters.

B. Automatic Speech Recognition

We conduct experiments on the benchmark ASR corpora of AMI meeting transcription [48], Librispeech [49] and AISHELL-2 [50]. The AMI dataset comprises approximately 100 hours of meeting recordings with 3-5 speakers per meeting recorded by independent headset microphones (IHM). About 81 hours of the data are used as training set and around 9 hours for development (Dev) and evaluation (Eval) set [46]. The Librispeech corpus contains read speech data of audiobooks by multiple speakers, and has been carefully segmented and aligned. About 960 hours of the data are used as training set, and 20 hours for development and testing. The Dev and Test sets both comprise two subsets, {Dev,Test}-clean and {Dev,Test}-other. The 'other' sets are more acoustic challenging than the 'clean' sets. The training data of AISHELL-2 contains 1,000 hours of Mandarin speech (around 1 million utterances), and the dev and test sets contains 2,500 and

5,000 utterances recorded via three parallel channels, *i.e.*, iOS, Android and Microphone.

1) ASR Models: The state-of-the-art sequence-to-sequence architecture for ASR is the convolution-augmented Transformer (Conformer) [1], where convolutional layers are introduced to enhance the modeling of local feature patterns. We improve the Conformer model to a convolution-augmented Hiformer model, named *Chiformer*. Experimental comparison is conducted between the following acoustic models:

- **Transformer** [12], [47]. For the AMI corpus, we build a Transformer-based model, where the encoder and decoder are composed of 12 and 6 layers with 2048 dimensions respectively. The self-attention modules have 4 heads of 256 dimensions, *i.e.*, the attention dimension is 1024. For Librispeech, we directly include the reported results by [47] in Table VI. The Transformer in [47] has 24 layers for the encoder and 6 layers for decoder, with 4096 dimensions for each layer and 4 heads for self-attention modules.
- **Conformer** [1]. Compared with the Transformer, the Conformer encoder improves the feed forward layers to a feedforward module. In the feedforward module, layer normalization is applied first, and then two linear layers are utilized to expand the hidden representation and transform the representation back to the original dimension. The Swish activation is used to regularize the network. Between the multi-head attention and the top feedforward module, a convolution module with point-wise convolution and a gated linear unit is added, followed by a depth-wise convolutional layer with batch normalization and another point-wise convolutional layer. For the AMI and Librispeech corpora, we build two Conformer models with the same configuration, with 12 Conformer blocks for the encoder and 6 Transformer blocks for the decoder. The block dimension is 2048, and the self-attention modules have 4 heads. We implement the Transformer and Conformer models based on ESPnet[3].
- **Chiformer**, We introduce the hi-attention mechanism to the baseline Conformer to build three Chiformer models for the three corpora, respectively. The configurations of the three Chiformer models are the same as the Conformer models except for the introduction of hi-attention modules and the increase of attention dropout rate from 0 to 0.1. This makes it possible for fair comparison. The hi-attention considers two historical layers with a dilation factor of 1 in the experiments. The Chiformer has a parameter size of 126.64M compared to the baseline Conformer of 116.15M, which is only an increase of ∼10% in the total size.

Three language models (LMs) are trained for the AMI, Librispeech and AISHELL-2 systems respectively. All LMs are composed of 16 Transformer layers of 2048 dimensions with 8 attention heads. The reference transcripts of AMI training data, the standard Librispeech LM corpus, and the training set of AISHELL-2 are used for LM training, respectively. We adopt a joint CTC/attention-based encoder-decoder structure for the two corpora. The training weight for the CTC branch is 0.3.

2) Results: The AMI experimental results are shown in Table V. The proposed Chiformer model achieves significantly better performance than our implemented Conformer model, with a word error rate (WER) reduction of over 0.4% on the Eval sets. The significance is evaluated by the matched pairs sentence segment word error (MPSSWE) test using the NIST ASR scoring toolkit (SCTK). This implies the effectiveness of the proposed hi-attention mechanism. Note that the only difference between the Conformer and the Chiformer architecture is the introduction of the hi-attention mechanism. The difference of our implemented Conformer and the Conformer from ESPnet official repository is mainly caused by the training setting. We use 2 GPUs while the ESPnet model uses 8 GPUs.

