Streaming and Non-Autoregressive End-to-End Models for Speech Recognition

Zhengkun Tian
zhengkun.tian@nlpr.ia.ac.cn
Institute of Automation, Chinese Academy of Sciences

Contents

1. Background

End-to-end Model Speech Recognition

◎ CTC（DeepSpeech）

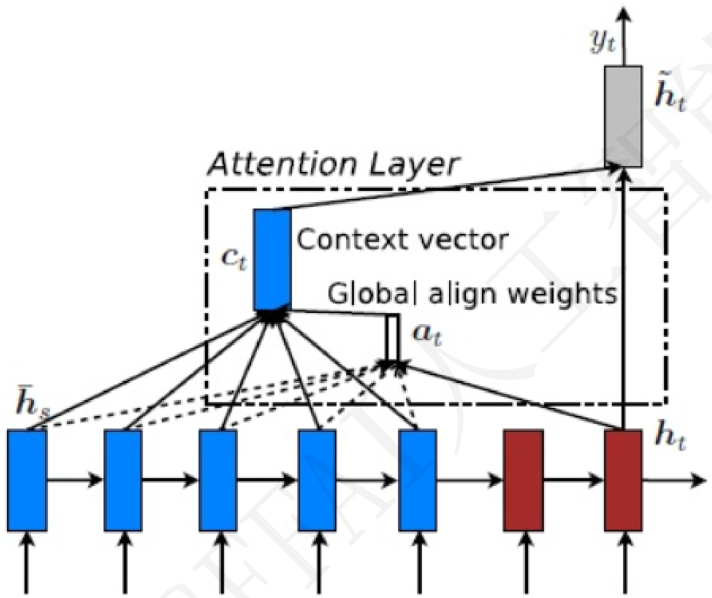
◎ Transducer (RNN-T/SA-T)

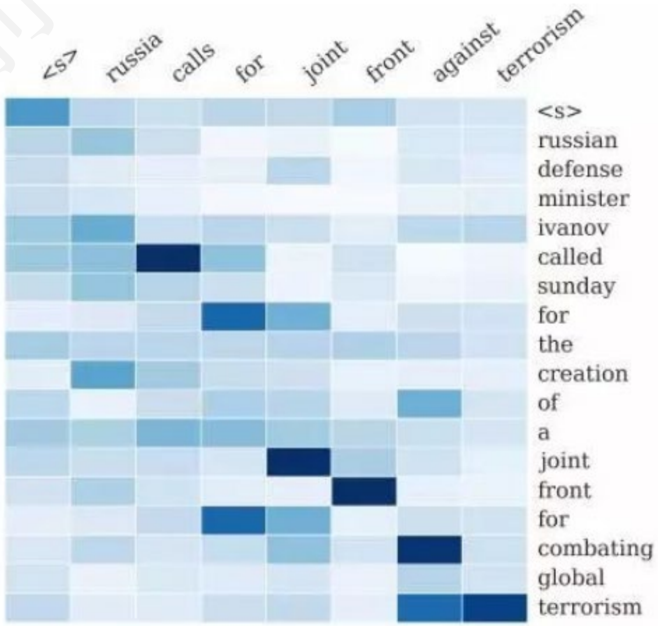
◎ **Attention-based Model**

2. Streaming ASR

The Weakness of Attention

Most of attention mechanisms needs the whole sequence as input to compute attention weights.





2.1. Monotonic Attention

- Location Attention
- Monotonic Chunk-wise Attention (MoChA)

