MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Dual Causal/Non-Causal Self-Attention for Streaming End-to-End Speech Recognition

Moritz, Niko; Hori, Takaaki; Le Roux, Jonathan

TR2021-094 August 26, 2021

Abstract

Attention-based end-to-end automatic speech recognition (ASR) systems have recently demonstrated state-of-the-art results for numerous tasks. However, the application of self-attention and attention-based encoder-decoder models remains challenging for streaming ASR, where each word must be recognized shortly after it was spoken. In this work, we present the dual causal/non-causal self-attention (DCN) architecture, which analyzes a fixed number of look-ahead frames at each self-attention layer of a deep neural network, without causing the overall context to grow beyond the look-ahead of a single layer when using multiple DCN layers. DCN is compared to chunk-based and restricted self-attention using streaming transformer and conformer architectures, showing improved ASR performance over restricted self-attention and competitive ASR results compared to chunk-based self-attention, while providing the advantage of frame-synchronously processing input and output frames. Combined with triggered attention, the proposed streaming end-to-end ASR systems obtained state-of-the-art results on the LibriSpeech, HKUST, and Switchboard ASR tasks.

Interspeech 2021

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Dual Causal/Non-Causal Self-Attention for Streaming End-to-End Speech Recognition

Niko Moritz, Takaaki Hori, Jonathan Le Roux

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

{moritz, thori, leroux}@merl.com

Abstract

Attention-based end-to-end automatic speech recognition (ASR) systems have recently demonstrated state-of-the-art results for numerous tasks. However, the application of self-attention and attention-based encoder-decoder models remains challenging for streaming ASR, where each word must be recognized shortly after it was spoken. In this work, we present the dual causal/non-causal self-attention (DCN) architecture, which in contrast to restricted self-attention prevents the overall context to grow beyond the look-ahead of a single layer when used in a deep architecture. DCN is compared to chunk-based and restricted self-attention using streaming transformer and conformer architectures, showing improved ASR performance over restricted self-attention and competitive ASR results compared to chunk-based self-attention, while providing the advantage of frame-synchronous processing. Combined with triggered attention, the proposed streaming end-to-end ASR systems obtained state-of-the-art results on the LibriSpeech, HKUST, and Switchboard ASR tasks.

Index Terms: speech recognition, triggered attention, dual causal/non-causal self-attention, streaming ASR

1. Introduction

In recent years, end-to-end (E2E) automatic speech recognition (ASR) has superseded conventional hybrid DNN/HMM solutions [1, 2]. Three main reasons can be noted for this development: 1) E2E ASR solutions have a much less complex training pipeline, 2) the inference process of E2E ASR systems can be optimized more easily, and 3) E2E ASR models have proven to achieve competitive or better performance compared to conventional hybrid solutions. The most popular E2E ASR models are based on the connectionist temporal classification (CTC) [3], the RNN transducer (RNN-T) [4], and/or an attention-based encoder-decoder architecture [5, 6]. Besides the progress that has been made in building E2E ASR systems [7, 8], more advanced neural network architectures are an important factor in the dramatic improvement of ASR results in the past decade. The most successful ASR architectures are transformer and conformer-based [9, 10], which both apply self-attention and source (or encoder-decoder) attention for analyzing temporal information of a sound signal. However, both attention types typically require a full input sequence corresponding to an entire speech utterance, e.g., extracted using speech activity detection, even for recognizing an ASR output corresponding to the beginning of a long utterance.

Several methods have been proposed to improve the streaming properties of source attention in order to generate ASR outputs with a controlled delay. The neural transducer [11, 12] and blockwise synchronous beam search [13] both apply the source attention on chunks of encoder frames with a fixed striding instead of on the full encoder sequence corresponding to a complete utterance. Other methods perform the chunking with an adaptive stride, e.g., by computing a selection probability such as Monotonic Chunkwise Attention (MoChA) [14, 15] or by detecting word boundaries using a scout network [16]. In [17], we introduced triggered attention (TA), which exploits the temporal alignment properties of CTC to identify encoder frames containing relevant information for an ASR output and to trigger an attention-based ASR decoder at such time positions. Unlike other solutions, TA provides a frame-synchronous one-pass decoding algorithm with joint CTC/attention scoring [18, 19].

For streaming E2E ASR, restricted self-attention (RSA) and chunk-based self-attention (CSA) are often used [19–22], where the latter has been shown to achieve better ASR performance but with the drawback of working in a non-frame-synchronous manner and requiring higher computational costs, which is often addressed by compressing the past context using context inheritance or augmented memory [13, 23].

