ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10097062

DYNAMIC CHUNK CONVOLUTION FOR UNIFIED STREAMING AND NON-STREAMING CONFORMER ASR

Xilai Li, Goeric Huybrechts, Srikanth Ronanki, Jeff Farris, Sravan Bodapati

AWS AI Labs

{lixilai, huybrech, ronanks, jjfarris, sravanb}@amazon.com

ABSTRACT

Recently, there has been an increasing interest in unifying streaming and non-streaming speech recognition models to reduce development, training and deployment cost. The best-known approaches rely on either window-based or dynamic chunk-based attention strategy and causal convolutions to minimize the degradation due to streaming. However, the performance gap still remains relatively large between non-streaming and a full-contextual model trained independently. To address this, we propose a dynamic chunk-based convolution replacing the causal convolution in a hybrid Connectionist Temporal Classification (CTC)-Attention Conformer architecture. Additionally, we demonstrate further improvements through initialization of weights from a full-contextual model and parallelization of the convolution and self-attention modules. We evaluate our models on the open-source Voxpopuli, LibriSpeech and in-house conversational datasets. Overall, our proposed model reduces the degradation of the streaming mode over the non-streaming full-contextual model from 41.7% and 45.7% to 16.7% and 26.2% on the LibriSpeech *test-clean* and *test-other* datasets respectively, while improving by a relative 15.5% WER over the previous state-of-the-art unified model.

Index Terms— End-to-end speech recognition, Unified ASR, Streaming ASR, Conformer

1. INTRODUCTION

End-to-end (E2E) automatic speech recognition (ASR) models such as attention-based encoder-decoder (AED) [1, 2], CTC [3, 4] and Transducer [5, 6, 7] have gained a lot of attention over the past decade due to their simplicity in the integration of the pronunciation, language and acoustic models into a single neural network. While state-of-the-art E2E models work remarkably well in a non-streaming fashion, they suffer from degradation when operating in a streaming manner as the requirement of transcribing text in real time poses an extra challenge. Numerous works try to bridge the gap with non-streaming ASR by training a model specific for a streaming ASR task [8, 9], focusing on mitigating the trade-off between latency and accuracy. While some slight improvements have been observed, the gap with models that take the full acoustic sequence as input (a.k.a. full-contextual models) remains non-negligible [10, 11].

In recent years, efforts have been made to unify streaming and non-streaming into a single model [12, 13, 14, 15, 16, 17], which helps reduce development, training and deployment cost. A commonly explored solution is to expose the unified model to various contexts at training time thereby making the model less susceptible to accuracy degradation at inference time under different latency conditions. In [15], a dynamic chunk training technique is adopted where the input is split into several fixed sized chunks and the audio frames within each chunk attend on themselves and frames from all the previous chunks. They vary the chunk size dynamically from 1 to the maximum utterance length in the batch, so the trained model learns to predict with arbitrary chunk size. [13] and [18] present a quite similar dynamic chunk-based attention strategy, but other methods exists too. [19] for instance, first processes input features with a streaming encoder before passing these to a non-streaming encoder. A single decoder then learns to decode either using the output of the streaming or the non-streaming encoder. [14] introduced dual-mode ASR with shared weights for both streaming and full-context speech recognition to further optimize the performance of streaming ASR. Similarly, the Dual Causal/Non-causal (DCN) self-attention network proposed in [16] processes two sequences of causal and non-causal frames in parallel and prevents the overall context to grow beyond the look-ahead of a single layer. The authors in [17] employ self-supervised pre-training with wav2vec 2.0 [20] and fine-tune the model through dual-mode training.

In general, streaming and non-streaming ASR systems are trained independently for optimized performance. These recent works on unified ASR are a great step towards a single, easy-to-use solution regardless of the inference mode. However, we identify two main shortcomings in the literature: First, the performance gap between streaming and non-streaming of an unified model still remains significant, especially when a Conformer encoder is used [13, 16]. Second, the gap between non-streaming and a full-contextual model enlarges with the increase in the amount of training data [13, 15].

