Interspeech 2022
18-22 September 2022, Incheon, Korea

Streaming Align-Refine for Non-autoregressive Deliberation

Weiran Wang *Ke Hu* *Tara N. Sainath*

Google, Inc.

{weiranwang, huk, tsainath}@google.com

Abstract

We propose a streaming non-autoregressive (non-AR) decoding algorithm to deliberate the hypothesis alignment of a streaming RNN-T model. Our algorithm facilitates a simple greedy decoding procedure, and at the same time is capable of producing the decoding result at each frame with limited right context, thus enjoying both high efficiency and low latency. These advantages are achieved by converting the offline Align-Refine algorithm to be streaming-compatible, with a novel transformer decoder architecture that performs local self-attentions for both text and audio, and a time-aligned cross-attention at each layer. Furthermore, we perform discriminative training of our model with the minimum word error rate (MWER) criterion, which has not been done in the non-AR decoding literature. Experiments on voice search datasets and Librispeech show that with reasonable right context, our streaming model performs as well as the offline counterpart, and discriminative training leads to further WER gain when the first-pass model has small capacity.
Index Terms: streaming ASR, non-autoregressive decoding, discriminative training

1. Introduction

There has been a surge of interest in non-autoregressive (non-AR) automatic speech recognition (ASR) models that are not constrained to decode in a left-to-right manner [1, 2, 3, 4, 5, 6, 7, 8]. These models make parallel update steps during inference, i.e., each decoding step can modify multiple or all positions of previous step results simultaneously. And they are attractive due to the simplicity and efficiency of their inference procedure.

Non-AR models can be largely categorized into two classes. The first class of models iteratively refine the label sequences [2, 3, 4, 6], following the general framework of mask-predict [1]: in each refinement step, certain positions of the input label sequence are replaced by a special `[mask]` token, and the model learns to predicts all masked tokens simultaneously given the partially observed token sequence. A challenge to this approach is to estimate the length of the ground truth label sequence, for which heuristics based on CTC decoding results [3, 5] and the dynamic length prediction task [4] have been developed.

The second class of non-AR methods perform iterative refinement instead on alignments, which are sequences containing underlying tokens (including blanks for non-emission) at each frame. Representative methods in this class are Imputer [7] and Align-Refine [8, 9]. The former gradually reveals positions of a fully masked alignment in a fixed number of steps, and its training procedure suffers from exposure bias. The latter updates complete alignment in each refinement step, effectively allowing complex edits to the label sequences.

Previous work [9] has proposed a practical use case of Align-Refine for deliberation [10, 11]. The authors use a small causal RNN-T [12, 13], which runs fast and has reasonable accuracy, to generate the initial hypothesis alignment, as opposed to using CTC [14] as first-pass in the original algorithm of [8]. The authors then apply Align-Refine to the initial alignment for a few steps to generate new hypotheses of improved accuracy.

We note however, an important aspect of ASR—latency—has not been addressed by existing alignment-based non-AR methods. In this work, we develop a streaming version of Align-Refine, which is capable of producing results through greedy decoding as data comes in, with controllable delay at each frame; this allows us to extend the use of Align-Refine into application scenarios with more stringent requirements on latency. To achieve this goal, we propose a novel transformer decoder architecture that performs local self-attentions for both text and audio separately, and a time-aligned cross-attention at each layer; this architecture incorporates audio right context without incurring unnecessary model delay. While there have been use of other non-AR methods in the streaming mode [6, 15], the inference procedures of these methods are not as simple as ours.

Furthermore, given our clean formulation and inference procedure, we perform discriminative training of streaming Align-Refine, with the minimum word error rate (MWER) criterion [16]. We found discriminative training to provide further WER gain when the first-pass model has small capacity. To the best of our knowledge, this is the first time sequence training is used in the non-AR decoding literature. Experimental results on voice search datasets and Librispeech show that with reasonable right context, streaming Align-Refine performs similarly well as the offline counterpart, and compares favorably against existing non-AR methods and deliberation methods.

2. Streaming Non-AR deliberation

2.1. Review of offline Align-Refine

We generally follow the framework of [9] to apply Align-Refine [8] for deliberation. We use a streaming RNN-T [12, 13] to generate first-pass hypothesis by autoregressive beam search, which achieves both good accuracy (by modeling label dependency) and low latency (without using right context). Beam search returns the *alignment* of the most probable hypothesis for each input utterance. The hypothesis alignment is a sequence of discrete tokens corresponding to the inferred labels of each frame; each label can be `<blank>` to indicate non-emission.