In this work, we present the dual causal/non-causal self-attention (DCN) architecture for streaming E2E ASR. DCN provides frame-synchronous processing with a fixed look-ahead size without causing an increasing receptive field for a growing number of consecutive DCN layers, which is unlike the delay aggregation of deep RSA architectures [19, 20]. We combine DCN-based encoder models with TA for streaming E2E ASR using transformer- and conformer-based neural network architectures. RSA, DCN, and CSA-based ASR results are compared using the LibriSpeech, HKUST, and Switchboard tasks. For all ASR tasks, the proposed streaming E2E ASR system achieves state-of-the-art results. In addition, the decoding emission delays of various TA-based models are analyzed, demonstrating that TA decoding is well aligned with the true time positions of words and that streaming models with less look-ahead are more prone to delaying ASR decoding outputs.

2. System Architecture

ASR systems in this work make extensive use of the scaled dot-product attention mechanism, which can be written as

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

where $Q \in \mathbb{R}^{n_q \times d_q}$, $K \in \mathbb{R}^{n_k \times d_k}$, and $V \in \mathbb{R}^{n_v \times d_v}$ are the queries, keys, and values, where the d_* denote dimensions and the n_* denote sequence lengths, $d_q = d_k$, and $n_k = n_v$. Multiple attention heads may be used by

$$\text{MHA}(\hat{Q}, \hat{K}, \hat{V}) = \text{Concat}(\text{Head}_1, \ldots, \text{Head}_{d_h})W^{\text{H}}, \quad (2)$$

$$\text{and Head}_i = \text{Attention}(\hat{Q}W_i^Q, \hat{K}W_i^K, \hat{V}W_i^V), \quad (3)$$

where $\hat{Q}$, $\hat{K}$, and $\hat{V}$ are inputs to the multi-head attention (MHA) layer, Head_i represents the output of the i-th attention head for a total number of d_h heads, $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_q}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ as well as $W^H \in$

$\mathbb{R}^{d_h d_v \times d_{\text{model}}}$ are trainable weight matrices with typically $d_k = d_v = d_{\text{model}}/d_h$, and Concat denotes concatenation.

In this work, transformer and conformer architectures are employed in a joint CTC/attention based E2E ASR system [10,24]. Both architectures first extract 80-dimensional log-mel spectral energies plus 3 extra features for pitch information [25]. The derived feature sequence X is processed by a two-layer convolutional neural network (CNN) module, which generates feature sequence X_0 with a frame rate of 40 ms [19].

For $e = 1, \ldots, E$ blocks, the **transformer** encoder applies multi-head self-attention (MHA), a feed-forward neural network module (FF), and layer normalization (LN) as follows:

$$\tilde{X}_e = \text{LN}_{e,1}(X_{e-1}) \tag{4}$$

$$\overline{X}_e = X_{e-1} + \text{MHA}_e(\tilde{X}_e, \tilde{X}_e, \tilde{X}_e), \tag{5}$$

$$X_e = \overline{X}_e + \text{FF}_e(\text{LN}_{e,2}(\overline{X}_e)), \tag{6}$$

where each FF_e module consists of two linear layers of inner dimension d_{in} with a ReLU non-linearity in between. A sinusoidal positional encoding (PE) [26] is added to X_0 prior to feeding it to the transformer neural network. Dropout with a probability of 0.1 is used after self-attention and after each layer of the FF module.

The **conformer** encoder is composed of $e = 1, \ldots, E$ conformer blocks that can be written as

$$\tilde{X}_e = X_{e-1} + \frac{1}{2}\text{FF}_{e,1}(\text{LN}_{e,1}(X_{e-1})), \tag{7}$$

$$\tilde{\tilde{X}}_e = \text{LN}_{e,2}(\tilde{X}_e) \tag{8}$$

$$\overline{X}_e = \tilde{X}_e + \text{MHA}^{\text{pos}}_e(\tilde{\tilde{X}}_e, \tilde{\tilde{X}}_e, \tilde{\tilde{X}}_e), \tag{9}$$

$$\overline{\overline{X}}_e = \overline{X}_e + \text{Conv}_e(\text{LN}_{e,3}(\overline{X}_e)), \tag{10}$$

$$X_e = \overline{\overline{X}}_e + \frac{1}{2}\text{FF}_{e,2}(\text{LN}_{e,4}(\overline{\overline{X}}_e)), \tag{11}$$

where MHA^{pos} denotes multi-head self-attention with relative positional encoding [10], Conv is a convolution module, and both FF module architectures are similar to FF of the transformer. Dropout with probability 0.1 is used after MHA^{pos}, Conv, and each layer of FF. The Conv module starts with a pointwise convolution layer and a gated linear unit (GLU) followed by a 1-D depthwise convolution, batch normalization, Swish activation, and another pointwise convolution layer [10].