In this work, we propose a *dynamic chunk convolution* (DCConv), a non-causal convolution with an improved training strategy for unified non-streaming and streaming Conformers [21]. This builds further upon the dynamic chunk training (DCT) from [15], in which the core idea is to divide the input into chunks with a chunk size that gets dynamically generated at training time. The difference lies in our novel convolution which better mimics the inference conditions at training time while keeping a rich acoustic representation and therefore results in superior performance. Besides the proposed DCConv, we extend the original DCT with two other key contributions: a) we demonstrate a better performance in both streaming and non-streaming when the model is fine-tuned from a baseline full-contextual model; b) we further optimize the streaming performance by parallelizing the convolution and self-attention modules within each Conformer block. An extensive ablation study is performed varying chunk size, overlapping chunk ratio and left context size. Empirical evaluations measured on Voxpopuli showcase the efficacy of our proposed approach in terms of the accuracy vs latency trade-off. Overall, the proposed model achieves an average relative improvement of 15.5% WER over the previous state-of-the-art [15] and obtained an absolute WER of 2.0% and 2.4% on the LibriSpeech *test-clean* dataset in the non-streaming and streaming mode, respectively.

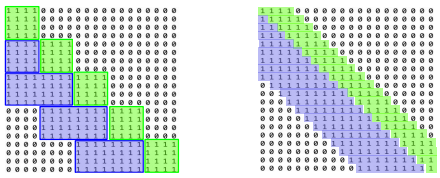

Fig. 1. (a) An example of a chunk mask of size 4, left context size 8 and sequence length 20. (b) An example of a window mask of right context size 3, left context size 8 and sequence length 20.

2. APPROACH

2.1. Model architecture

We consider a joint CTC-Attention framework [22, 23] for training our unified models. It consists of three components: a *Shared Encoder*, a *CTC Decoder* and an *Attention Decoder*. For our experiments, we only use the *CTC Decoder* at inference time and therefore we configure the *Attention Decoder* with a shallow transformer [24]. For the *Shared Encoder*, we consider two variants: the Conformer architecture [21] and a parallel Conformer [25]. The Conformer architecture consists of a regular Conformer encoder block [21] in which the convolution module appears after the multi-head self-attention (MSA). We demonstrate that using a parallel Conformer (P-Conf) encoder block [25] instead, which has the convolution and MSA existing next to each other, is beneficial for the unified streaming scenario. The advantage of the P-Conf is to capture both global and local context explicitly in the MSA and convolution branch respectively. As opposed to the recently proposed Branchformer [25], which also uses the parallel structure, we leverage the block for the streaming application too. The P-Conf reduces the overall receptive field due to its parallel nature while maintaining the same model capacity, resulting in more robust streaming performance.

2.2. Dynamic Chunk Training for Self-Attention

For unified models to perform well, they must be exposed to both limited and full context during their training. To accomplish this, [15] propose dynamic chunk training (DCT) for self-attention layers. The DCT idea involves varying the chunk size dynamically from 1 to the max utterance length for different batches in training. This is achieved by applying a dynamic chunk mask to the attention score matrix for each self-attention layer, which is illustrated in Eq. 1:

$$\text{Attn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\textbf{Mask}(\boldsymbol{Q}\boldsymbol{K}^T)/\sqrt{d})\boldsymbol{V} \quad (1)$$

where Q, K, V and d denote the queries, keys, values and embedding dimension respectively. Unlike the window mask (Fig. 1b), the chunk mask (Fig. 1a) strictly enforces the look-ahead size by setting the chunk size, while the receptive field with window masking grows linearly with the stacking of more layers. In this work, we randomly sample the chunk size between 8 (=320ms) and 32 (=1280ms) frames and the left context size between 0 and all left chunks, so that the model becomes robust to numerous sizes at inference time.

2.3. Dynamic Chunk Convolution

The convolution operator is a key component of Conformer ASR models [21]. However, the conventional convolution results in significant accuracy degradation due to the mode mismatch between training and inference, as shown in Fig. 2a. Indeed, the chunk's rightmost frames can see context from the next chunk on their right during training. Whereas at inference, this right chunk context is not available, causing a discrepancy. This inter-chunk correlation is even more magnified when stacking more Conformer blocks.