The Librispeech results are presented in Table VI. It can be found that the Chiformer model still achieves better or comparable performance than the Transformer and Conformer baselines. Note that the WER of 2.8% on the test-clean set is quite low, and a reduction of 0.2% is already statistically significant. This again verifies the effectiveness of the hi-attention mechanism even when the training dataset is relatively large. We also investigate the effect of the hi-attention on various parts of the Chiformer model on the Librispeech corpus. It can be found that reverting the hi-attention to self-attention in various parts results in performance degradation. On the AISHELL-2 corpus, the Chiformer achieves better performance than the Conformer with and without LM rescoring, as shown in Table VII, where the Transformer and Conformer results from ESPnet official repository are also included. These observations, consistent with those in the MT experiments, validate that the Hiformer model with hi-attention is effective compared to the Transformer counterparts.

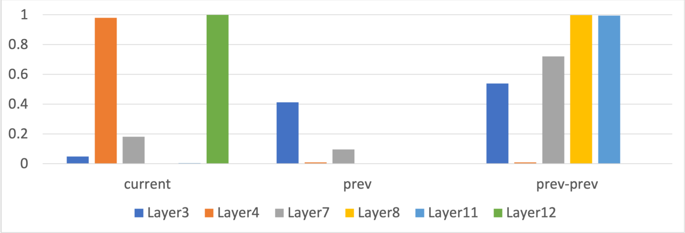

Fig. 3. Visualization of trainable weights in the encoder hi-attention for combining attention outputs from the current layer, the previous layer and the layer before the previous (prev-prev).

VII. Analysis

A. Combination Options in Hi-Attention

We compare the two options of combining hi-attention outputs from historical layers, *i.e.*, concatenating the multi-head hi-attention outputs across layers and then projecting to the model hidden space, as illustrated in Eq. (9), or simply summing up the heads of the intra-layer attention and the corresponding heads of the inter-layer attention as Eq. (10). We compare the two options on the JRC-Acquis

[3] https://github.com/espnet

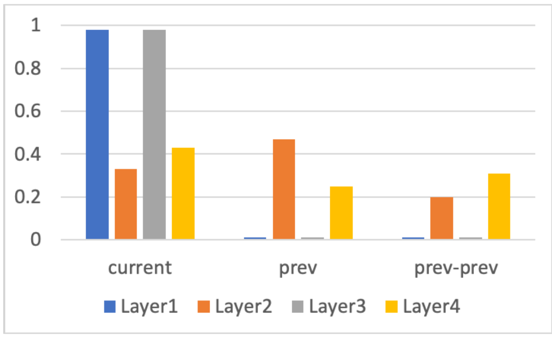

Fig. 4. Visualization of trainable weights in the decoder hi-attention for combining attention outputs from the current layer, the previous layer and the layer before the previous (prev-prev).

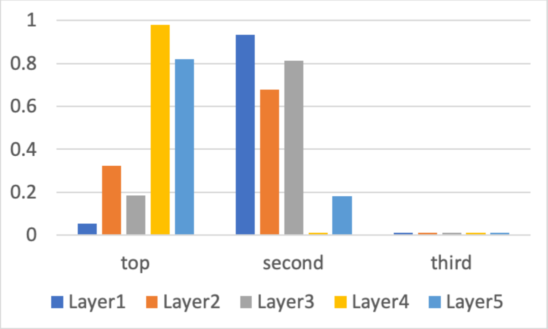

Fig. 5. Visualization of trainable weights in the cross-coder hi-attention for combining attention outputs from the top, the second and the third encoder layers.

corpus in Table II. It can be found that the concatenating option consistently outperforms the summing option on all four translation directions. We compare the model parameters and training/decoding time of the two options on the WMT'14 En⇒De translation task in Table III. The concatenating option still outperforms the summing option, but the number of model parameters is larger and the required training cost is higher.

B. Contribution of Historical Layers

To investigate how previous layers contribute to the current layer, we add trainable weights to the hi-attention for combination of the current layer's attention outputs and previous layers' attention outputs in Eq. (10) on the ASR Chiformers trained on the AISHELL-2 corpus. The trained weights in the hi-attention modules for the encoder, decoder and cross-coder are shown in Fig. 3, 4 and 5, respectively. It can be found that in the encoder hi-attention, different layers have different weight distributions on the current and the previous layers. This indicates that previous layers have different contributions to the current hi-attention at different layers. In the decoder hi-attention, the weights for the current layers are slightly larger. In the cross-coder hi-attention, the weights for the top and second encoder layers already dominate the weights, which indicates that the contributions from the second layer are as important as the top layer. The weights visualization suggests that the information from historical layers is important for the current layer. We also investigate adding regularization to the encoder to enhance the diversity of information encoded in different layers by separating the hi-attention modules in the encoder. Specifically, we follow [40] to divide the encoder of 12 layers into 3 blocks with 4 successive layers in each block. The hi-attention modules can only attend to the historical layers in the same block. The different blocks are expected to learn diversified information with such a regularization method. Though in our experiments performance degradation is observed due to the regularization, the trainable weights in the encoder hi-attention with regularization, as in Fig. 6, presents a clearer pattern where more weights are assigned to the current layers (compared to Fig. 3), while the other previous layers also make contributions to the current layers.