A CTC and an attention decoder branch are jointly trained using the multi-objective loss function

$$\mathcal{L} = -\gamma \log p_{\text{ctc}} - (1-\gamma) \log p_{\text{dec}}, \tag{12}$$

where p_{ctc} and p_{dec} denote the CTC and attention decoder loss with hyperparameter γ controlling the weighting between the two. In the **CTC** branch, the encoder output X_E is layer normalized and frames are projected to a probability distribution of the size of the ASR labels plus one for the CTC blank symbol by using a linear layer followed by a softmax function. In the **decoder** branch, a decoder model is trained using

$$p_{\text{dec}}(Y|X_E) = \prod_{l=1}^{L} p(y_l|\boldsymbol{y}_{1:l-1}, X_E) \tag{13}$$

with label sequence $Y = (y_1, \ldots, y_L)$, label subsequence $\boldsymbol{y}_{1:l-1} = (y_1, \ldots, y_{l-1})$, and the encoder output sequence X_E. The posterior probability $p(y_l|\boldsymbol{y}_{1:l-1}, X_E)$ is computed by the decoder model as follows:

$$\boldsymbol{z}^0_{1:l} = \text{EMBED}(\langle \text{s} \rangle, y_1, \ldots, y_{l-1}) + \text{PE}, \tag{14}$$

$$\tilde{X}_E = \text{LN}(X_E), \tag{15}$$

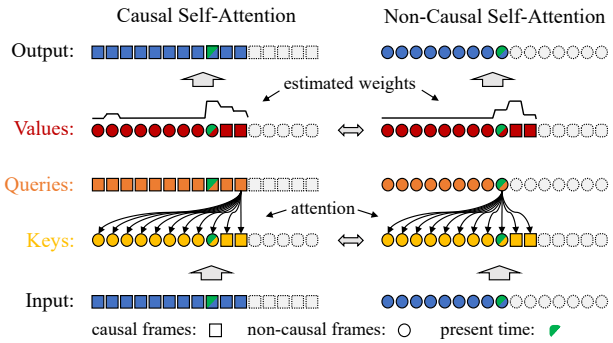

Figure 1: *Dual causal/non-causal self-attention (DCN) with 2 look-ahead frames.*

$$\hat{\boldsymbol{z}}^d_{1:l} = \text{LN}_{d,1}(\boldsymbol{z}^{d-1}_{1:l}), \tag{16}$$

$$\overline{\boldsymbol{z}}^d_l = \boldsymbol{z}^{d-1}_l + \text{MHA}^{\text{self}}_d(\hat{\boldsymbol{z}}^d_l, \hat{\boldsymbol{z}}^d_{1:l}, \hat{\boldsymbol{z}}^d_{1:l}), \tag{17}$$

$$\overline{\overline{\boldsymbol{z}}}^d_l = \overline{\boldsymbol{z}}^d_l + \text{MHA}^{\text{src}}_d(\text{LN}_{d,2}(\overline{\boldsymbol{z}}^d_l), \tilde{X}_E, \tilde{X}_E)), \tag{18}$$

$$\boldsymbol{z}^d_l = \overline{\overline{\boldsymbol{z}}}^d_l + \text{FF}_d(\text{LN}_{d,3}(\overline{\overline{\boldsymbol{z}}}^d_l)), \tag{19}$$

for $d = 1, \ldots, D$, where D denotes the number of decoder blocks, PE is a sinusoidal positional encoding, EMBED converts the input label sequence $(\langle \text{s} \rangle, y_1, \ldots, y_{l-1})$ into a sequence of trainable embedding vectors $\boldsymbol{z}^0_{1:l}$, and $\langle \text{s} \rangle$ denotes a start of sentence symbol. Finally, output vector $\boldsymbol{z}^D_l$ is further processed using LN, a linear layer, and a softmax function to compute a probability distribution over the ASR labels.