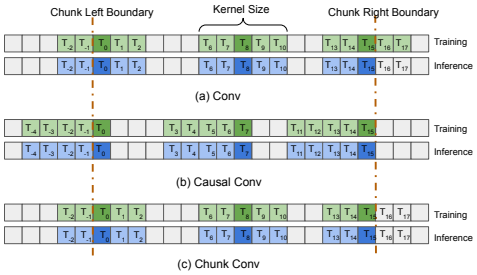

Fig. 2. Different types of convolutions: (a) regular convolution, (b) causal convolution, and (c) chunk convolution, with example kernel size 5 and chunk size 16.

One solution that is adopted in [15] consists of using a causal convolution, as shown in Fig. 2b. The left-shifted convolution kernel restricts its span from having access to any frames beyond the chunk's right boundary. This still leads to performance degradation though, as the lack of within-chunk future context for the frame being processed results in a poorer acoustic representation.

In this work, we therefore propose a novel non-causal *dynamic chunk convolution* (Fig. 2c). As opposed to the conventional convolution (Fig. 2a), the chunk convolution operator has no access to any future context beyond its right boundary. This trick allows training to more closely match the streaming inference setting where no future context beyond the right chunk boundary is available either. As opposed to causal convolution (Fig. 2b), the DCConv chunk has access to a limited within-chunk future context of the current frame. This extra within-chunk future context results in more accurate acoustic modeling and therefore better overall accuracy. The authors of [26] introduce a similar non-causal convolution. Unlike their convolution though, ours caches the output of the preceding chunk(s) and pads it to the current chunk, resulting in a superior representation. The non-causal convolution of [26] is also used in the Emformer [27] architecture for streaming use-cases with a fixed chunk size, while ours is implemented in a Conformer architecture with DCT and is applicable to both streaming and non-streaming ASR. These distinctions enable our model to be utilized in a wider range of settings.

$$\boldsymbol{X}_C^i = \boldsymbol{X}_{[iC-L:(i+1)C]} = [\boldsymbol{X}_{[iC-L:iC]}, \boldsymbol{X}_{[iC:(i+1)C]}] \quad (2)$$

$$\boldsymbol{X}_C^{i'} = \text{Conv}(\boldsymbol{X}_C^i) \rightarrow \boldsymbol{X}^{'} = \text{Concat}(\boldsymbol{X}_{C[L:]}^{i'}) \quad (3)$$

As shown in Eq. 2-3, we implement the DCConv by splitting the input sequence $\boldsymbol{X}$ into chunks $\boldsymbol{X}_C^i$ where i denotes the index of the chunk within the sequence and C the chunk size. Each chunk has a left context size $L = (kernel\_size - 1)/2$. After the convolution is applied on every chunk, we concatenate $\boldsymbol{X}_C^{i'}$ from which we have removed the first L output frames that correspond to the input left context. This DCConv operator does not slow down the training since all the chunks are independent from each other. Furthermore, we ensure to synchronize the size of both the chunk mask for the self-attention layers and for the DCConv such that the overall look-ahead size of the encoder is strictly set to the specified common size.

2.4. Fine-tuning Baseline Full-Contextual Model

Lastly, we showcase that fine-tuning from a baseline full-contextual model results in better overall performances. Instead of training a unified model from scratch, we initialize the weights from a pre-trained full-contextual model. This approach allows to leverage the non-streaming performance of the full-contextual model. Simultaneously, the model will perform better in the streaming mode too as it can transfer common speech recognition knowledge gained from the non-streaming pre-training.

Table 1. Small- and Large-Scale experiments with different architectures and training strategies. All streaming evaluations are done with a 640ms chunk size, 50% overlapping, 1280ms left context and averaged encoder latency of roughly 480ms.