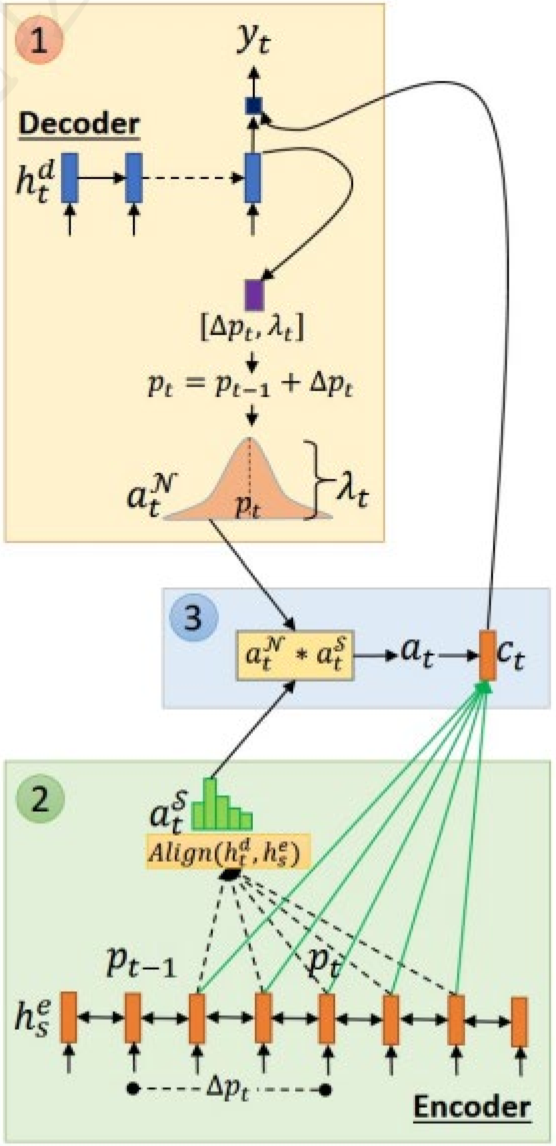
◎ Location Attention

- Constrained position prediction

$$\Delta p_t = C_{max} * \mathrm{sigmoid}(V_p^\top \tanh(W_p h_t^d))$$

- Unconstrained position prediction

$$\Delta p_t = \exp(V_p^\top \tanh(W_p h_t^d))$$



◎ Location Attention

- Scale variable

$$\lambda_t = \exp(V_\lambda^\top \tanh(W_p h_t^d))$$

- Scaled Gaussian Distribution:

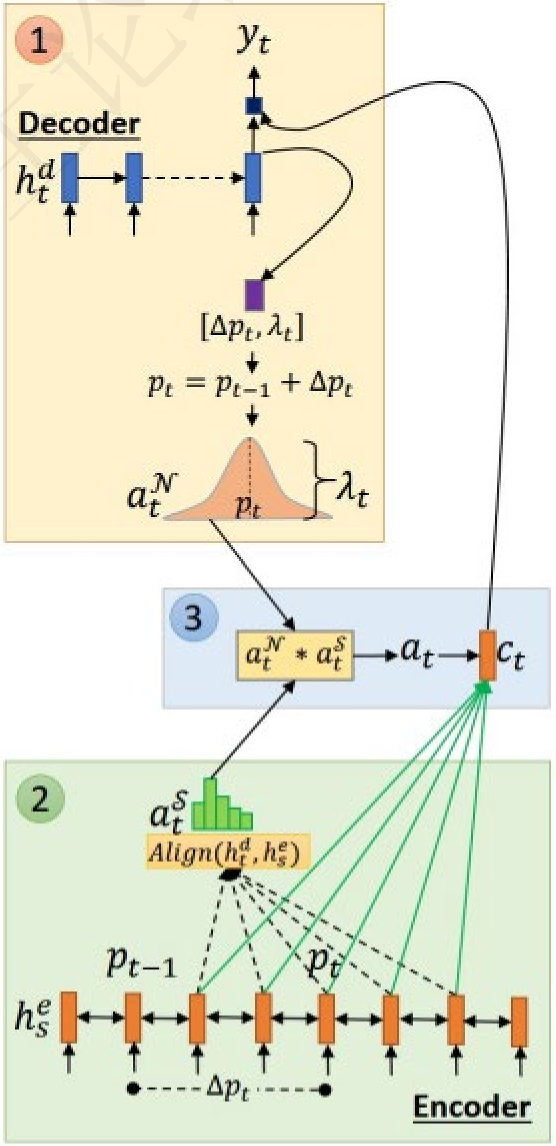
$$a_t^{\mathcal{N}}(s) = \lambda_t * \exp\left(-\frac{(s-p_t)^2}{2\sigma^2}\right).$$

- Locality-based Alignment Generation

$$a_t^{\mathcal{S}}(s) = \mathrm{Align}(h_s^e, h_t^d),\ \forall s \in [p_t - 2\sigma, p_t + 2\sigma].$$

- Context Calculation

$$c_t = \sum_{s=(p_t-2\sigma)}^{(p_t+2\sigma)} \left(a_t^{\mathcal{N}}(s) * a_t^{\mathcal{S}}(s)\right) * h_s^e$$



