Perceiver encoder for automatic speech recognition

Kaustubh Kulkarni [1] , Venkateswarlu Gonuguntla [1, 2, †]
[1] Symbiosis Centre for Medical Image Analysis, Symbiosis International (Deemed University), Pune 412115, India
[2] Quantum AI School for Advanced Research, Alliance University, Bengaluru 560100, India

***Abstract*—The combination of convolutions, which capture local context, and self-attention mechanisms has enhanced the performance of modern automatic speech recognition (ASR) systems. In this paper, we propose the use of a perceiver encoder in ASR models. The perceiver encoder employs a cross-attention layer in the first layer to project the input space into a lower-dimensional latent space. Subsequent self-attention mechanisms operate within this latent space. We demonstrate that the perceiver encoder can compete with state-of-the-art speech encoder mechanisms on publicly available speech-datasets. Furthermore, to the best of our knowledge, this work represents the first application of the perceiver encoder to an ASR system, offering a novel approach to speech recognition.**

I. Introduction

The widespread adoption of ASR technology has transformed various aspects of modern life. The evolution of ASR systems has been marked by significant advances, from acoustic modeling with Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) systems [1] to more sophisticated deep learning architectures. The early deep learning ASR systems employed HMMs to represent acoustic models, where state output probabilities were computed using a deep neural network (DNN) [2]. These HMM-DNN systems require a framewise posterior computation using a HMM-GMM model to initialize DNN training. This process had limitations of training an additional HMM-GMM model.

End-to-end (E2E) ASR systems address these limitations by leveraging connectionist temporal classification (CTC) [3]. CTC enables direct estimation of character sequences from the posterior probabilities of underlying neural networks. These neural networks typically employ attention-based encoder-decoder (AED) models, utilizing recurrent neural network (RNN) based encoders and decoders [4] or transformer-based encoders and decoders [5] [4]. The posteriors computed by the CTC are combined with the posteriors generated by the decoder using a multitask learning framework [6]. This approach enables the model to jointly optimize both the CTC and the decoder objectives.

Recent studies have introduced several encoder architectures that combine self-attention with local feature extraction mechanisms to improve speech recognition performance.The conformer encoder [7] integrates convolutions after self-attention to incorporate local information, complementing the global information computed by self-attention. The branchformer [8] employs two parallel branches in the encoder. Self-Attention Branch which computes global dependencies. MLP branch with convolution gated units that computes local context. The E-branchformer [9] improves performance by applying a better merging method that improves the combination of global and local context.

The architectures mentioned above [8]–[10] are extensions of the conformer encoder. In this work, we propose an alternative encoder design for the ASR attention based encoder–decoder (AED) model, inspired by the perceiver architecture [11]. The perceiver was introduced to address the quadratic complexity of transformer models by achieving linear scaling with respect to input size, using cross attention between the input sequence and a set of latent representations.

A key limitation of the perceiver, however, is its mapping of variable length speech inputs to fixed length latent outputs. To address this, we introduce a sliding window mechanism that preserves the temporal variability of the speech signal. While this modification sacrifices the strict linear complexity of the original perceiver, it leads to improved performance. We theorize that the sliding window method would work as speech recognition is a monotonic task.

We evaluate the proposed perceiver based encoder on automatic speech recognition tasks, comparing it against the transformer encoder [12], [13], the conformer encoder [5], and their variants. Experimental results demonstrate that the proposed encoder achieves superior performance while maintaining a more compact model size. All models are trained and evaluated within a hybrid CTC–Attention framework [6].

II. Related Work

In this section, we summarize the relevant literature into two main categories. First, we discuss architectures in speech recognition that incorporate both global and local information. Some approaches in the literature extract global and local information sequentially [7], [10], while others extract them in parallel [8], [9]. Second, we review two efficient transformer models related to our work, which addresses the quadratic complexity of self-attention. A comprehensive survey of various efficient transformers can be found in [14]. We compare the perceiver to these models because perceiver was originally proposed to address the quadratic complexity of the self-attention layer.