For **streaming** E2E ASR, model components that process temporal information must be restricted to only use a limited future context for reducing the output latency. In the presented architectures, the self-attention layers of Eqs. (5) and (9), the Conv module of Eq. (10), and the source attention of Eq. (18) typically require a wide temporal context. RSA [19,20,26], CSA [13,21], and the dual causal/non-causal (DCN) self-attention introduced below are used in this work to limit the algorithmic latency of self-attention. For streaming ASR with the conformer architecture, a causal Conv module is used by restricting the convolution window of the 1-D depthwise convolution to 16 past frames plus the current frame. Finally, streaming ASR with source attention is realized using TA.

3. Dual Causal/Non-Causal Self-Attention

The most widely used streaming self-attention architectures are restricted self-attention (RSA) and chunk-based self-attention (CSA). RSA uses a limited number of look-ahead frames per self-attention layer [19], which results in a growing receptive field when a series of RSA layers are used. The reason for the growing delay is that frames input to RSA may be non-causal, i.e., they were computed requiring future context, and thus delays add up for each additional layer that uses a look-ahead for computing an output. To avoid this build up, DCN processes two sequences of causal and non-causal frames in parallel as shown in Fig. 1. For the first encoder block, the input feature sequence X_0 is simply duplicated to derive a causal sequence X^{c}_0 and a non-causal sequence X^{nc}_0 of frames. Next, both sequences are processed by two parallel RSA processes, one with causal RSA using zero look-ahead frames and one with non-causal RSA using a fixed number of look-ahead frames. DCN first transforms the causal and non-causal input frames into causal and non-causal key, value, and query frames. Key and value frames are interchanged between the parallel causal and non-causal RSA processes such that for the non-causal RSA

process the look-ahead frames correspond to causal frames, and for the causal RSA process the frames that are further into the past than the look-ahead size of the non-causal frames are non-causal frames. This ensures that both RSA processes do not attend to any frames that use information beyond the attention context. Since two sequences must be forwarded to the next DCN layer, both sequences must be processed by all modules of an encoder block, which increases the computational costs. However, the computational cost is approximately equivalent to CSA with 50% chunk overlap, which also almost doubles the number of frames that are being processed.

Finally, at the output of the encoder, the non-causal encoder sequence X_E^{nc} is forwarded to the CTC and decoder branches, while the causal sequence X_E^{c} is dropped. In order to enforce a consistency between the causal and non-causal sequences, we explored inplace knowledge distillation (KD) [27, 28] between X_E^{c} (student) and X_E^{nc} (teacher) using the mean squared error (MSE) loss. In our experiments, the MSE loss is multiplied with a weight of 1.0 and added to Eq. (12), unless otherwise noted in the results section below. All model parameters are shared for processing the causal and non-causal frames in the encoder, except parameters of the normalization layers.

4. Triggered Attention

For streaming E2E ASR, we use the triggered attention (TA) technique, which exploits the alignment properties of CTC to enable frame-synchronous decoding of encoder-decoder based ASR models [17,18]. TA training uses CTC-based forced alignment to determine the temporal position of the label y_l for $l = 1, \ldots, L$ in the encoder output sequence $X_E = (\boldsymbol{x}_1^E, \ldots, \boldsymbol{x}_N^E)$ in order to condition the source attention mechanism of the decoder on previous encoder frames plus a fixed number of look-ahead frames $\varepsilon^{\mathrm{dec}}$ relative to the determined label positions n'_l. The TA loss can be written as

$$p_{\mathrm{ta}}(Y|X_E) = \prod_{l=1}^{L} p(y_l|\boldsymbol{y}_{1:l-1}, \boldsymbol{x}_{1:\nu_l}^E) \tag{20}$$

with $\nu_l = n'_l + \varepsilon^{\mathrm{dec}}$, where n'_l denotes the position of the first occurrence of label y_l in the CTC forced alignment sequence [17, 18], and $\boldsymbol{x}_{1:\nu_l}^E = (\boldsymbol{x}_1^E, \ldots, \boldsymbol{x}_{\nu_l}^E)$ corresponds to the truncated encoder sequence. The term $p(y_l|\boldsymbol{y}_{1:l-1}, \boldsymbol{x}_{1:\nu_l}^E)$ represents the posterior probability function of the TA decoder that substitutes Eq. (13) for streaming ASR and which is computed similarly to Eqs. (14) to (19) but with the restricted encoder sequence $\boldsymbol{x}_{1:\nu_l}^E$ deployed in Eq. (18).