◎ Location Attention

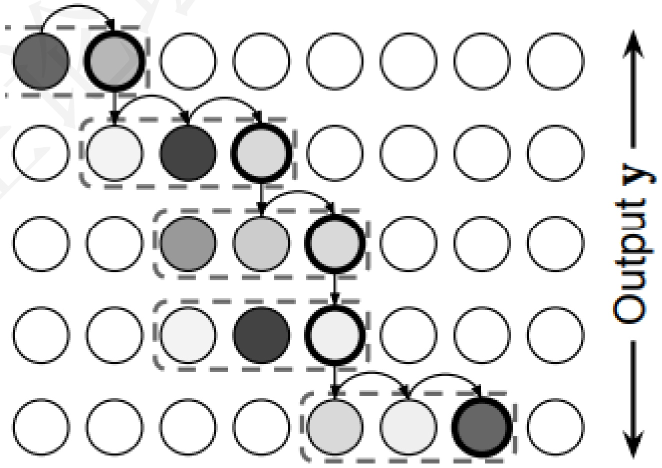
Table 1: Results from baseline and proposed models on ASR task with TIMIT test set.

Model			Test PER (%)
Global Attention Model (Baseline)			
Att Enc-Dec (pretrained with HMM align) (Chorowski et al., 2014)			18.6
Att Enc-Dec (Pereyra et al., 2017)			23.2
Att Enc-Dec (Luo et al., 2016)			24.5
Att Enc-Dec with MLP Scorer (ours)			23.8
Att Enc-Dec with *local-m* (ours) (Luong et al., 2015)			-
Local Attention Model (Proposed)			
Monotonicity	**Locality**		
Pos Prediction Δp_t	**Alignment Score**(h_s^e, h_t^d)	**Func. Type**	**Test PER (%)**
Const (*sigmoid*)	No	-	23.2
Const (*sigmoid*)	Yes	Bilinear	21.9
Const (*sigmoid*)	Yes	MLP	21.7
Unconst (*exp*)	No	-	23.1
Unconst (*exp*)	Yes	Bilinear	**20.9**
Unconst (*exp*)	Yes	MLP	21.4

Table 2: Results from baseline and proposed method on G2P task with CMUDict test set

Model	PER (%)	WER (%)
Baseline		
Enc-Dec LSTM (2 lyr) (Yao and Zweig, 2015)	7.63	28.61
Bi-LSTM (3 lyr) (Yao and Zweig, 2015)	5.45	23.55
Att Enc-Dec with Global MLP Scorer (ours)	5.96	25.55
Att Enc-Dec with *local-m* (ours) (Luong et al., 2015)	5.64	24.32
Proposed		
Att Enc-Dec + Unconst (exp) ($2\sigma = 2$)	5.45	**23.15**
Att Enc-Dec + Unconst (exp) ($2\sigma = 3$)	**5.43**	23.19

◎ MoChA



- Compute attention energy

$$e_{i,j} = \mathrm{MonotonicEnergy}(s_{i-1}, h_j)$$

- Compute probability of sampling

$$p_{i,j} = \sigma(e_{i,j})$$

- Compute chunkwise softmax energies over a size-w chunk

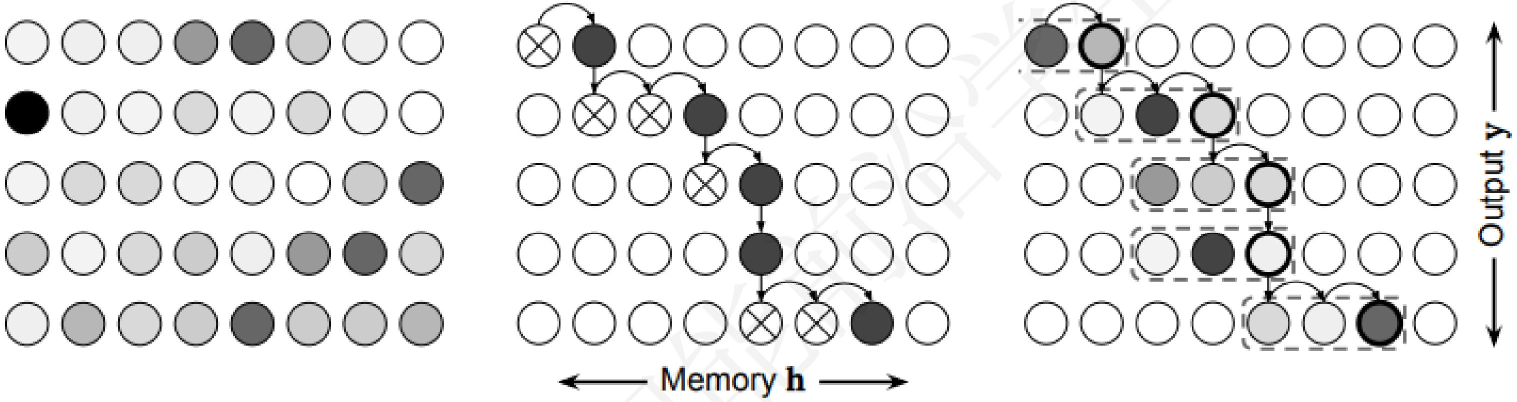
$$v = t_i - w + 1$$
$$u_{i,k} = \mathrm{ChunkEnergy}(s_{i-1}, h_k), k \in \{v, v+1, \ldots, t_i\}$$