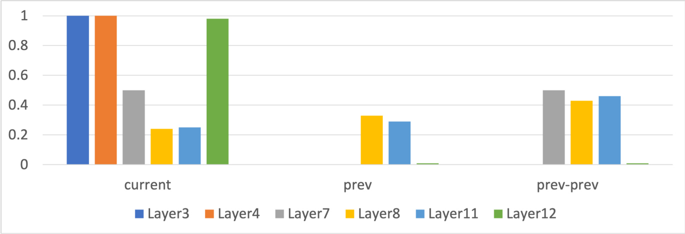

Fig. 6. Visualization of trainable weights in the encoder hi-attention with regularization for combining attention outputs from the current layer, the previous layer and the layer before the previous (prev-prev).

C. Considered Historical Layers and Dilation Factor

TABLE VIII
EFFECT OF NUMBER OF CONSIDERED HISTORICAL LAYERS AND DILATION FACTOR ON THE HIFORMER MODEL, BLEU SCORES EVALUATED ON TEST SETS OF THE JRC-ACQUIS CORPUS.

Layer	Dilation	Es⇒En	En⇒Es	De⇒En	En⇒De
0	–	64.18	61.63	60.36	55.08
1	1	64.80	61.97	**61.62**	55.81
2	1	64.88	62.05	61.39	55.97
	2	**64.91**	62.07	61.52	**56.17**
	3	64.76	**62.18**	61.19	55.59
3	1	64.83	61.84	61.09	55.67
	2	64.73	61.88	61.23	55.65

We investigate the effect of number of considered historical layers and the dilation factor on the Hiformer structure in Table VIII. Generally, using hi-attention brings performance gains. It can be also found that the configuration of considering two historical layers with a dilation factor of two achieve generally better performance than the other settings on the corpus.

D. Hiformer on Various Sentence Lengths

We analyze the performance of the Transformer and the Hiformer on various groups of sentence lengths, as in Fig. 7. We divide the WMT'14 En⇒De test set into different subsets according to sentence lengths, *i.e.*, number of words. The numbers of sentences in the intervals of <15, 15-29, 30-44 and ≥45 are 600, 1,376, 720 and 302, respectively. It can be found that the Hiformer significantly outperforms the Transformer in all four groups. This indicates the hi-attention is beneficial to modeling sequences with various lengths.

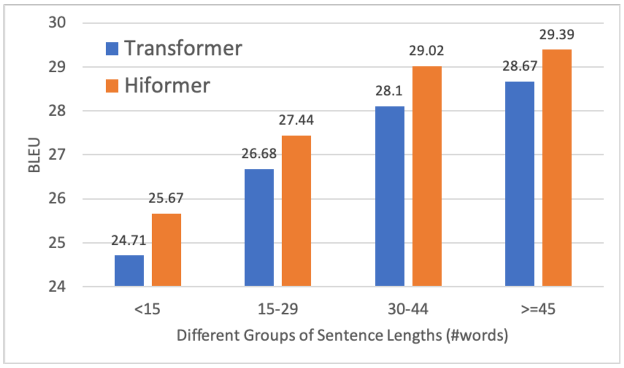

Fig. 7. Comparison between the Transformer baseline and the Hiformer on different groups of sentence lengths based on the WMT'14 En⇒De dataset.

VIII. Conclusions

In sequence modeling, recent analyses show that multi-level hierarchical information is represented at various layers of encoder-decoder structures, such as Transformer. This hierarchical information is important for structured prediction, *e.g.*, machine translation (MT) and automatic speech recognition (ASR). However, the conventional self-attention mechanism in the sequence models only considers the intra-layer information integration by retrieving the keys and values in the same layer, and lacks explicit connections with previous layers. In this work, we propose a novel structure, named Hiformer, with a hierarchical attention (hi-attention) mechanism to enhance the models' ability to leverage hierarchical information from historical layers. Inter-layer connections are explicitly established by retrieving the keys and values of attention modules in historical layers according to the queries in the current layer's attention modules. Extensive experiments conducted on the benchmark MT and ASR corpora demonstrate the effectiveness of the proposed Hiformer structure, with significant performance improvement measured by BLEU score and WER on MT and ASR tasks, respectively. In the future, we plan to apply the Hiformer structure to other tasks, *e.g.*, speech synthesis, and to semi-supervised representation learning.