The first-pass hypothesis alignment is fed to the Align-Refine decoder for S steps of iterative refinement. Each refinement step takes in an initial alignment and outputs a complete updated alignment; all steps share the same model parameters. Similarly to the decoder module in attention-based model [17], the Align-Refine decoder consists of a series of transformer layers to integrate the text-side information from alignment and the audio-side information from encoder output. A schematic diagram of the transformer architecture of [9] is provided in Figure 1. On the text side ("Alignment self-atten"), each transformer layer performs a self-attention for the text features, and use the result as query to perform cross-attention on audio features, whose result is in turn used as text-side features for the

Copyright © 2022 ISCA

10.21437/Interspeech.2022-10715

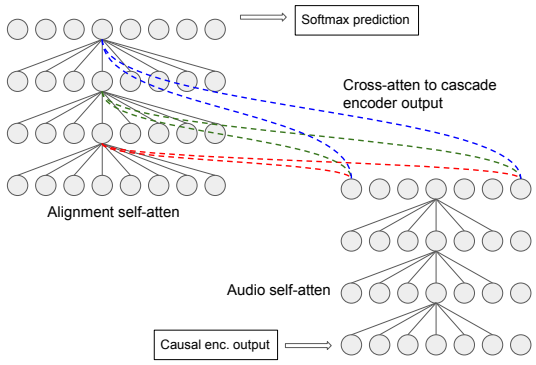

Figure 1: *Transformer architecture of offline Align-Refine [9].*

next layer. A final softmax layer is used on top of the last transformer layer output to predict labels and <blank>'s. On the audio side ("Audio self-atten"), the authors propose to use additional non-causal cascaded encoder on top of the causal encoder of first-pass RNN-T, to extract audio features of rich right context; this architectural design indeed significantly boost the deliberation accuracy. Since the model of [9] works in the offline mode, the alignment self-attentions use full context, and each transformer layer attends to the cascaded encoder output with full context. All layers are trained jointly using the CTC loss [14], which marginalizes alignments between CTC predictions and ground truth label sequences.

The CTC greedy alignment, obtained by picking the most probably token for each frame in parallel, is used as the output alignment and fed to the next refinement step. The greedy alignment after all S steps is collapsed (by removing repetitions and then <blank>'s) into a label sequence as the final decoding result. The overall maximum likelihood estimation (MLE) loss is the averaged CTC losses at all steps: for input utterance X with label sequence y, we minimize $\ell^S_{\text{MLE}}(X, y) := -\frac{1}{S}\sum_{i=1}^{S} \log P^i_{ctc}(y|X)$ where $\log P^i_{ctc}(y|X)$ is the full-sum conditional log-probability under the CTC model at step i.

2.2. Streaming Align-Refine

We make architectural changes to convert Align-Refine to be streaming compatible. First, we ensure the alignment self-attention is "local", so that the attention context vector of each frame only depends on a local window around itself, i.e., representation of each token is computed by attending to a number of previous tokens and a small number of future tokens. The total amount of right context accumulates with the depth of the architecture and becomes its model delay, e.g., if we have $L = 6$ layers with per-layer right context $C = 2$, the model waits for $L \times C = 12$ future frames to output for the current frame.

Second, we ensure the cross-attention between alignment audio features is "local", so that the context vector of alignment frame (acting as the "query") only depends on a window of audio feature frames (acting as the "key" and "value"). Note that owing to the RNN-T alignment topology [12], the alignment sequence and audio sequence are of different lengths. As an example, assume the input utterance has transcription "hello world" which is decomposed into 3 wordpieces {_hello, _wor, ld}, and assume the encoder output consists of 5 audio frames. Then a plausible RNN-T alignment is

<b> _hello <b> <b> <b> _wor ld <b>

where <b> denotes <blank>. Each <blank> token advances the audio frame index by 1 whereas each non-blank token indicates label emission without advancing in time. By counting the number of previous blanks, we determine the "time", or equivalently the audio frame index, when each token is emitted. For the above example, tokens are output at frame indices (0-based)

0 1 1 2 3 4 4 4

where _wor and ld are both emitted at audio frame 4 as allowed by RNN-T. We use these timestamp information to construct "local" cross-attention, so that a token emitted at audio frame index t attends to a window of audio frames around t.