At inference time, we use the TA decoding algorithm of [18, 19], which extends the frame-synchronous CTC prefix beam search algorithm with TA-based on-the-fly rescoring.

5. Experiments

5.1. Settings

ASR tasks used for the present experiments are the LibriSpeech corpus of read English audio books [29], the Switchboard (SWBD) corpus of conversational English telephone speech [30], and the Hong Kong University of Science and Technology (HKUST) corpus of Mandarin telephone speech [31].

Hyperparameters of the transformer and conformer models are set to $d_{\mathrm{model}} = 256$, $d_{\mathrm{ff}} = 2048$, $d_h = 4$, $E = 12$, and $D = 6$ for HKUST and SWBD, while d_{model} and d_h are increased to 512 and 8 for LibriSpeech. A symmetric depthwise convolution window of size 31 frames is used by the Conv module of the full-sequence conformer encoder. For streaming ASR, a causal (left-sided) convolution window of size 17 is used instead. The Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$, and learning rate scheduling similar to [26] with 25000 warmup steps is applied for training. The learning rate factor is set to 5.0 and the number of training epochs amounts to 50 for HKUST and to 100 for LibriSpeech as well as SWBD. Weight factor γ, which is used to balance the CTC and decoder objectives, is set to 0.3 for LibriSpeech and HKUST and to 0.2 for SWBD. Label smoothing with a penalty of 0.1 and SpecAugment are used for all experiments [32]. A task specific RNN-based language model (LM) is trained via stochastic gradient descent using the official training text data of each task [29–31] and employed via shallow fusion during decoding. For HKUST and SWBD, the RNN-LM consists of 2 LSTM layers with 650 units each. For LibriSpeech, 4 LSTM layers with 2048 units each are used. When indicated in the LibriSpeech results, a Transformer-based LM (Tr-LM) with 16 layers is used instead. ASR output labels consist of a blank token plus 5,000 (LibriSpeech) or 2,000 (SWBD) subwords obtained by the SentencePiece method [33]. 3,653 character-based output symbols are use for the Mandarin HKUST task.

In this work, TA-based models are obtained by fine-tuning a pre-trained model with a full-sequence based decoder for 10 epochs using the Adam optimizer without learning rate scheduling. *Fine-tuning* has proved to be very effective in this work, reducing the training time and improving ASR accuracy compared to TA models trained from scratch such as in [19].

For offline decoding, the LM weight/CTC weight/beam size are set to 0.6/0.4/20 (LibriSpeech), 0.3/0.5/10 (HKUST), and 0.3/0.3/20 (SWBD). For frame-synchronous TA decoding [18, 19], the CTC LM weight α_0/CTC weight λ/LM weight α/pruning width θ_1/pruning width θ_2/insertion bonus β/pruning size K/beam size P are set to 0.8/0.4/0.6/16.0/6.0/2.0/300/30 (LibriSpeech), 0.4/0.6/0.2/8.0/4.0/1.0/200/20 (HKUST), and 0.4/0.6/0.3/8.0/2.0/1.0/200/20 (SWBD).

5.2. Chunk-based self-attention

The CSA mechanism first segments the whole utterance into chunks using a hop size of 50%. In order to enable a fair comparison to RSA and DCN, CSA in this work attends to all frames of a current chunk as well as to the first half of all previous chunks, which is different to other CSA methods [13, 23]. At the last encoder layer, to avoid redundancy due to overlap, only frames of the first half of each chunk are forwarded, except for the last chunk of an utterance for which all frames are maintained. The CSA latency is controlled by the chunk size.

5.3. ASR Results

Results of offline or partially streaming ASR models are shown for HKUST, SWBD, and LibriSpeech in the top half of Tables 1 and 2, whereas fully streaming ASR results are shown in the bottom half. Transformer (Tr) and conformer (Co) results are separated by dashed lines. ASR results of the full-sequence based encoder models demonstrate that TA decoding [18, 19] achieves comparable results to offline joint CTC/attention decoding [25]. For HKUST and LibriSpeech, even slightly improved character error rates (CERs) and word error rates (WERs) can be observed with the TA decoding algorithm. Note that fine-tuning a full-sequence based decoder to a TA decoder with limited look-ahead improves ASR results considerably compared to training from scratch. For example, for an RSA-based encoder with one frame look-ahead per layer