		Relative WER				**Absolute WER**			
		Conversational		Multi-accent		WSJ		Voxpopuli	
Model	Training Strategy	Non-streaming	Streaming	Non-streaming	Streaming	Non-streaming	Streaming	Non-streaming	Streaming
Transformer	(A) Full-context	-	-	-	-	7.1	9.7	13.9	18.2
	(B) DCT	-11.0%	10.2%	-4.2%	6.9%	7.8	8.9	14.7	17.1
Conformer	(C) Full-context	**16.0%**	11.6%	**10.4%**	10.9%	**6.1**	8.5	12.5	16.1
	(D) DCT	14.0%	14.3%	8.9%	12.6%	**6.1**	7.6	**12.4**	15.6
	(E) DCT w/ Causal Conv	6.0%	23.1%	0.5%	8.1%	7	9.2	15	18.9
	(F) DCT w/ DCConv	9.0%	27.2%	2.6%	18.6%	6.5	7.3	13.1	14.2
	(G) + Fine-tune	**16.0%**	29.3%	9.9%	21.9%	6.4	7.2	**12.4**	14.0
P-Conformer	(H) Full-context	**16.0%**	14.3%	**10.4%**	12.6%	6.3	8.3	**12.4**	16
	(I) DCT w/ DCConv	8.0%	27.9%	2.1%	19.4%	6.5	7.3	13.0	14.0
	(J) + Fine-tune	**16.0%**	**30.6%**	9.9%	**22.3%**	6.2	**6.9**	12.6	**13.8**
Conformer-Large	(K) Full-context	-	-	-	-	**4.5**	6.2	9.2	12
	(L) DCT w/ DCConv + Fine-tune	**1.9%**	**22.9%**	**0.0%**	**11.0%**	4.6	**5.6**	**9.1**	**10.5**

3. EXPERIMENTAL SETTINGS

3.1. Datasets

Training data: We consider 3 different speech corpora varying in size for training our models: A *large-scale* 50k+ hour English corpus and a *small-scale* 5k hour subset, sampled from in-house paired audio and text data. Both corpora include audio files with a good mix of accents, speakers, sampling rates and background noise. The third dataset is the open-source *LibriSpeech* [28] corpus, for which we combine *train-clean-100*, *train-clean-360* and *train-other-500* to have 960 hours of training data. These 3 data regimes are representative of a wide range of end-to-end ASR systems for various speech applications.

Benchmarking: For the LibriSpeech experiments in section 4.3, we evaluate our models on *test-clean* and *test-other*. For the small- and large-scale experiments in sections 4.1 and 4.2 respectively, we use the following test sets: (1) *Conversational*: A 10+ hour in-house dataset with utterances resembling user inputs to goal oriented conversational dialog systems. The average utterance length is roughly 10 words; (2) *Multi-accent*: A 100+ hour in-house long-form audio dataset, composed of 12 different accents spoken across the US. The average utterance length is roughly 16 words after segmentation; (3) *Wall Street Journal (WSJ)*: We use WSJ's eval_test92 [29], prepared using Kaldi's [30] WSJ recipe. The dataset is 0.7h long. The average utterance length is 16 words; (4) *Voxpopuli* [31]: We use the English test partition, which is 4.9h long. The average utterance length is 24 words. We report absolute word error rate (WER) on WSJ and Voxpopuli and relative WER (WERR) on in-house datasets.

3.2. Experiment Setup

For training, we use the AED architecture with a Conformer as the encoder, and a shallow single-layer transformer [24] as the attention-based decoder. For *LibriSpeech experiments*, we use a Conformer-12x512x8, which consists of 12 encoder layers with 512 feature dimensions and 8 self-attention heads. We train a 24-layered transformer-based neural LM on the *librispeech-train* dataset to use for rescoring. For the *small-scale experiments* we use a Conformer-16x512x4, whereas for the *large-scale experiments* we use a Conformer-20x512x8. The kernel size of our convolution modules is 31. We optimise our model via the hybrid CTC and attention losses. All of our models are trained using ESPNet [32], with the Adam optimizer [33] and a warm-up learning rate scheduler.

For front-end, we use 80 dimensional log-mel features and SpecAugment [34] to perform data augmentation. The BPE embedding is 1024 and 2048 for the small- and large-scale experiments respectively. We train a 4-gram LM on the training text for shallow fusion. For evaluation, we discard the attention-based decoder and only use the CTC decoder to generate outputs with a CTC prefix beam search and beam size of 50. A CTC decoder optimises the real-time factor (RTF) compared to the attention-based decoder, as the latter is non-autoregressive and also needs triggered attention [35] for streaming inference and is therefore slower. We opt for the CTC decoder as ensuring a low RTF is key for streaming applications.

4. RESULTS

4.1. Ablation Study on Small-Scale Model

In the next subsections, we start with discussing the different contributions of our work by analyzing the small-scale results in Table 1.