† Corresponding author: Venkateswarlu Gonuguntla (e-mail: venkateswarlu.phd@gmail.com)

The conformer [7] extracts local information by applying convolutions after self-attention, sequentially. These convolutions and self-attention are sandwiched between two feedforward layers. The original conformer utilizes a single-kernel convolution block. In contrast, [10] extends this by adding a gated multiple-kernel convolution block and proposes efficient techniques for combining outputs from multiple convolution kernels. [10] also gives efficient ways to combine the outputs of the multiple convolution kernels. Branchformer and E-branchformer [8], [9] combine global and local information in parallel branches. The branchformer computes self-attention in one branch and the convolutionally gated multilayer perceptron (cgMLP) [15] in the other branch. CgMLP was successfully applied to speech recognition by [16]. The E-branchformer improves the branchformer by incorporating more efficient methods to merge the outputs of both branches [8]. Beyond speech recognition, similar methods for extracting global and local information sequentially and in parallel have been explored in natural language processing (NLP) and computer vision. However, due to space limitations, we do not discuss them in this paper.

Efficient transformers address the quadratic complexity of self-attention. Two notable models for comparison with our proposed approach are Longformer and Fastformer. Longformer [17] utilizes a sliding window mechanism to compute local attention within a fixed range while incorporating global attention at preselected token locations. In this, tokens within the window attend only to others within the same window, whereas preselected tokens with global attention attend to all tokens, and all tokens attend to these globally attended positions. This ensures that the computational complexity scales linearly with the window size. Fastformer [18] efficiently processes input sequences by deriving query, key, and value representations and extracting a single global context vector through queries by attention pooling. This global context vector is subsequently multiplied with all keys. The pooling operation is repeated in successive layers, allowing linear scalability with respect to input size.

This paper introduces a novel acoustic encoder architecture based on the perceiver model [11]. The proposed design can be viewed as an alternative to the previously discussed approaches. The perceiver replaces conventional self-attention over the input sequence with cross-attention between the input and a learnable latent array, effectively projecting the input into a lower-dimensional latent space. Subsequent self-attention operations are then performed within this latent space. In contrast to models such as longformer and fastformer, which operate directly in the original input feature space, the perceiver encoder decouples computation from input length by operating in the latent space. This property makes it a compelling architecture to explore for automatic speech recognition (ASR) tasks.

III. Model Architecture

This section presents the proposed approach for applying the Perceiver architecture [11] to speech recognition tasks. The perceiver architecture is an alternative to other speech recognition approaches as it employs a cross-attention mechanism. This mechanism maps the input space to a smaller latent space in the first layer, and subsequent self-attention layers operate within this latent space. The perceiver also produces a fixed length output to preserve the variable temporal dimension, and we use the sliding-window technique. This technique is explained in figure 1 and the perceiver encoder is explained in figure 2. In the rest of this section, we first define the notation for the speech recognition problem, then introduce the perceiver-based encoder, the sliding window technique, and the loss function.

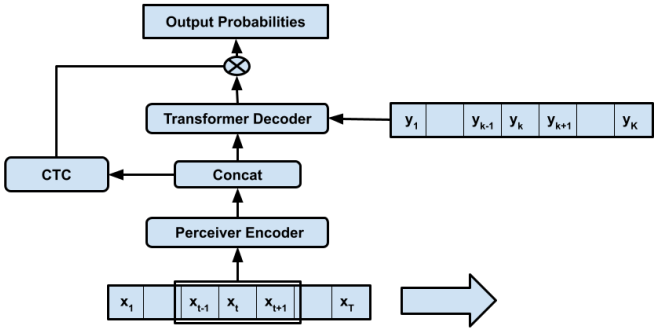

Fig. 1. Overview of the proposed architecture, incorporating the perceiver encoder and the sliding window approach. The sliding window is centered on frame x_t, with the enclosed frames processed through the perceiver encoder and subsequently concatenated to preserve the temporal dimension T.