References

[1] A. Gulati, J. Qin, C. Chiu, and et al., "Conformer: Convolution-augmented transformer for speech recognition," *Proc. of INTERSPEECH*, 2020.

[2] D. Cai, Y. Wang, H. Li, W. Lam, and L. Liu, "Neural machine translation with monolingual translation memory," in *Proc. of ACL*, 2021.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of NeurIPS*, 2017.

[4] H. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," *Proc. of NeurIPS*, 2021.

[5] A. Pasad, J. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2021.

[6] I. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[8] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[9] A. Graves, "Sequence transduction with recurrent neural networks," *ICML Representation Learning Workshop*, 2012.

[10] ——, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[12] Y. Wang, A. Mohamed, D. Le, and et al., "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. of ICASSP*, 2020.

[13] A. Raganato and J. Tiedemann, "An analysis of encoder representations in transformer-based machine translation," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. The Association for Computational Linguistics, 2018.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] A. Bapna, M. Chen, O. Firat, and et al., "Training deeper neural machine translation models with transparent attention," in *Proc. of EMNLP*, 2018.

[16] R. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *Proc. of NeurIPS*, 2015.

[17] J. Yang, Y. Yin, L. Yang, S. Ma, H. Huang, D. Zhang, F. Wei, and Z. Li, "GTRANS: Grouping and fusing transformer layers for neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[18] H.-Y. Huang, C. Zhu, Y. Shen, and W. Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," in *International Conference on Learning Representations*, 2018.

[19] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," in *Proc. of ICLR*, 2017.

[20] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 154–163, 2020.

[21] Z. Zhang, S. Liu, M. Li, M. Zhou, and E. Chen, "Stack-based multi-layer attention for transition-based dependency parsing," in *Proc. of EMNLP*, 2017.

[22] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. of ICML*, 2021.

[23] C. McDonald and D. Chiang, "Syntax-based attention masking for neural machine translation," in *Proc. of NAACL*, 2021.

[24] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T. Liu, "Incorporating BERT into neural machine translation," in *Proc. of ICLR*, 2020.

[25] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. of ACL*, 2016.

[26] A. Bapna and O. Firat, "Non-parametric adaptation for neural machine translation," in *Proc. of NAACL*, 2019.

[27] B. Li, Z. Wang, H. Liu, Y. Jiang, Q. Du, T. Xiao, H. Wang, and J. Zhu, "Shallow-to-deep training for neural machine translation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 995–1005. [Online]. Available: https://aclanthology.org/2020.emnlp-main.72

[28] T. He, X. Tan, Y. Xia, D. He, T. Qin, Z. Chen, and T.-Y. Liu, "Layer-wise coordination between encoder and decoder for neural machine translation," in *Proc. of NeurIPS*, 2018.

[29] X. Liu, L. Wang, D. Wong, L. Ding, L. Chao, and Z. Tu, "Understanding and improving encoder layer fusion in sequence-to-sequence learning," in *Proc. of ICLR*, 2021.

[30] Z.-Y. Dou, Z. Tu, X. Wang, S. Shi, and T. Zhang, "Exploiting deep representations for neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4253–4262.

[31] B. Yang, J. Li, D. F. Wong, L. S. Chao, X. Wang, and Z. Tu, "Context-aware self-attention networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 387–394.

[32] Y. Wang, H.-Y. Lee, and Y.-N. Chen, "Tree transformer: Integrating tree structures into self-attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1061–1070.

[33] Q. Zhang, H. Lu, H. Sak, and et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *Proc. of ICASSP*, 2020.

[34] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for ASR," in *Proc. of ICASSP*, 2018.

[35] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[36] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[37] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proc. of ACL*, 2019.

[38] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. of EMNLP*, 2004.

[39] J. Zhang, M. Utiyama, E. Sumita, G. Neubig, and S. Nakamura, "Guiding neural machine translation with retrieved translation pieces," in *Proc. of NAACL*, 2018.

[40] X. Wei, H. Yu, Y. Hu, Y. Zhang, R. Weng, and W. Luo, "Multiscale collaborative deep models for neural machine translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 414–426.

[41] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1810–1822.

[42] R. Steinberger, B. Pouliquen, A. Widiger, and et al., "The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages," in *Proc. of International Conference on Language Resources and Evaluation*, 2006.

[43] J. Gu, Y. Wang, K. Cho, and V. Li, "Search engine guided neural machine translation," in *Proc. of AAAI*, 2018.

