INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Exploring Transformers for Large-Scale Speech Recognition

Liang Lu, Changliang Liu, Jinyu Li and Yifan Gong

Microsoft

{liang.lu, chanliu, jinyli, yifan.gong}@microsoft.com

Abstract

While recurrent neural networks still largely define state-of-the-art speech recognition systems, the Transformer network has been proven to be a competitive alternative, especially in the offline condition. Most studies with Transformers have been constrained in a relatively small scale setting, and some forms of data argumentation approaches are usually applied to combat the data sparsity issue. In this paper, we aim at understanding the behaviors of Transformers in the large-scale speech recognition setting, where we have used around 65,000 hours of training data. We investigated various aspects on scaling up Transformers, including model initialization, warmup training as well as different Layer Normalization strategies. In the streaming condition, we compared the widely used attention mask based future context lookahead approach to the Transformer-XL network. From our experiments, we show that Transformers can achieve around 6% relative word error rate (WER) reduction compared to the BLSTM baseline in the offline fashion, while in the streaming fashion, Transformer-XL is comparable to LC-BLSTM with 800 millisecond latency constraint.

Index Terms: Speech recognition, Transformer, Transformer-XL

1. Introduction

State-of-the-art speech recognition systems usually rely on the recurrent neural networks (RNNs) with long short-term memory (LSTM) [1] units as their backbones. Recently, there have been an increasing interests in exploring Transformers [2] for speech recognition, inspired by their success in nature language processing such as machine translation [2] and language modeling [3]. Compared to RNNs, Transformers do not process the input signal in a sequential fashion. Instead, they reply on the self-attention mechanism to capture the temporal correlations among the sequential signals, which circumvents the expensive back-propagation through time (BPTT) [4] algorithm used to train RNNs. Therefore, Transformers can capture long-term correlations with much less computation complexity. Another advantage is that it is simpler to parallelize the computations in Transformers, which can reduce the time to train deeper models on a large scale dataset.

For speech recognition, Transformers have achieved competitive recognition accuracy compared to RNN-based counterparts within both end-to-end [5, 6, 7, 8, 9, 10] and hybrid [11, 12] frameworks. However, the superior results are usually achieved in the offline condition, while in the streaming fashion, Transformers have shown significant degradation in terms of accuracy from previous results [5, 12], even in a condition of a large latency constraint. For example, [12] reports around 25% - 40% accuracy degradation compared with the offline baseline with approximately 2.5 seconds of latency when using a hybrid model, while in [13], the streaming Transformer with around 1.2 second latency lagged behind its offline baseline by approximately 15% - 23% from experiments with an sequence-to-sequence (S2S) model. In addition, the studies of Transformer model mostly focus on a relatively small dataset from a single domain such as Librispeech dataset, and some forms of data argumentation approaches are applied to tackle the data sparsity issue. It is relatively less well understood where Transformer stands in a large-scale setting.

In this paper, we aim at understanding the behaviors of Transformers for large scale speech recognition, and performing a fair comparison to LSTMs in both offline and streaming conditions. We use around 65,000 hours of training data, and apply data-parallelization across up to 64 GPUs. We present our approaches to address the technical challenges in cross-machine multi-GPU training of deep Transformers, including model initialization and warmup training. In the offline condition, we show that our Transformers can outperform the bi-directional LSTM (BLSTM) [14] baseline with approximately 6% relative with similar number of model parameters, while in the streaming fashion, we introduce the Transformer-XL [3] based steaming model, which is computationally tractable for inference. Our results show that Transformer-XL is on par with latency-controlled BLSTM (LC-BLSTM) [15] with the same latency constraint.

2. Related Work

There have been a few studies on Transformers for end-to-end speech recognition, particularly in the context of the S2S model with attention [5, 6, 7, 16], as well as Transformer Transducers [8, 17]. In [5, 10], the authors compared RNNs with transformers for various speech recognition tasks, and obtained competitive or even better results with Transformers. Within the hybrid framework, the authors in [11, 12] also reported very strong results on the Librispeech benchmark with Transformers. The time restricted self-attention investigated in [18] is closely related to the Transformer-XL network in this work as they both perform chunk-wise training. The key difference is that in Transfromer-XL, we also take the hidden states from previous chunk as feature to capture the long-term information from the past. In terms of streamable Transformers, attention mask based approach to control the left and right context is mostly studied in the previous works, such as in the S2S model [13], Transformer-Transducer [17] and the hybrid model [12]. From the reported results, streaming Transformers still lag behind their offline counterparts remarkably in terms of the recognition accuracy, especially in the low-latency scenarios.