A. Notations

Let a speech signal be represented as time-series of vectors $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots \boldsymbol{x}_T)$, where T is the total number of frames in the sequence, $\boldsymbol{x}_t$ is the observation vector belonging to the t^{th} frame and $\boldsymbol{x} \in \mathcal{R}^d$ is a d-dimensional feature. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_l, \ldots, v_L\}$ be the vocabulary of L characters. Let the sequence of characters corresponding to the sequence $\mathbf{X}$ be denoted by $\mathbf{Y} = (y_1, y_2, \ldots, y_m, \ldots, y_M)$ where y_m is the mth character in $\mathbf{Y}$, M is the number of characters in $\mathbf{Y}$ and $y_m \in \mathcal{V}$. Let $\hat{\mathbf{Y}}$ be the estimated character sequence of the multitask learning algorithm [6].

B. Perceiver encoder

The Perceiver model aims to provide a universal framework for multiple modalities. It's an efficient transformer that tackles the quadratic complexity of self-attention blocks. Figure 2 provides a visual explanation of the perceiver encoder architecture. In our work, the perceiver encoder learns hidden speech embeddings from input $\mathbf{X}$ with dimensions $T \times d$. The input is projected into keys K and values V with dimensions $T \times E$. The perceiver utilizes cross-attention in the first layer

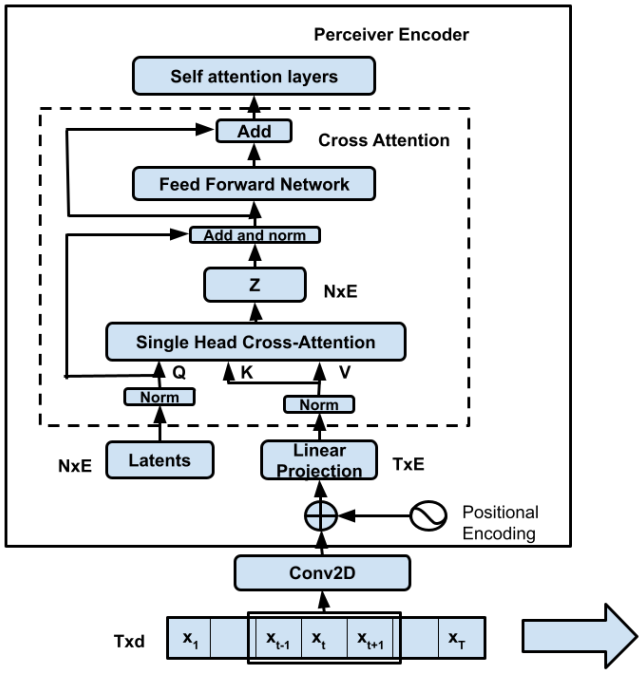

Fig. 2. Architecture of the perceiver encoder. The decoder, a standard transformer decoder, is omitted for brevity. The important block is the cross-attention block which uses the latents as queries to bottleneck the features $\mathbf{X}$ in a lower dimensions. Subsequent self-attention layers perform in this lower dimension.

as given in figure 2, where queries Q are the latents of the learnable array of dimension $N \times E$, with $T >> N$. In the first step, the multiplication of Q and K^T will create attention weights of the size $N \times T$ and after the softmax the multiplication with the values V will create the array $\mathbf{Z}$ of size $N \times E$. Subsequent layers resemble self-attention layers in transformers, but operate in $N \times E$ dimensions. This reduces the computational complexity of self-attention from $\mathcal{O}(N_{enc}T^2)$ to $\mathcal{O}(NT + N_{enc}N^2)$, where $N << T$ and N_{enc} are the number of self-attention layers of the encoder. The hyperparameter N needs to be tuned to optimize the loss function.