Third, we propose a novel decoder architecture, shown in Figure 2, to utilize additional audio self-attention like cascaded encoder but without incurring further delay. For each layer, besides the alignment self-attention and time-aligned cross-attention, we compute in parallel an audio self-attention with the same amount of right context in "time" (audio frame index). While the cross-attention output is passed to the next layer as alignment features, the audio self-attention output is passed to the next layer as audio features. In such a way, attention operations are synchronized according to the audio time, and delays from alignment side and audio side do not add up. Specifically, in the schematic diagram of Figure 2, we have $L = 3$ self-attention operations for both alignments and audio, but the effective depth of the architecture $L + 1 = 4$ for determining model delay,[1] versus $2L$ if they are stacked as in Figure 1.

2.3. Discriminative training

It is known in the literature that, after initial training of an ASR model with the MLE criterion, finetuning it with a loss closer to the final evaluation criterion (i.e., WER) often yields further accuracy gain [18, 19, 16]. In this work, we perform discriminative training of Align-Refine using the minimum word error rate (MWER) criterion. For an input utterance X, we forward the Align-Refine model to perform S' refinement steps, with S' potentially different from the S used in initial MLE training. And at the end of S' refinement steps, we perform CTC beam search instead of greedy decoding, to output $K > 1$ hypotheses denoted as $\hat{y}_1, \ldots, \hat{y}_K$. We then compute the log-probabilities of the hypotheses under our model, defined as

$$\log P(\hat{y}_k|X) = \frac{1}{S'}\sum_{i=1}^{S'} \log P^i_{ctc}(\hat{y}_k|X), \quad k = 1, \ldots, K.$$

We make the approximation that probability over possible label sequences concentrate in the top-K space, to compute

$$P_k = \frac{P(\hat{y}_k|X)}{\sum_{y'} P(y'|X)} \approx \frac{P(\hat{y}_k|X)}{\sum_{k=1}^{K} P(\hat{y}_k|X)},$$

and subsequently the MWER loss

$$\ell^{S'}_{\text{MWER}}(X, y, \{\hat{y}_1, \cdots, \hat{y}_K\}) := \sum_{k=1}^{K} P_k \cdot \text{NWE}(\hat{y}_i, y)$$

where $\text{NWE}(\hat{y}_i, y)$ measures the number of word errors between hypothesis $\hat{y}_i$ and ground truth y. We use superscript S' to signify that hypotheses are obtained after S' refinement steps. As is common in the literature, for better learning stability, we minimize a composite loss

$$\ell^{S'}_{\text{MWER}}(X, y, \{\hat{y}_1, \cdots, \hat{y}_K\}) + \gamma \cdot \ell^{S'}_{\text{MLE}}(X, y)$$

over the training set in the discriminative training phase, and we fix $\gamma = 0.005$ in this work. Note that while we use beam search in generating multiple hypotheses for MWER training, we still perform greedy decoding for final inference.

[1] We could remove audio self-attention at the bottom to make the effective depth L instead of $L + 1$, though we have not in this paper.

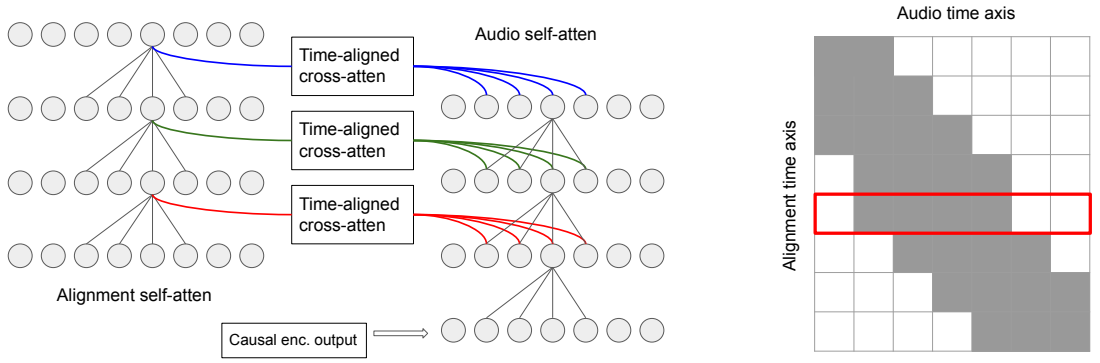

Figure 2: *Left: Transformer architecture of streaming Align-Refine where audio self-attention is synchronized with alignment self-attention. Right: The attention mask with shape* 8×7 *shared by all time-aligned cross-attention operations in the left figure. Black indicates non-zero attention weights. Highlighted row indicates that frame 4 of the alignment feature sequence attends to a window around frame 3 of the audio feature sequence, with a left context of 2 frames and a right context of 1 frame.*