3. Transformer Model

3.1. Self-attention with Multiple Heads

The attention mechanism is well understood [19]. In [2], the authors introduced a machine translation model based on self-attention, which can be represented in the form of the dot-

http://dx.doi.org/10.21437/Interspeech.2020-2638

product attention as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \qquad (1)$$

where Q, K, V are referred to the query, key and value according to [2]. In self-attention, Q, K and V are from the source sequence, while in the conventional attention model [19], Q is from the decoder hidden state, and K is from the encoder hidden state. In Eq (1), d_k is the dimension of the model.

Another key idea in [2] is the multi-head attention mechanism, which performs multiple attention operations in parallel using different model parameters. The outputs from different attention heads are then concatenated and projected before being fed into the next layer, which can be shown as:

$$\text{MultiHead}(Q, K, V) = [H_1, H_2, \cdots, H_N]W^O \qquad (2)$$
$$\text{where } H_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

where N is the number of attention heads, and W_i^Q, W_i^K, W_i^V are parameters for the i-th attention head, and W^O is the projection matrix to reduce the dimension of the concatenated vector.

3.2. Depth-scale Initialization and Warmup Training

Training neural networks in a large scale setting requires parallelization across multiple GPUs. In our experiments, we did not experience convergence issues when training a shallow Transformer with random initialization on a single machine with 4-8 GPUs. However, we observed poor convergence or even divergence when performed parallelization across multiple machines with 32-64 GPUs with a randomly initialized model. To address this problem, we performed warmup training on a single machine with 4 GPUs until the model has seen ∼640 hours of training data, before switching to cross-machine parallelization. This approach worked well for Transformers with up to 12 layers in our experiments, however, when we increased the depth of the Transformers to 24 layers and beyond, the model did not converge even during the warmup stage. This is primarily due to the gradient explosion and vanishing problem in deep Transformers as investigated in [20], which shows the gradient norms of lower layers were significantly smaller that those of deeper layers. This indicates that the supervision signal becomes much weaker after back-propagation from the top layer to lower layers. To address this problem, the authors proposed the depth-scale model initialization, which normalized the ℓ_2-norm of the weight parameters by their corresponding depth index. For the model parameters from the l-th layer, they can be initialized as:

$$W \in \mathbb{R}^{d_i \times d_o} \sim \mathcal{U}\left(-\frac{\gamma}{\sqrt{l}}, \frac{\gamma}{\sqrt{l}}\right), \gamma = \sqrt{\frac{6}{d_i + d_o}}, \qquad (3)$$

where d_i and d_o are input and output dimension respectively. With depth-scale initialization, we did not observe convergence issues during warmup training, and were able to train a Transformer-XL with 100 layers and over 200 million parameters in the warmup stage. However, we did not train this model until convergence due to the high computational cost.

3.3. Pre-Norm vs. Post-Norm

Layer normalization (LN) [21] has been a *de facto* in Transformers for a smooth model convergence during training. In the seminal Transformer paper [2], LN is applied after the element-wise residual addition, i.e.,

$$x_{l+1} = \text{LN}(x_l + \mathcal{F}(x_l, \theta_l)), \qquad (4)$$

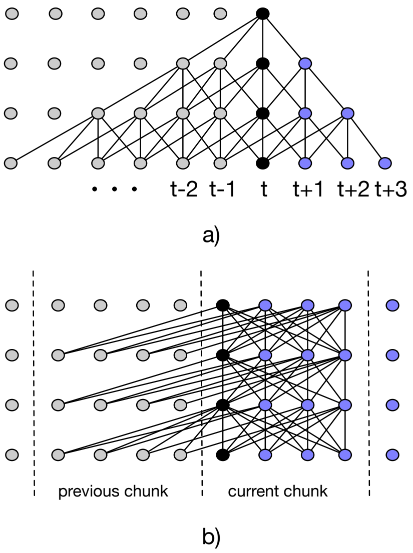

Figure 1: *a) Context expansion for a streaming Transformer based on attention masks. In this example, the attention context window is* $[-2, 1]$ *for each layer. With 3-layers, the context size increases to* $[-6, 3]$. *The gray nodes denote the past frames and hidden states, and the blue nodes represent the future frames and hidden states. b) A Transformer-XL with chunk size as 4. The context size does not changes as the model goes deeper.*