[44] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007, pp. 177–180.

[45] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[46] G. Sun, C. Zhang, and P. Woodland, "Transformer language models with LSTM-based cross-utterance information representation," in *ICASSP*, 2021.

[47] G. Synnaeve, Q. Xu, J. Kahn, and et al., "End-to-end ASR: from supervised to semi-supervised learning with modern architectures," *Proc. of ICML Workshop on Self-supervision in Audio and Speech*, 2020.

[48] J. Carletta, S. Ashby, S. Bourban, and et al., "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*, 2005.

[49] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. of ICASSP*, 2015.

[50] J. Du, X. Na, X. Liu, and H. Bu, "AISHELL-2: Transforming Mandarin ASR Research Into Industrial Scale," *ArXiv*, Aug. 2018.

Xixin Wu (Member, IEEE) received his B.S., M.S. and Ph.D. degrees respectively from Beihang University, Tsinghua University and The Chinese University of Hong Kong. He is currently an Assistant Professor at the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong. Before this, he worked as a Research Associate with the Machine Intelligence Laboratory, Engineering Department of Cambridge University, and a Research Assistant Professor at the CUHK Stanley Ho Big Data Decision Analytics Research Centre. His research interests include speech synthesis and recognition, speaker verification, and neural network uncertainty. He is a Member of ISCA.

Hui Lu received his B.S. degree in Communication Engineering from Tongji University, Shanghai, China, in 2017. He received his M.S. degree in computer technology from Tsinghua University, Beijing, China, in 2020. He is currently pursuing his Ph.D. degree at the Human-Computer Communications Lab (HCCL) in the Chinese University of Hong Kong, Hong Kong SAR, China. His research interests include speech synthesis and voice conversion.

Kun Li received his Master degree in Computer Science from Sun Yat-sen University in 2021, and Bachelor degree in Electrical Engineering and Automation from South China University of Technology in 2015. He is now pursuing his Ph.D. degree at the Human-Computer Communications Lab (HCCL) in the Chinese University of Hong Kong, Hong Kong SAR, China. His research interests include natural language processing and text generation.

Zhiyong Wu (Member, IEEE) received the B.S. and the Ph.D. degrees in computer science and technology from Tsinghua University, Beijing, China, in 1999 and 2005, respectively. From 2005 to 2007, he was a Postdoctoral Fellow with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong (CUHK), Hong Kong. He then joined the Graduate School at Shenzhen (now Shenzhen International Graduate School), Tsinghua University, Shenzhen, China, and is currently an Associate Professor. He is also a Coordinator with Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems. His research interests include intelligent speech interaction, more specially, speech processing, audiovisual bimodal modeling, text-to-audio-visual-speech synthesis, and natural language understanding and generation. He is a Member of International Speech Communication Association and China Computer Federation.

Xunying Liu (Member, IEEE) received the Ph.D. degree in speech recognition and the M.Phil. degree in computer speech and language processing from the University of Cambridge, Cambridge, U.K., prior to his undergraduate study with Shanghai Jiao Tong University, Shanghai, China. He was a Senior Research Associate with Machine Intelligence Laboratory, Cambridge University Engineering Department, University of Cambridge, and since 2016, he has been an Associate Professor with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong. His current research interests include large vocabulary continuous speech recognition, statistical language modeling, audio-visual speech processing, machine learning, language learning, speech synthesis and assistive technology. He and his students were the recipients of a number of best paper awards and nominations, including the Best Paper Award at ISCA Interspeech2010 for the paper titled Language Model Cross Adaptation for LVCSR System Combination and the Best Paper Award at IEEE ICASSP2019 for their paper titled BLHUC: Bayesian Learning of Hidden Unit Contributions for Deep Neural Network Speaker Adaptation. He is a Member of ISCA.

Helen Meng (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA. In 1998, she joined the Chinese University of Hong Kong, Hong Kong, where she is currently the Chair Professor with the Department of Systems Engineering & Engineering Management. She was the former Department Chairman and the Associate Dean of Research with the faculty of Engineering. Her research interests include human–computer interaction via multimodal and multilingual spoken language systems, spoken dialog systems, computer-aided pronunciation training, speech processing in assistive technologies, health- related applications, and big data decision analytics. She was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING between 2009 and 2011. She was the recipient of the IEEE Signal Processing Society Leo L. Beranek Meritorious Service Award in 2019. She was also on the Elected Board Member of the International Speech Communication Association (ISCA) and an International Advisory Board Member. She is a ISCA, HKCS, and HKIE.