Table 1: *CERs and WERs for the HKUST and SWBD ASR tasks.*

Encoder			Decoder		HKUST [%]		SWBD [%]	
Self-attention	Delay	Type	Type	Delay	dev	test	callhm	swbd
Full-seq.	Full	Tr	Offline	Full	20.7	21.3	17.3	8.6
Full-seq.	Full	Tr	TA	Full	20.5	21.2	17.7	9.1
Full-seq.	Full	Tr	TA	480 ms	20.3	21.0	17.7	8.8
RSA	480 ms	Tr	Offline	Full	22.9	23.2	18.5	9.2
CSA	480 ms	Tr	Offline	Full	22.0	22.2	17.3	8.7
DCN w/o KD	480 ms	Tr	Offline	Full	22.4	22.5	18.2	8.9
DCN	480 ms	Tr	Offline	Full	22.1	22.3	17.6	8.9
Full-seq.	Full	Co	Offline	Full	19.1	20.1	14.8	7.1
Full-seq.	Full	Co	TA	Full	18.8	19.9	14.9	7.3
RSA	480 ms	Co	Offline	Full	21.2	22.0	18.3	8.5
DCN	480 ms	Co	Offline	Full	20.2	21.2	16.9	8.1
RSA	480 ms	Tr	TA	480 ms	23.1	23.4	19.6	10.2
CSA	480 ms	Tr	TA	480 ms	22.2	22.4	18.4	9.3
CSA	480 ms	Tr	TA	320 ms	22.3	22.4	18.4	9.5
CSA	640 ms	Tr	TA	320 ms	21.8	21.9	18.0	9.1
DCN	480 ms	Tr	TA	480 ms	22.3	22.2	18.6	9.5
DCN	480 ms	Tr	TA	320 ms	22.3	22.3	18.7	9.8
DCN	640 ms	Tr	TA	320 ms	21.9	22.0	18.9	9.8
RSA	480 ms	Co	TA	480 ms	21.7	22.1	18.8	9.1
DCN	480 ms	Co	TA	480 ms	20.6	21.3	17.8	8.2
DCN	640 ms	Co	TA	320 ms	**20.6**	**21.1**	**17.2**	**8.1**

(480 ms delay) and a TA decoder with 12 frames look-ahead (480 ms delay) the test-clean/test-other WERs of LibriSpeech are improved from 3.1%/8.1% [19] (trained from scratch) to 2.9%/7.6% (fine-tuning, cf. Table 2). DCN and CSA outperform RSA across all experiments often by a large margin. The RSA encoder with 480 ms delay corresponds to 1 frame look-ahead for each of the 12 encoder blocks, since the RSA delay grows linearly with the number of RSA layers and the frame rate amounts to 40 ms. DCN and CSA both avoid a growing delay with the number of encoder blocks. Thus, for the 480 ms and 640 ms encoder delays, DCN and CSA process 12 and 16 look-ahead frames at each block.

The effect of using knowledge distillation (KD) for the DCN-based encoder is shown in Table 1 by comparing the "DCN w/o KD" and "DCN" results for the "offline" decoder. KD improves HKUST results by up to 0.3% and SWBD results by up to 0.6%. The best streaming E2E ASR results are highlighted in bold, which are obtained using the DCN-based conformer architecture with 640 ms (encoder) + 320 ms (decoder) delay.

5.4. Decoding Delay

Tables 1 and 2 show ASR results and algorithmic delays of the encoder and decoder neural networks. However, a model may learn to delay an output until it has seen sufficient context or the decoding algorithm may also contain mechanisms to postpone the recognition of ASR labels. Therefore, in this section, we analyze the emission delay of the TA decoding algorithm, which is measured by computing the time difference between the word-level forced alignments provided by [34, 35] and the trigger positions of the TA decoding when recognizing a complete word. The emission delays are estimated across all four test sets of LibriSpeech for all words of sentences that are recognized correctly using $delay_w = t_w^{\mathrm{pred}} - t_w^{\mathrm{grdt}}$, where w denotes a word ID, t_w^{pred} the predicted time position, and t_w^{grdt} the ground-truth time position of the word end.