where θ_l denotes all the model parameters in the l-th layer, and $\mathcal{F}(\cdot)$ denotes any type of non-linear operation. This normalization approach, referred to as Post-Norm, is observed to results in poor convergence in machine translation tasks for deeper Transformers [22]. In our experiments, the Post-Norm approach worked well for 12-layer Transformers trained with data-parallelization on a single machine with 4-8 GPUs. However, when we run cross-machine parallelization with 32 or 64 GPUs, we observed divergence during training with Post-Norm. As investigated in [22], Post-Norm poses a high risk of vanishing or exploding gradients problem, which is more apparent in the multi-GPU training with a very large minibatch size. We tackled this problem by switching to Pre-Norm as proposed in [22], where the LN is applied as

$$x_{l+1} = x_l + \mathcal{F}(\text{LN}(x_l), \theta_l). \qquad (5)$$

The Pre-Norm approach was also applied in [12]. With this approach, we were able to train a 48-layer Transformer-XL model with 64 GPUs without any convergence issues.

3.4. Streaming Transformer

ASR systems are usually deployed in the streaming fashion which produces real-time speech transcription with certain latency constraint. To enable Transformers for streaming ASR, we have to limit the feature context visible to the model in both training and inference stage. A typical approach is based on the future context lookahead with an attention mask as shown in Figure 1.a), where each node can only access limited number of activation states from both the future and the past timesteps. However, this approach suffers from the issue of context expansion, as the context size increases as the model goes deeper. Consequently each layer can only access features from a very small context window in order to fit into the latency constraint. This hinders the model to unlock the information from the data, and possibly explain the inferior recognition accuracy observed in this work as well as in [12, 13] when employing this approach. Another disadvantage of this method is the high computational cost, as all the nodes in the context window need to be updated for every decoding step. This results in intractable computational complexity for real-time inference.

In this work, we explore another streaming approach base on the Transformer-XL network [3]. Unlike the mask-based lookahead approach, Transformer-XL consumes the input sequence in a chunk-wise fashion, which means the information is fully visible to all the nodes when they reside in the same chunk. In order to capture the information from the long-term history, Transformer-XL takes the hidden states from the previous chunk as additional feature, while does not pass the gradients through those hidden states to avoid the memory explosion. More precisely, this operation can be represented as:

$$\tilde{H}^l_{c-1} = \text{SG}(H^l_{c-1}), \tag{6}$$

$$\tilde{K}^l_c = \text{Concat}[\tilde{H}^l_{c-1}; K^l_c], \tilde{V}^l_c = \text{Concat}[\tilde{H}^l_{c-1}; V^l_c], \tag{7}$$

$$H^l_c = \text{MultiHead}(Q^l_c, \tilde{K}^l_c, \tilde{V}^l_c), \tag{8}$$

where c denote the index of the chunk, SG refers to the stop-gradient operation, and Concat indicates the concatenation operation of the two matrices along the time axis. The interpretation is that in the self-attention layer, the queries are from the current chunk, while the keys and values are from the current chunk as well as the previous chunk.

Transformer-XL enjoys two key advantages compared with the mask-based lookahead approach for streaming ASR. Firstly, we can emit the labels of all the acoustic frames within the chunk without refreshing the hidden states of self-attention, and therefore significantly cut down the computational cost for inference. Secondly, Transformer-XL enables us to train much deeper models. Recall that the self-attention operation has the memory complexity of $O(T^2)$, where T is length of the input sequence. In Transformer-XL, the length of the chunk can be much shorter, which saves memory for deeper or larger models. As aforementioned, we were able to train a Transformer-XL with 100 layers without blowing up the GPU memory in the warmup stage. However, Transformer-XL has the disadvantage of longer training time due to the chunk-wise processing.