C. Sliding window

This section highlights the necessity of a sliding window operation (as illustrated in figure 1) in the perceiver architecture to handle time-varying speech signals. The original architecture is unsuitable for time-varying speech signals with varying temporal dimensions T. The output of the cross-attention and subsequent self-attention layers would be in $N \times E$ dimensions, losing the variable temporal dimension T. To preserve the temporal dimension, a sliding window approach is proposed. The operation of the sliding window is centered around time instant t, with a window length W (always an odd number). For each instant t, the output is an array of dimension $N \times E$, which is flattened and concatenated. The concatenated output preserves the dimension that is varying over time, resulting in a $T \times (N \times E)$ array. This array is then the input to the CTC block and the decoder. The study employs a standard transformer decoder and a standard sinusoidal positional encoding. It is important to note that while the computational complexity of self-attention is reduced as discussed in the previous section III-B. That advantage is lost in the sliding window approach as the window is applied a total of T times. We apply a convolutional subsampling layer with stride 2 after the sliding window to further reduce the temporal dimension. This was done to make the experiments fit on the single GPU. This layer is not required if multiple GPUs are present. The following section details the loss function used.

D. Loss function

The multitask learning loss is setup as the framewise CTC loss $\mathcal{L}_{ctc}$ and the labelwise decoder loss $\mathcal{L}_{dec}$. These terms are combined as a weighted sum. The weights are controlled by the hyperparameter λ. The total loss $\mathcal{L}$is estimated as given in [6].

$$\mathcal{L} = \lambda\mathcal{L}_{ctc} + (1 - \lambda)\mathcal{L}_{dec} \tag{1}$$

These two losses, CTC in the frame space and the decoder loss in the label space are combined with onepass dynamic programming [6]. The CTC applies a monotonic alignment and avoids early prediction of end-of-sentence label by the decoder.

IV. Experiments

We evaluated our model on Librispeech-100 [19] and AISHELL-1 [20] datasets. The experiments were conducted on an NVIDIA DGX machine with 4 A-100 GPUs. All Librispeech-100 and AISHELL-1 experiments were run on a single A-100 GPU. The publicly available code of the perceiver encoder was integrated into the list of encoders available with the ESPnet toolkit [4]. We use the hyperparameter settings recommended by the recipes of ESPnet, specifically for the parameter in eq. 1 λ is 0.3. For all our experiments, we used 80 dimensional log-mel features computed with $25ms$ window and $10ms$ stride. We augment the training set with speech perturbation $\{0.9, 1.0, 1.1\}$ and SpecAugment [21]. For Librispeech-100 and AISHELL-1 datasets all the comparison models consist of 12 encoder and 6 decoder layers, attention dimension of 256 and 4 attention heads. The parameters of the perceiver are discussed in IV-A.

A. Analysis of perceiver parameters

In this section, we analyze the performance of the perceiver encoder with respect to three key parameters: the number of latents N, the sliding window size W, and the number of encoder layers N_{enc}. The decoder utilized in this study is a standard transformer decoder with a fixed number of 6 layers, as recommended in most ESPnet recipes. To determine the

TABLE I
COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART-METHODS ON THE LIBRISPEECH 100HRS DATASET (WER%) AND THE AISHELL-1 DATASET (CER%). THE RESULTS ARE REPORTED FOR THE OPTIMAL VALUES OF $W = 25$, $N = 64$ AND $N_{enc} = 6$

Dataset →	Librispeech-100 (WER)					AISHELL-1 (CER)		
Method ↓	#params (M)	Dev Clean	Dev Other	Test Clean	Test Other	#params (M)	Dev	Test
Transformer [22]	40.9	8.02	20.14	8.39	20.34	47.7	5.16	5.53
Conformer [5]	37.4	6.59	17.24	6.89	17.27	54.2	4.23	4.63
Branchformer [8]	39.8	6.8	17.2	7.0	17.4	45.4	4.19	4.43
E-Branchformer [9]	38.47	6.6	16.6	6.9	17.2	45.4	4.17	4.47
Multi-Convformer [10]	37.2	5.87	16.63	6.18	17.00	54.2	4.16	4.46
Longformer [17]	24.5	6.7	17.3	7.1	17.9	36.46	4.24	4.54
Fastformer [18]	27.5	7.0	17.4	7.3	17.8	35.3	4.22	4.58
Perceiver (proposed)	24.35	**5.37**	**16.2**	**5.9**	**16.5**	29.5	**4.15**	**4.41**