3. Experiments on voice search datasets

In this section, we follow the experimental setup of [9] on voice search datasets so that our results are directly comparable. The training set includes anonymized and hand-transcribed multi-domain audio data [20] and has gone through multi-condition training (MTR [21]) and random 8kHz down-sampling [22] as augmentation. We use a development set of 12K anonymized and hand-transcribed utterances that are representative of Google's Voice Search traffic, with an average duration of 5.5 seconds; this set is denoted by VS. Final evaluation is performed on five test sets. The first set is the side-by-side losses test (SXS), containing 1K utterances where the quality of the E2E model transcription has more errors than a state-of-the-art conventional model [23]. The rest four are TTS generated test sets containing rare proper nouns (RPN) which appear less than 5 times in the training set. These sets cover the Maps, News, Play, and QSession domains and are denoted RPN-M, RPN-N, RPN-P, and RPN-Q respectively, each containing 10K utterances.

3.1. Cascaded RNN-T as first-pass model

The first-pass model for generating the initial hypothesis is a cascaded model designed for on-device usage [24], consisting of two conformer encoders and a embedding-based decoder. The first encoder is causal and is used by the decoder to generate partial results in streaming mode, while the second encoder sits on top of the causal encoder with a right-context of 0.9s and is used for generating more accurate final hypotheses. Note that [9] used the causal pass of this model, with a total of 56M weight parameters, to generate initial hypotheses. While we mostly focus on the same setup as it better simulates the streaming usage of Align-Refine, we also provide results for deliberating the non-causal pass which has 155M weight parameters. We freeze the first-pass model for deliberation training. The MLE training phase takes 800K steps, with a global batch size of 4096 utterances. In the MWER training phase, batch size is reduced to 2048 and we evaluate models at 25K steps.

3.2. Streaming Align-Refine for causal RNN-T

We use a model architecture similar to that of [9] for Align-Refine: the transformer decoder consists of $L = 6$ transformer layers with attention dimension 640 and 8 attention heads, and we perform $S = 3$ refinement steps during MLE training. We achieve different amount of model delay by varying the number of right-context frames C at each layer, and thus the model delay per refinement step is $L \times C \times f$ where f is the decoder frame size ($60ms$ in this section) without audio self-attention, and $(L + 1) \times C \times f$ with audio self-attention.

Table 1: *VS WERs (%) of Align-Refine for different amounts of model delays, with up to three refinement steps during inference. The first pass model is causal RNN-T of 56M weight parameters, and has a WER of 7.8% on VS.*

per-step model delay (secs)	Inference refinement step		
	1	2	3
Alignment self-attention only			
0.00	7.6	7.5	7.5
0.36	7.3	7.2	7.1
0.72	6.9	6.6	6.6
1.80	6.5	6.2	6.1
+ Audio self-attention			
0.00	7.5	7.3	7.3
0.42	6.6	6.4	6.3
0.84	6.3	6.0	6.0
2.10	6.1	5.7	5.7
+ Continue with discriminative training			
0.00	7.3	7.1	7.1
0.42	6.5	6.2	6.2
0.84	6.2	5.9	5.9
2.10	5.9	5.5	5.5

We set the beam size to 4 for the first-pass causal RNN-T model during both training and inference. We provide the WERs of Align-Refine using only alignment self-attention (with time-aligned cross-attention performed on causal encoder outputs) in the top section of Table 1. While Align-Refine barely improves over the first pass when not using any right context (i.e., 0.0 sec of model delay), the accuracy improves sharply as the amount of right context increases. We then include audio self-attention in the model architecture, which leads to larger model size but not much increase in delay, and results of these models are given in the middle section of Table 1. Observe that audio self-attention consistently reduces WERs for all model delays, implying that right context in the audio modality is complementary to right context in the text modality.

We then finetune the models with audio self-attention, us-

Table 2: *Test WERs (%) on voice search datasets for causal first-pass (top section) and non-causal first-pass (bottom section).*