4.1.1. DCConv vs Causal and Normal Convolutions

We demonstrate the effectiveness of our novel DCConv by performing an ablation study on three different models: a DCT model with normal convolution (D), a DCT model with causal convolution (E), and a DCT model with our own DCConv (F). Our model outperforms the two baselines on every dataset in the streaming mode. In the non-streaming application on the other hand, our model always beats the DCT model with causal convolution, but does degrade the model with regular convolution. Devising our convolution in such a way that within-chunk future context is used for a more informative acoustic representation (as opposed to (D)), while refraining of using outside-chunk future context to match the training and inference modes more closely (as opposed to (E)), is therefore empirically shown to be advantageous in most cases.

4.1.2. Fine-tuning from Full-Contextual Model

Fine-tuning (G) from a pre-trained full-contextual model instead of training a DCConv model from scratch (F) leads to improvements in both the non-streaming and streaming mode for every single dataset. We observe an average relative WER improvement of 5.5% and 2.4% in the non-streaming and streaming modes respectively. Furthermore, with the exception of the WSJ dataset, we now always outperform the regular convolution model in the non-streaming mode. Fine-tuning helps as the model relies on previously gained knowledge from a pre-trained model instead of learning from scratch. It maintains and even improves the non-streaming performance of the full-contextual model, while boosting the streaming performance as it leverages previously learned speech recognition knowledge common to both streaming modes.

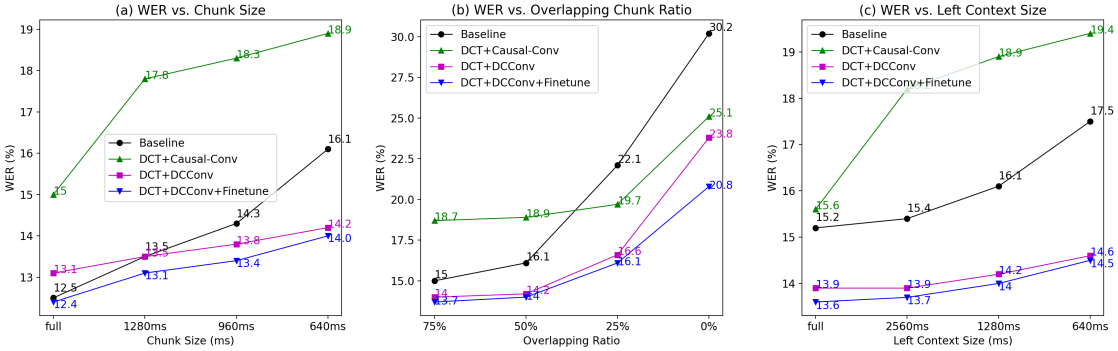

Fig. 3. Ablation study on small-scale models on how the chunk size, overlapping ratio and left context size affect the streaming decoding WER on the Voxpopuli testset. (a) Different chunk sizes with 50% overlapping and 1280ms left context. (b) Different overlapping ratios with 640ms chunk size and 1280ms left context. (c) Different left context sizes with 640ms chunk size and 50% overlapping chunk decoding.

4.1.3. P-Conf vs Conformer

Our experiments indicate that using a P-Conf instead of a regular Conformer is beneficial in the streaming mode, while it shows similar results in non-streaming. This holds for the full-contextual (H), DCConv (I) and fine-tuned DCConv (J) models, where using a P-Conf (J) leads to an relative WER improvement of 1.9%, 0.5%, 4.2% and 1.4% respectively across the 4 datasets over Conformer (G). We hypothesize that the lower receptive field of the P-Conf as a result of the parallel MSA and convolution model during training makes the model more robust in the streaming mode, as it more closely matches the inference conditions.