4. Experiments and Results

4.1. Experimental setup

In our experiments, the models were trained with around 65,000 hours of anonymized and transcribed Microsoft data, recorded in various conditions. We fit the Transformer models into the hybrid architecture, where we used the hidden Markov models (HMMs) for sequence modeling. The number of tied triphone states is 9404 in our experiments, and the input feature is 80-dimension log Mel filter banks sampled every 20 milliseconds (ms) with the frame skipping strategy [23]. The language model is a 5-gram with around 100 million (M) ngrams. We used a `dev` set with around 7459 utterances, which are spontaneous conversational speech recorded in both close-talk and far-filed conditions mostly from non-native speakers. The speaking style and acoustic condition of the `dev` set are not well represented by our training data, so that we can avoid over-training by tuning on this dataset. Our `eval` set is from the far-field scenario mostly by native speakers, which has around 13,368 utterances, and it matches our training data well.

Table 1: *Results of the Transformers with convolution layers in the offline mode. The number of heads is 4, and the number of layers is 12.* `IC` *stands for interleaved 1D convolution. All models have around 50 million (M) model parameters including convolution layers.*

Model	`IC`	Size(M)	Encode layer	d_k	`dev`
Transformer	✗	51.5	Linear	620	34.7
	✓	50.0	Linear	512	20.2
	✓	51.5	VGG	512	19.6
	✗	52.0	VGG	620	19.4

Table 2: *Results of the 12-layer Transformer model with different number of attention heads. VGG net was used as the encoding layer for all the Transformers. N denotes the number of attention heads, and d_k is the model dimension as in Eq(1).*

Model	`IC`	Size (M)	N	d_k	`dev`
Transformer	✓	50.5	4	512	19.6
	✓	50.5	8	512	19.7
	✓	50.5	16	512	18.8
	✗	52.0	4	620	19.4
	✗	53.5	8	624	18.4
	✗	53.5	16	624	18.6
BLSTM	–	55.0	–	–	19.5

We compare Transformers with a bidirectional LSTM (BLSTM) based acoustic model in the offline condition, and an LC-BLSTM [15] in the streaming condition. Both BLSTM and LC-BLSTM in our experiments have 6 hidden layers, and each layer has 600 hidden units and cells for each direction. For LC-BLSTM, the chunk size is set to be 40. For Transformers, to limit the scope of our investigate, we set the dropout ratio to be 0.1, and the number of hidden units in the feedforward layer to be 2048 in all our experiments. All the models were trained with the cross-entropy criterion. We used the Adam optimizer [24] to train all our Transformer and LSTM models. For 12-layer Transformers, we set the mini-batch size as 16,000 frames for each GPU, and it took around 4 days for the models to converge with 32 Tesla V100 GPUs, which corresponds to training the models for 5-6 epochs. The training time for deeper Transformers is longer. For Transformers with 24 layers and beyond, we halved the mini-batch size due to the memory constraint, and tuned the learning rate scheduler accordingly. This increased the training time by roughly 50%. Transformer-XL is much more expensive due to the chunk-wise training fashion. Its training time is approximately 50% longer compared with a vanilla Transformer with the same model structure. For a 48-layer Transformer-XL, the training time is around 3 weeks with data parallelization across 64 Tesla V100 GPUs.

4.2. Convolution Layers and Attention Heads

The self-attention operation cannot maintain the monotonicity of input sequence, which is particularly harmful for time-synchronous acoustic model such as the hybrid model studies in this paper. The positional encoding approach in [2] is shown to be less effective to the speech recognition problem [12, 25], while convolutional layers are proven to be more powerful to encode the positional information. In Table 1, we compare the two schemes of using convolution layers in Transformers in the offline condition, namely, the interleaved 1D convolution with self-attention from our previous study [25], and using the VGG net [26] as the input encoding layer. The kernel size for the 1D convolution is 3, while the VGG net has 4 layers of 2D convolutions with 3x3 filters. When the VGG encoder was applied, we used features of frame rate as 10ms, and employed a max-pooling layer to down-sample the features by a factor of 2. As shown in Table 1, the Transformer model performed poorly without any convolution layers. Both interleaved con-