Method	Additional weights	Total model delay (secs)	WERs (%)					
			VS	SXS	RPNM	RPNN	RPNP	RPNQ
Causal RNN-T	0	0.00	7.8	37.5	16.6	11.4	40.9	25.6
Attention seq2seq Delib. [25]	48M	∞	6.0	34.3	**13.8**	10.2	**36.2**	22.2
Offline Align-Refine [9]	55M	∞	5.7	32.0	14.6	10.0	38.3	23.5
Streaming Align-Refine	70M	0.00	7.1	35.0	15.8	11.6	39.7	24.3
		0.84	6.2	32.6	15.0	10.5	38.5	22.7
		1.68	5.9	31.1	14.5	10.2	38.3	22.3
		4.20	**5.5**	**30.1**	14.1	**9.9**	37.4	**21.8**
Non-causal RNN-T	99M	0.90	5.2	27.6	12.9	9.0	37.8	20.2
Attention seq2seq Delib. [25]	149M	∞	**4.9**	**25.2**	**12.4**	**7.4**	**34.1**	**18.8**
Stream Align-Refine	169M	5.10	**4.9**	27.3	12.8	8.8	37.2	19.9

ing the discriminative training procedure discussed in Sec 2.3. Empirically, we observe no benefit by using more than $S' = 1$ refinement steps in ℓ_{MWER}, and we stick to this configuration here. Results of discriminatively trained models are presented in the bottom section of Table 1. These models consistently outperform their MLE-trained counterparts, across all delays.

As noted by [9] for offline Align-Refine, significant WER gains are achieved in the first two refinement steps during inference. Interestingly, we observe a new type of trade-off here: it may be as accurate to run a model of smaller delay for two steps, than to run a model of larger delay for a single step. For example, in the bottom section of Table 1, running the discriminatively trained model with 0.42s per-step delay for 2 steps leads to a WER of 6.2%, the same as running the model with 0.84s per-step delay for 1 step. Similarly, running the model with 0.84s per-step delay for 2 steps leads to a WER of 5.9%, which is the same as running the model with 2.1s per-step delay, but the former as a smaller total delay of 1.68s.

3.3. Align-Refine for non-causal RNN-T

Given the large WER improvements for deliberating a small first-pass, a natural question is whether our method is still useful when the initial hypothesis is generated by a stronger model. To answer this question, we apply streaming Align-Refine with 2.1s per-step delay to the non-causal RNN-T of 155M parameters (causal encoder + non-causal encoder + RNN-T decoder) and 0.9s model delay. We feed the causal encoder output to Align-Refine, which has its own audio self-attention. This model improves the first-pass WER of 5.2% to 5.0% in one step, and to 4.9% in another step, without discriminative training (which did not help further).

3.4. Comparisons with other methods

We provide test sets WERs of our models with audio self-attention in Table 2, along with comparisons with a few models. We take the best streaming from Sec 3.2 and Sec 3.3 and evaluate them with 2 refinement steps. For each method, we provide the amount of additional weight parameters on top of the 56M causal RNN-T, as well as the total model delay. In the case of causal first-pass (Table 2 top section), we compare with the best offline model from [9], as well as an attention-based seq2seq deliberation model similar to that of [25]. Observe that with sufficient right context (between 1.68s to 4.2s delay), streaming Align-Refine performs as well as the offline counterpart and outperforms prior methods on most test sets with 4.2s delay.

For the non-causal first-pass (Table 2 bottom section), we compare again with [25], where the attention-based decoder performs beam search with beam size 8. Streaming Align-Refine performs similarly to this model on VS (mostly containing frequent words) with a simpler inference procedure, while attention-based deliberation is more accurate on rare words.

Table 3: *Librispeech WERs (%) of non-AR methods.*

Method	dev_clean	dev_other	test_clean	test_other
First-pass	3.6	9.5	4.0	9.1
Streaming Align-Refine per-step delay and WERs @step 1/2				
0.84s	3.5/3.4	9.1/8.9	3.7/3.6	8.8/8.6
2.10s	3.2/3.1	8.6/8.3	3.4/3.3	8.5/8.2
4.20s	3.1/3.0	8.4/8.0	3.3/3.2	8.3/8.0
21.0s	**2.9/2.8**	**8.0/7.7**	**3.2/3.0**	**7.8/7.6**
Imputer [7]			4.0	11.1
Offline CTC + Align-Refine [8]			3.6	9.0

4. Experiments on Librispeech

We perform experiments on Librispeech [26] to compare with existing non-AR methods. The first-pass is a 122M causal RNN-T with conformer encoder and embedding decoder, and uses a beam size of 8 for inference. The token set contains 1024 wordpieces. We use the streaming Align-Refine architecture found on voice search (70M parameters, including audio self-attention) without further tuning. We have trained four models of different per-step model delays: 0.84s, 2.1s, 4.2s, and 21.0s, with the last setup simulating offline Align-Refine. We do not find discriminative training to be helpful in this setup, and report WERs from MLE training. We compare our results with those of offline non-AR methods [7, 8], as shown in Table 3. Note that our first-pass RNN-T already has WERs similar to the final WERs of prior work. Streaming Align-Refine consistently improves over the strong first-pass, and its accuracy steadily increases with model delay. We measure the inference speed of methods with a single Intel Xeon CPU (@2.20GHz). The real-time-factor (RFT) of the first-pass is 0.123, while the RTF of streaming Align-Refine is 0.045 per refinement step.