4.1.4. Ablation Study with Different Streaming Parameters

We take a closer look at the latency vs accuracy trade-off impact with different model parameters that can be controlled during streaming inference. In Fig. 3, we run an ablation study on the DCT hyper-parameters. Fig. 3a demonstrates improvement in WER with the increase in chunk size. Likewise, Fig. 3b and Fig. 3c reveal better WER performance with more overlap between successive chunks and greater left context size. All are the result of a wider and therefore superior acoustic representation. However, these improvements go at the expense of speed due to the latency-accuracy trade-off. Depending on the use-case we can then select one or the other setting. More importantly, we observe for all settings that our fine-tuned DCConv model performs the best, illustrating the robustness of our approach. In our subsequent experiments, we opt for a 640ms chunk with a 1280ms left context in order to keep those values low to mimic a real-life streaming setting. Furthermore, we stick to a 50% overlapping ratio as this ratio performs only slightly worse than the 75% ratio but provides better latency.

4.2. Result with Large-Scale Model

In Table 1, we also compare a fine-tuned DCConv Conformer model (L) to a baseline full-contextual model (K) on a large-scale dataset. The results show no degradation (except for WSJ), and even minor improvements, in the non-streaming mode. This is despite the fact that our model was trained in a unified fashion. In the streaming mode, we observe an average WERR improvement of 14.0% across all datasets. This validates the gains of our suggested contributions.

4.3. Results on LibriSpeech

In Table 2, we illustrate the performance of our proposed approach when trained and tested on the widely used public LibriSpeech dataset. First, we compare our DCConv model (C) to a full-contextual model trained without DCT (A) and a DCT model with a regular causal convolution (B) in a non-streaming and a 50% overlapping streaming mode. We observe a 28.9% WERR improvement compared to the full-contextual model in the streaming mode, emphasizing the utility of DCT training. Additionally, we notice an average 7.9% WERR improvement compared to the DCT model with regular causal convolution for both streaming modes, proving the effectiveness of our devised convolution. Furthermore, we demonstrate that fine-tuning (D) instead of training a DCConv model from scratch is especially helpful if you wish to keep a high non-streaming performance of your unified model. It even outperforms the full-contextual model by a WERR improvement of 5.3% in the non-streaming mode, while keeping the overlapping streaming performance almost intact. Lastly, we observe a minor 2.8% relative WER improvement when using a P-Conf (F) instead of the conventional one (D) in the streaming mode.

Table 2. LibriSpeech experiment. Chunk size of 640ms and left context size of 1280ms for 50% overlapping chunk streaming.

Model	test-clean		test-other	
	Non-streaming	Streaming	Non-streaming	Streaming
(A) Conformer (Full-context)	2.1	3.6	5.1	9.4
(B) + DCT w/ causal conv	2.6	2.9	5.8	6.8
(C) + DCT w/ DCConv	2.3	2.6	5.4	**6.6**
(D) + Fine-tune	**2.0**	**2.5**	**4.8**	**6.6**
(E) P-Conformer (Full-context)	2.1	3.4	4.9	9.1
(F) + DCT w/ DCConv + Fine-tune	**2.0**	**2.4**	**4.8**	**6.5**

Overall, compared to the full-contextual Conformer model (A), our final fine-tuned DCConv P-Conf model (F) improves the WER by 32.1% in the streaming mode and reduces the degradation of that mode over the non-streaming full-contextual model from 41.7% and 45.7% to 16.7% and 26.2% on the *test-clean* and *test-other* datasets respectively. Moreover, we further improve on the state-of-the-art model (B) that reduces this streaming gap too by an average 15.5% WERR across the 4 settings.

5. CONCLUSION

In this work, we propose a novel *dynamic chunk convolution* that further improves the existing dynamic chunk training. We achieve this as our convolution better mimics the inference conditions at training time, while keeping a rich acoustic representation. Additionally, we introduce a fine-tuning mechanism and a parallel Conformer block for the unified ASR setting. Our results demonstrate that our unified streaming model closes and even exceeds the gap with a full-contextual model operating in a non-streaming mode, while also showcasing improvements in the streaming mode under different latency constraints. Overall, we outperform the previous state-of-the-art by an average 15.5% WERR across the LibriSpeech datasets.

6. REFERENCES

[1] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[2] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling unsegmented sequence data with Recurrent Neural Networks," in *Proc. ICML*, 2006, pp. 369–376.

[4] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. ICML*, 2016, pp. 173–182.

[5] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[7] Siddharth Dalmia, Yuzong Liu, Srikanth Ronanki, and Katrin Kirchhoff, "Transformer-transducers for code-switched speech recognition," in *Proc. ICASSP*, 2021, pp. 5859–5863.