Table 3: *Results of streaming Transformer models. The number of layers is 12.*

Model	`IC`	Size (M)	N	d_k	Context	`dev`
Transformer	✓	50.5	16	512	[-∞, ∞]	18.8
	✓	50.5	16	512	[-∞, 16]	20.6
	✓	50.5	16	512	[-∞, 28]	20.7
	✓	50.5	16	512	[-∞, 40]	20.0
	✗	53.5	8	624	[-∞, ∞]	18.4
	✗	53.5	8	624	[-∞, 4]	23.0
	✗	53.5	8	624	[-∞, 16]	21.1
	✗	53.5	8	624	[-∞, 28]	21.8
	✗	53.5	8	624	[-∞, 40]	19.8
Transformer-XL	✓	50.5	16	512	[-40, 40]	20.4
	✗	53.5	8	624	[-40, 40]	21.0
BLSTM	–	55.0	–	–	$[-\infty, \infty]$	19.5
LC-BLSTM	–	55.0	–	–	[-1, 40]	20.2

volution and VGG net can significant improve the accuracy of Transformer models. In addition, when applying a VGG as the encoder, it is more beneficial to remove the interleaved convolutions but increase the model dimension of self-attention layers if the model size is constrained to be the same.

Table 2 shows the results of the Transformers with different numbers of attention heads. With the interleaved convolution, the Transformer with 16 attention heads achieved the lowest WER, while for the vanilla Transformer, 8 attention heads are sufficient. We did not further increase the number of the attention heads in our experiments due to the memory constraint. Compared to the BLSTM with around 55 million model parameters, the Transformer can achieve around 6% relative WER reduction with a similar model size.

4.3. Results of Streaming Transformers

The previous experiments focused on the offline scenario. In this section, we evaluate the accuracy of Transformers in the steaming condition. We investigated the mask-based lookahead approach, as well as the Transformer-XL network discussed in section 3.4. The results are shown in Table 3 with various latency constraints. For Transformers with attention masks, the context window in Table 3 refers to the overall accumulated latency from both convolution and self-attention. For instance, $[-\infty, 40]$ corresponds to looking ahead 3 frames for each self-attention layer in a 12-layer Transformer without interleaved convolution, with an additional 4 frames latency from the VGG encoder. Since our Transformers operated at the 20 ms frame rate, 40 frames correspond to 800 ms latency. For the attention mask based Transformer, we did not limit the left context, so it is marked as $-\infty$, while the context window as $[-\infty, \infty]$ refers to the offline system. For Transformer-XL, we set the chunk size as 40, and since the model takes the hidden states from the previous chunk as feature, we denote the context size as $[-40, 40]$. Note that there are not overlaps in the chunks during both training and inference, and it emits 40 outputs each time during inference. For LC-BLSTM, the chunk size is also 40, and because it takes the previous hidden state as the representation of history, arguably, we denote the context size as $[-1, 40]$. The chunks in LC-BLSTM are overlapped by 20 frames, so it only emits 20 outputs each time during inference.

From the results in Table 3, we have the following observations. First, without the interleaved convolution, the streaming Transformers based on attention masks degrade the recognition accuracy by 18% - 25% compared to the offline baseline model in the low latency condition. With the interleaved convolution, the accuracy loss is much smaller. This may be due to that the interleaved convolution layers can compensate the reordering effect of self-attention operations, and maintain the monotonicity of the input sequence [25]. When the latency constraint is less tight, the effect of interleaved convolution layers is diminishing. Second, Transformer-XL still lags behind the vanilla Transformer with the context window of $[-\infty, 40]$. This is not surprising as in Transformer-XL, the previous chunk is only an approximation of the full history. Improving the strength of Transformer-XL to capture the information from the long-term history in a computationally feasible manner, such as the Compressive Transformer [27], is worth further investigation. Third, the gap between the streaming Transformer (or Transformer-XL) and its offline model is larger than that between LC-BLSTM and BLSTM. The offline Transformer outperforms BLSTM by a considerable margin, while Transformer-XL is only comparable with LC-BLSTM in terms of WERs. Though the Transformer with attention mask as $[-\infty, 40]$ can outperform LC-BLSTM, it is not computationally feasible during inference. This observation may align well with the argument that Transformers are more powerful to capture the long-term correlations in sequential signals. In a scenario with limited feature context, however, Transformers are hindered to release their modeling power.