5. Conclusions

We have proposed a streaming non-autoregressive decoding method for second-pass deliberation. Our method improves WERs of both small and large streaming RNN-T models with controllable model delay, and benefits from discriminative training when the first-pass has small capacity. As a future direction, we would like to incorporate large amount of unpaired data into our model training [5, 27], to better capture label dependency and improve on rare word recognition.

6. References

[1] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Empirical Methods in Natural Language Processing*, 2019.

[2] N. Chen, S. Watanabe, J. Villalba, and N. Dehak, "Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition," *arXiv preprint arXiv:1911.04908*, 2019.

[3] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, "Mask CTC: Non-autoregressive end-to-end ASR with CTC and mask predict," *arXiv preprint arXiv:2005.08700*, 2020.

[4] Y. Higuchi, H. Inaguma, S. Watanabe, T. Ogawa, and T. Kobayashi, "Improved Mask-CTC for non-autoregressive end-to-end ASR," in *ICASSP*, 2021.

[5] K. Deng, Z. Yang, S. Watanabe, Y. Higuchi, G. Cheng, and P. Zhang, "Improving non-autoregressive end-to-end speech recognition with pre-trained acoustic and language models," in *ICASSP*, 2022.

[6] T. Wang, Y. Fujita, X. Chang, and S. Watanabe, "Streaming end-to-end ASR based on blockwise non-autoregressive models," *Interspeech*, 2021.

[7] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," in *International Conference on Machine Learning*, 2020.

[8] E. Chi, J. Salazar, and K. Kirchhoff, "Align-refine: Non-autoregressive speech recognition via iterative realignment," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2020.

[9] W. Wang, K. Hu, and T. Sainath, "Deliberation of streaming RNN-Transducer by non-autoregressive decoding," in *ICASSP*, 2022.

[10] Y. Xia, F. Tian, L. Wu, J. Lin, T. Qin, N. Yu, and T.-Y. Liu, "Deliberation networks: Sequence generation beyond one-pass decoding," in *Advances in Neural Information Processing Systems*, 2017.

[11] K. Hu, T. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *ICASSP*, 2020.

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

[13] Y. He, T. Sainath, R. Prabhavalkar, I. McGraw, and *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine learning*, 2006.

[15] Y. Fujita, T. Wang, S. Watanabe, and M. Omachi, "Toward streaming ASR with non-autoregressive insertion-based model," in *Interspeech*, 2021.

[16] R. Prabhavalkar, T. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *ICASSP*, 2018.

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

[18] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013.

[19] M. Shannon, "Optimizing expected word error rate via sampling for speech recognition," *arXiv preprint arXiv:1706.02776*, 2017.

[20] A. Narayanan, R. Prabhavalkar, C.-C. Chiu, D. Rybach, T. Sainath, and T. Strohman, "Recognizing long-form speech using streaming end-to-end models," in *ASRU*, 2019.

[21] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *Interspeech*, 2017.

[22] J. Li, D. Yu, J.-T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *IEEE Spoken Language Technology Workshop (SLT)*, 2012.

[23] G. Pundak and T. Sainath, "Lower frame rate neural network acoustic models," in *Interspeech*, 2016.

[24] T. Sainath, Y. He, A. Narayanan, R. Botros, W. Wang, D. Qiu, C. cheng Chiu, R. Prabhavalkar, A. Gruenstein, A. Gulati, B. Li, D. Rybach, E. Guzman, I. McGraw, J. Qin, K. Choromanski, Q. Liang, R. David, R. Pang, S. Chang, T. Strohman, W. R. Huang, W. Han, Y. Wu, and Y. Zhang, "Improving the latency and quality of cascaded encoders," in *ICASSP*, 2022.

[25] K. Hu, R. Pang, T. Sainath, and T. Strohman, "Transformer based deliberation for two-pass speech recognition," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021.

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

[27] K. Hu, T. Sainath, Y. He, R. Prabhavalkar, T. Strohman, S. Mavandadi, and W. Wang, "Improving deliberation by text-only and semi-supervised training," *Interspeech*, 2022.