Figure 2 shows the decoding delays of different E2E ASR models, via a box and whisker plot summarizing the emission delays estimated from more than 100k words for each ASR setup. Apparently, TA decoding, where timing is mainly defined by CTC, does not artificially delay ASR outputs, since recognized words are overall well aligned with the forced alignment results of a hybrid ASR system [35]. Furthermore, by comparing "Tr-DCN", "Tr-CSA", and "Tr-RSA1", which all use the same encoder and decoder delays of 480 ms, it can be noticed that DCN and CSA tend to produce less delay than RSA. We suppose the reason is that ASR models with less or no look-ahead tend to delay the emission of ASR labels, as can be seen from the RSA-based models with no look-ahead ("Tr-RSA0") and with 1 frame look-ahead per layer ("Tr-RSA1"), where the receptive field is not constant but linearly increases with the depth of the model. In addition, it is shown that conformer-based models generate larger ASR delays than transformer-based models. Our assumption is that the causal Conv module of the streaming conformer is the culprit due to a similar reason as discussed above regarding "Tr-RSA0".

Table 2: *WERs [%] for the LibriSpeech ASR task. * indicates usage of the Tr-LM for decoding.*

	Encoder			Decoder		dev		test	
	Self-attention	Delay	Type	Type	Delay	clean	other	clean	other
	Full-seq.	Full	Tr	Offline	Full	2.4	5.9	2.6	5.9
	Full-seq.	Full	Tr	TA	480 ms	2.3	5.9	2.6	6.2
	RSA	480 ms	Tr	Offline	Full	2.5	6.7	2.8	7.1
	CSA	480 ms	Tr	Offline	Full	2.4	6.6	2.6	6.7
	DCN	480 ms	Tr	Offline	Full	2.5	6.7	2.6	6.8
	Full-seq.	Full	Co	Offline	Full	2.6	5.7	2.8	6.0
	Full-seq.	Full	Co	TA	Full	2.1	5.6	2.3	5.5
	RSA	480 ms	Co	Offline	Full	3.2	6.7	3.2	7.3
	DCN	480 ms	Co	Offline	Full	2.9	6.5	3.0	6.7
	RSA	0 ms	Tr	TA	480 ms	2.9	7.9	3.1	8.0
	RSA	480 ms	Tr	TA	480 ms	2.7	7.3	2.9	7.6
	CSA	480 ms	Tr	TA	480 ms	2.6	7.1	2.7	7.3
	CSA	640 ms	Tr	TA	320 ms	2.6	6.9	2.8	7.3
	DCN	480 ms	Tr	TA	480 ms	2.7	7.1	2.8	7.4
	DCN	640 ms	Tr	TA	320 ms	2.6	6.9	2.8	7.3
	RSA	480 ms	Co	TA	480 ms	2.5	6.7	2.7	7.1
	DCN	640 ms	Co	TA	320 ms	2.4	6.6	2.6	6.7
*	DCN	640 ms	Co	TA	320 ms	**2.2**	**6.4**	**2.5**	**6.3**

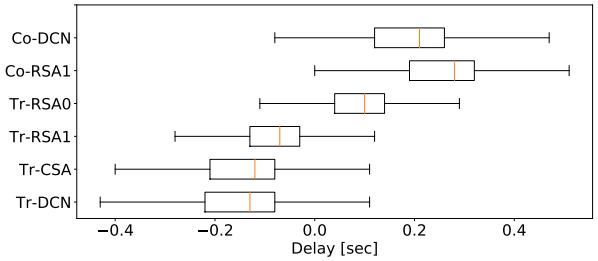

Figure 2: *Box and whisker plot of word emission delays for TA-based ASR decoding using different encoder architectures. All encoder and decoder models use the same look-ahead context of 480 ms except Tr-RSA0, where the encoder uses 0 look-ahead.*

6. Conclusions

We presented the dual causal/non-causal (DCN) self-attention architecture for streaming ASR, which performs causal and non-causal self-attention simultaneously in a frame-synchronous manner. DCN self-attention uses causal frames to prevent self-attention from using information beyond the attention context and to avoid a growing receptive field and latency for multiple consecutive layers. Combined with triggered attention (TA), the proposed streaming E2E ASR system demonstrates very strong results for various ASR tasks, e.g., achieving WERs of 2.5%/6.3% for the test-clean/test-other conditions of LibriSpeech with less than 1 second algorithmic delay. In addition, TA training by fine-tuning is shown to be effective, and the TA decoding latency is analyzed, demonstrating that ASR models with no or non-uniform look-ahead such as RSA are more prone to delaying the emission of ASR outputs.

7. References

[1] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao, and Y. Gong, "Developing RNN-T models surpassing high-performance hybrid models with customization capability," in *Proc. Interspeech*, Oct. 2020.