TABLE II
EFFECT OF SLIDING WINDOW SIZE ON WER% WITH $N_{enc} = 6$ AND $N = 64$

W	10	15	25	35	45
Dev Clean	12.9	8.7	**5.37**	5.8	6.45
Dev Other	18.6	17.2	**16.2**	16.3	16.4
Test Clean	10.9	7.1	**5.9**	6.2	6.5
Test Other	18.9	17.4	16.5	**16.3**	16.9

TABLE III
EFFECT OF NUMBER OF LATENTS N ON WER% WITH $N_{enc} = 6$ AND $W = 25$

N	16	32	64	128	256
Dev Clean	11.6	6.8	**5.37**	6.1	6.7
Dev Other	17.2	16.3	**16.2**	17.2	17.9
Test Clean	7.2	6.5	**5.9**	6.4	6.9
Test Other	16.9	16.7	**16.5**	16.3	16.4

TABLE IV
EFFECT OF THE NUMBER OF PERCEIVER ENCODER LAYERS ON WER% WITH $N = 64$ AND $W = 25$

N_{enc}	4	6	8
Dev Clean	6.4	**5.37**	5.4
Dev Other	16.9	**16.2**	16.4
Test Clean	6.1	**5.9**	5.9
Test Other	16.9	16.5	**16.3**

optimal parameter set, we performed a grid search on three parameters. Our results indicate that the optimal configuration consists of 6 self-attention layers in the perceiver encoder (excluding the first cross-attention layer), a sliding window size of 25, and 64 latents.

To investigate the impact of each parameter on the WER (word error rate), we conducted controlled experiments in which two parameters were fixed at their optimal values while varying the third. All experiments were performed on the Librispeech-100 dataset. The effects of sliding window size W on WER with $N_{enc} = 6$ and $N = 64$, the number of latents N on WER with $N_{enc} = 6$ and $W = 25$, and the number of perceiver encoder layers N_{enc} on WER with $N = 64$ and $W = 25$, are presented in tables II, III, and IV, respectively.

The results presented in tables II, III, and IV demonstrate a consistent trend in which the WER initially decreases with varying parameters before subsequently increasing. We also observe that an increase in the window size is associated with an increase in the CTC loss, suggesting potential alignment challenges at larger window sizes. In contrast, variations in latent size influence the overall loss of the system, whereas changes in window size predominantly impact the CTC loss disproportionately. Furthermore, for the optimal perceiver training parameters of $N = 64$, $N_{enc} = 6$, and $W = 25$, figure 3 illustrates the behavior of the loss functions and their components.These components are defined in eq. 1, for the perceiver encoder. The figure demonstrates that the loss function decreases monotonically with the number of epochs.

B. Comparison of perceiver encoder with other models

This section presents a comprehensive comparison of the perceiver encoder with state-of-the-art speech recognition models, including various speech encoder architectures and efficient transformer architectures. Extensive experimental results on the Librispeech-100 and AISHELL-1 datasets are summarized in table I. The results in table I are reported for the optimal values of the perceiver parameters. The first part of table I lists different speech encoder architectures, with results obtained using ESPnet recipes. In particular, our proposed perceiver architecture outperforms all speech encoders in the Librispeech-100 dataset and achieves comparable performance on the AISHELL-1 dataset. The second part of table I presents a comparison between our perceiver encoder and two efficient transformer architectures, longformer and fastformer. Longformer utilizes a sliding window attention mechanism to compute the attention weights within a conformer block, implemented using an ESPnet recipe. Fastformer is incorporated within the branchformer architecture, employing a fastformer attention pooling mechanism to compute global attention weights. Our proposed perceiver encoder outperforms both efficient transformer methods. We hypothesize that the perceiver's ability to discover a discriminative latent space contributes to its lower error rates.