[8] Tara N Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, et al., "Two-pass end-to-end speech recognition," *arXiv preprint arXiv:1908.10992*, 2019.

[9] Chung-Cheng Chiu and Colin Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382*, 2017.

[10] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar, "Transformer Transducer: A streamable speech recognition model with Transformer encoders and RNN-T loss," in *Proc. ICASSP*.

[11] Wenyong Huang, Wenchao Hu, Yu Ting Yeung, and Xiao Chen, "Conv-Transformer Transducer: Low latency, low frame rate, streamable end-to-end speech recognition," *arXiv preprint arXiv:2008.05750*, 2020.

[12] Anshuman Tripathi, Jaeyoung Kim, Qian Zhang, Han Lu, and Hasim Sak, "Transformer transducer: One model unifying streaming and non-streaming speech recognition," *arXiv preprint arXiv:2010.03192*, 2020.

[13] Binbin Zhang, Di Wu, Zhuoyuan Yao, Xiong Wang, Fan Yu, Chao Yang, Liyong Guo, Yaguang Hu, Lei Xie, and Xin Lei, "Unified streaming and non-streaming two-pass end-to-end model for speech recognition," *arXiv preprint arXiv:2012.05481*, 2020.

[14] Jiahui Yu, Wei Han, Anmol Gulati, Chung-Cheng Chiu, Bo Li, Tara N Sainath, Yonghui Wu, and Ruoming Pang, "Dual-mode ASR: Unify and improve streaming ASR with full-context modeling," *arXiv preprint arXiv:2010.06030*, 2020.

[15] Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei, "WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," *arXiv preprint arXiv:2102.01547*, 2021.

[16] Niko Moritz, Takaaki Hori, and Jonathan Le Roux, "Dual causal/non-causal self-attention for streaming end-to-end speech recognition," in *Proc. Interspeech*, 2021, pp. 1822–1826.

[17] Chunxi Liu, Yuan Shangguan, Haichuan Yang, Yangyang Shi, Raghuraman Krishnamoorthi, and Ozlem Kalinli, "Learning a dual-mode speech recognition model via self-pruning," *arXiv preprint arXiv:2207.11906*, 2022.

[18] Kwangyoun Kim, Felix Wu, Prashant Sridhar, Kyu J Han, and Shinji Watanabe, "Multi-mode Transformer Transducer with Stochastic Future Context," *arXiv preprint arXiv:2106.09760*, 2021.

[19] Arun Narayanan, Tara N Sainath, Ruoming Pang, Jiahui Yu, Chung-Cheng Chiu, Rohit Prabhavalkar, Ehsan Variani, and Trevor Strohman, "Cascaded encoders for unifying streaming and non-streaming ASR," in *Proc. ICASSP*.

[20] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[21] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented Transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[22] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, 2017, pp. 4835–4839.

[23] Saket Dingliwal, Monica Sunkara, Srikanth Ronanki, Jeff Farris, Katrin Kirchhoff, and Sravan Bodapati, "Personalization of CTC speech recognition models," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 302–309.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe, "Branchformer: Parallel MLP-Attention Architectures to Capture Local and Global Context for Speech Recognition and Understanding," in *Proc. ICML*, 2022, pp. 17627–17643.

[26] Yangyang Shi, Chunyang Wu, Dilin Wang, Alex Xiao, Jay Mahadeokar, Xiaohui Zhang, Chunxi Liu, Ke Li, Yuan Shangguan, Varun Nagaraja, et al., "Streaming transformer transducer based speech recognition using non-causal convolution," in *Proc. ICASSP*, 2022, pp. 8277–8281.

[27] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *Proc. ICASSP*, 2021, pp. 6783–6787.

[28] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[29] John Garofolo, David Graff, Doug Paul, and David Pallett, "CSR-I (WSJ0) Complete LDC93S6A," *Web Download. Philadelphia: Linguistic Data Consortium*, vol. 83, 1993.

[30] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. IEEE workshop on automatic speech recognition and understanding*, 2011.

[31] Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

[32] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "ESPNet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[33] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[35] Niko Moritz, Takaaki Hori, and Jonathan Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. ICASSP*, 2019, pp. 5666–5670.