[2] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S. y. Chang, W. Li, R. Alvarez, Z. Chen, C. C. Chiu, D. Garcia, A. Gruenstein, K. Hu, A. Kannan, Q. Liang, I. McGraw, C. Peyser, R. Prabhavalkar, G. Pundak, D. Rybach, Y. Shangguan, Y. Sheth, T. Strohman, M. Visontai, Y. Wu, Y. Zhang, and D. Zhao, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. ICASSP*, May 2020, pp. 6059–6063.

[3] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, Jun. 2006.

[4] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:abs/1211.3711*, 2012.

[5] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, Dec. 2015.

[6] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, pp. 1240–1253, 2017.

[7] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018, pp. 4774–4778.

[8] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech*, Sep. 2017, pp. 939–943.

[9] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, Oct. 2020.

[10] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on ESPnet toolkit boosted by conformer," *arXiv preprint arXiv:2010.13956*, Oct. 2020.

[11] N. Jaitly, Q. V. Le, O. Vinyals, I. Sutskever, D. Sussillo, and S. Bengio, "An online sequence-to-sequence model using partial conditioning," in *Proc. NIPS*, Dec. 2016, pp. 5067–5075.

[12] T. N. Sainath, C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, "Improving the performance of online neural transducer models," *arXiv preprint arXiv:abs/1712.01807*, 2017.

[13] E. Tsunoo, Y. Kashiwagi, and S. Watanabe, "Streaming transformer ASR with blockwise synchronous beam search," in *Proc. SLT*, Jan. 2021, pp. 22–29.

[14] C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *Proc. ICLR*, Apr. 2018.

[15] N. Ari, C. A. Cherry, W. Macherey, C.-C. Chiu, S. Yavuz, R. Pang, W. Li, and C. Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," in *Proc. ACL*, Jul. 2019, pp. 1313–1323.

[16] C. Wang, Y. Wu, L. Lu, S. Liu, J. Li, G. Ye, and M. Zhou, "Low latency end-to-end streaming speech recognition with a scout network," in *Proc. Interspeech*, Oct. 2020, pp. 2112–2116.

[17] N. Moritz, T. Hori, and J. Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. ICASSP*, May 2019, pp. 5666–5670.

[18] N. Moritz, T. Hori, and J. Le Roux, "Streaming end-to-end speech recognition with joint CTC-attention based models," in *Proc. ASRU*, Dec. 2019, pp. 936–943.

[19] N. Moritz, T. Hori, and J. Le Roux, "Streaming automatic speech recognition with the transformer model," in *Proc. ICASSP*, May 2020, pp. 6074–6078.

[20] 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 *Proc. ICASSP*, May 2020.

[21] L. Dong, F. Wang, and B. Xu, "Self-attention aligner: A latency-control end-to-end model for ASR using self-attention network and chunk-hopping," in *Proc. ICASSP*, May 2019, pp. 5656–5660.

[22] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, "Transformer-based online CTC/attention end-to-end speech recognition architecture," in *Proc. ICASSP*, May 2020, pp. 6084–6088.

[23] C. Wu, Y. Wang, Y. Shi, C.-F. Yeh, and F. Zhang, "Streaming transformer-based acoustic models using self-attention with augmented memory," in *Proc. Interspeech*, Oct. 2020, pp. 2132–2136.

[24] S. Karita, N. Yalta, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. Interspeech*, Sep. 2019, pp. 1408–1412.

[25] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, Sep. 2018, pp. 2207–2211.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Dec. 2017, pp. 6000–6010.

[27] J. Yu and T. Huang, "Universally slimmable networks and improved training techniques," *arXiv preprint arXiv:1903.05134*, Oct. 2019.

[28] J. Yu, W. Han, A. Gulati, C.-C. Chiu, B. Li, T. N. Sainath, Y. Wu, and R. Pang, "Dual-mode ASR: Unify and improve streaming ASR with full-context modeling," *arXiv preprint arXiv:2010.06030*, Jan. 2021.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, Apr. 2015.

[30] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*, Mar. 1992, pp. 517–520.

[31] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, "HKUST/MTS: A very large scale mandarin telephone speech corpus," in *Proc. ISCSLP*, vol. 4274, 2006, pp. 724–735.

[32] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[33] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[34] L. Lugosch, "Librispeech alignments (version 1.0)," Mar 2019. [Online]. Available: http://doi.org/10.5281/zenodo.2619474

[35] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using Kaldi," in *Proc. Interspeech*, Aug. 2017, pp. 498–502.