Table 4: *Results of deeper Transformer models. L denotes the model depth.*

Model	`IC`	Size(M)	L	Context	`dev`	`eval`
BLSTM	–	55.0	6	[-∞, ∞]	19.5	12.7
LC-BLSTM	–	55.0	6	[-1, 40]	20.2	12.9
Transformer	✗	53.5	12	[-∞, ∞]	18.4	11.9
	✗	97.0	12	[-∞, ∞]	18.3	–
	✗	101.7	24	[-∞, ∞]	17.8	11.7
Tranformer-XL	✗	53.5	12	[-40, 40]	21.0	12.9
	✗	101.7	24	[-40, 40]	19.1	12.4
	✓	50.5	12	[-40, 40]	20.4	12.9
	✓	95.5	24	[-40, 40]	19.3	12.6
	✓	185.7	48	[-40, 40]	18.5	12.2

4.4. Deeper Transformers

In Table 4, we show results from deeper Transformers, and results from the `eval` set. We observe similar trend on the `eval` set. We also investigated the tradeoff between increasing the dimension of hidden state in self-attention layers and increasing the depth the model. For the offline 12-layer Transformer, we increased d_k to 960, which resulted in a model with 97 million parameters. However, we only achieved very marginal improvement on the `dev` set. The gain from increasing the model depth to 24 layers is more considerable on the `dev` set, but the gain on the `eval` set is still small. It is possible that the model is overfitted, and increasing the dropout ratio may result in more gains. As for Transformer-XL, we can obtain gains in terms of accuracy by increasing the depth up to 48 layers. However, the gains are not as large as we have expected, and regularizing the deep Transformers may result in further improvements.

5. Conclusions

In this paper, we investigated Transformers for large-scale speech recognition. We presented our approaches to address the issues of scaling up Transformers with cross-machine multi-GPU training, and shown that our approach can train a Transformer with 48 layers and beyond. For streaming Transformers, we discussed the drawbacks of the attention mask based approach, and studied an alternative method based on Transformer-XL, which is much more computationally efficient for inference. We demonstrated that the offline Transformer can outperform BLSTM by 6% relative with similar number of model parameters, while the streamable Transformer-XL is comparable with LC-BLSTM.

6. References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[3] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Language modeling with longer-term dependency," in *Proc. ICLR*, 2019.

[4] P. J. Werbos *et al.*, "Backpropagation through time: what it does and how to do it," in *Proceedings of the IEEE*, vol. 78, no. 10, 1990, pp. 1550–1560.

[5] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *arXiv preprint arXiv:1909.06317*, 2019.

[6] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 5884–5888.

[7] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-attentional acoustic models," in *arXiv preprint arXiv:1803.09519*, 018.

[8] Z. Tian, J. Yi, J. Tao, Y. Bai, and Z. Wen, "Self-attention transducers for end-to-end speech recognition," in *arXiv preprint arXiv:1909.13037*, 2019.

[9] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *Proc. ICASSP*. IEEE, 2019, pp. 7115–7119.

[10] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and LSTM encoder decoder models for ASR," in *IEEE Automatic Speech Recognition and Understanding Workshop, Sentosa, Singapore*, 2019.

[11] K. J. Han, R. Prieto, K. Wu, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions," in *arXiv preprint arXiv:1910.00716*, 2019.

[12] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," in *Proc. ICASSP*. IEEE, 2020.

[13] N. Moritz, T. Hori, and J. L. Roux, "Streaming automatic speech recognition with the transformer model," *arXiv preprint arXiv:2001.02674*, 2020.

[14] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. ASRU*. IEEE, 2013, pp. 273–278.

[15] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition," in *Proc. ICASSP*, 2016.

[16] C. Wang, Y. Wu, Y. Du, J. Li, S. Liu, L. Lu, S. Ren, G. Ye, S. Zhao, and M. Zhou, "Semantic mask for transformer based end-to-end speech recognition," *arXiv preprint arXiv:1912.03010*, 2019.

[17] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *ICASSP*. IEEE, 2020, pp. 7829–7833.

[18] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for ASR," in *Proc. ICASSP*. IEEE, 2018, pp. 5874–5878.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[20] B. Zhang, I. Titov, and R. Sennrich, "Improving deep transformer with depth-scaled initialization and merged attention," in *EMNLP*, 2019.

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," in *NIPS Deep Learning Symposim*, 2016.

[22] 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 *ACL*, 2019.

[23] Y. Miao, J. Li, Y. Wang, S. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *Proc. ICASSP*, 2016.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, 2014.

[25] L. Lu, "A transformer with interleaved self-attention and convolution for hybrid acoustic models," *arXiv preprint arXiv:1910.10352*, 2019.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. ICLR*, 2015.

[27] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap, "Compressive transformers for long-range sequence modelling," *arXiv preprint arXiv:1911.05507*, 2019.