V. CONCLUSION

This paper introduces a novel speech encoder based on the perceiver architecture, a variant of efficient transformers.

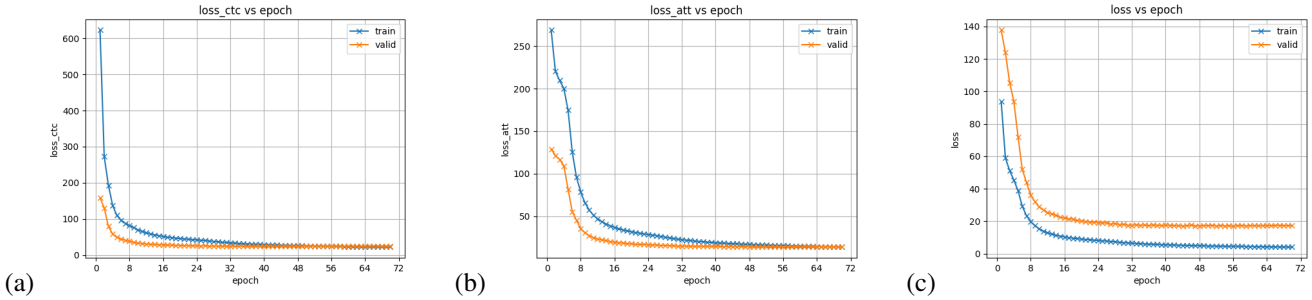

Fig. 3. Behavior of the loss functions and their components during training. (a) CTC loss of the perceiver training vs. the number of epochs. (b) Attention or the decoder loss vs. the number of epochs (c) Total loss in eq.

Our encoder leverages a cross-attention layer to project input features into a lower-dimensional latent space, followed by self-attention mechanisms operating within this space. To address the fixed-length output limitation of the perceiver encoder, we propose a sliding window approach to aggregate features along the temporal dimension. Our comprehensive analysis investigates the impact of various parameters on the perceiver encoder. Furthermore, we demonstrate that our proposed encoder outperforms several state-of-the-art methods on two publicly available datasets. Future research will delve into the advantages of operating in the perceiver's latent space. In addition, alternative feature merging techniques beyond concatenation will be investigated.

REFERENCES

[1] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[3] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural nets," in *ICML '06: Proceedings of the International Conference on Machine Learning*, 2006.

[4] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," 2018. [Online]. Available: https://arxiv.org/abs/1804.00015

[5] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020. [Online]. Available: https://arxiv.org/abs/2005.08100

[6] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[7] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020. [Online]. Available: https://arxiv.org/abs/2005.08100

[8] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, "Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding," 2022. [Online]. Available: https://arxiv.org/abs/2207.02971

[9] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," 2022. [Online]. Available: https://arxiv.org/abs/2210.00077

[10] D. Prabhu, Y. Peng, P. Jyothi, and S. Watanabe, "Multi-convformer: Extending conformer with multiple convolution kernels," 2024. [Online]. Available: https://arxiv.org/abs/2407.03718

[11] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," 2021. [Online]. Available: https://arxiv.org/abs/2103.03206

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[13] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, Dec. 2019. [Online]. Available: http://dx.doi.org/10.1109/ASRU46091.2019.9003750

[14] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2009.06732

[15] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to mlps," *CoRR*, vol. abs/2105.08050, 2021. [Online]. Available: https://arxiv.org/abs/2105.08050

[16] J. Sakuma, T. Komatsu, and R. Scheibler, "MLP-based architecture with variable length input for automatic speech recognition," 2022. [Online]. Available: https://openreview.net/forum?id=RA-zVvZLYIy

[17] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[18] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," 2021. [Online]. Available: https://arxiv.org/abs/2108.09084

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[20] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," 2017. [Online]. Available: https://arxiv.org/abs/1709.05522

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, ser. interspeech˙2019. ISCA, Sep. 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[22] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.