- Compute softmax-weighted average over the chunk

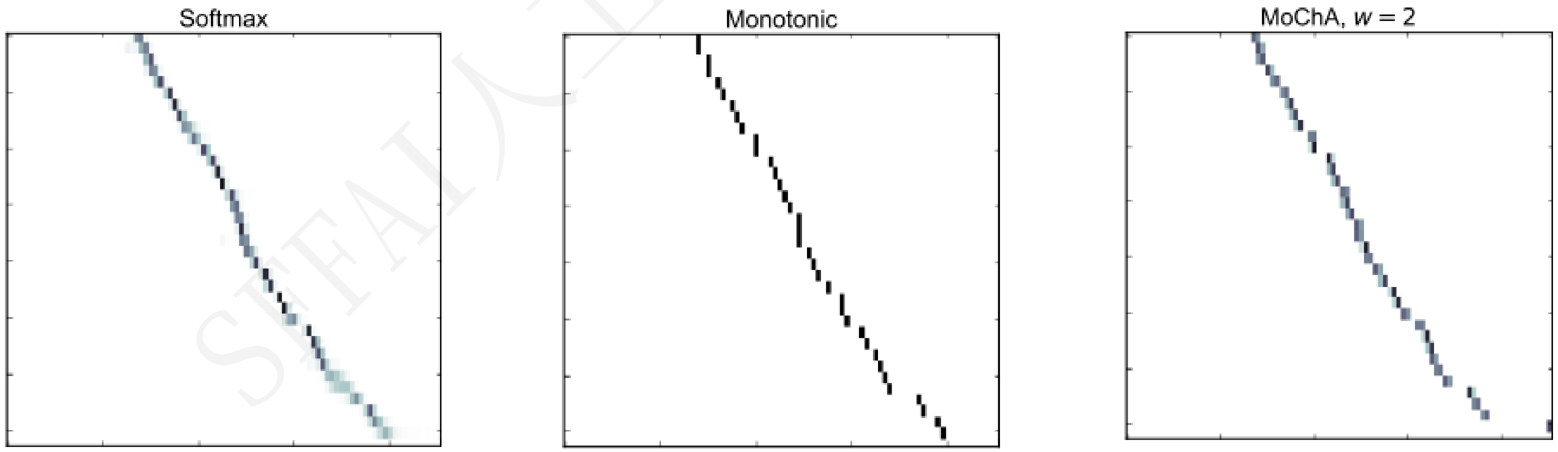
$$c_i = \sum_{k=v}^{t_i} \frac{\exp(u_{i,k})}{\sum_{l=v}^{t_i} \exp(u_{i,l})} h_k$$

◎ MoChA

- Compare different attention mechanisms



(a) Soft attention. (b) Hard monotonic attention. (c) Monotonic chunkwise attention.



◎ MoChA

Prior Result		WER
(Raffel et al., 2017) (CTC baseline)		33.4%
(Luo et al., 2016) (Reinforcement Learning)		27.0%
(Wang et al., 2016) (CTC)		22.7%
(Raffel et al., 2017) (Monotonic Attention)		17.4%
Attention Mechanism	**Best WER**	**Average WER**
Soft Attention (offline)	14.2%	$14.6 \pm 0.3\%$
MoChA, $w = 2$	13.9%	$15.0 \pm 0.6\%$

Table 1: Word error rate on the Wall Street Journal test set. Our results (bottom) reflect the statistics of 8 trials.

2.2 Accumulation of Information

Dynamically decide how many frames should be processed to predict a linguistic output.

◎ Adaptive Computation Steps

◎ Continuous Integrate-and-Fire

◎ Adaptive Computation Steps

Encoder-Decoder

$$\boldsymbol{h}_j^i = Recurrency(\boldsymbol{h}_{j-1}^i, [\boldsymbol{h}_{2j-1}^{i-1}, \boldsymbol{h}_{2j}^{i-1}])$$

$$\mathrm{p}(\mathrm{y}_i) = softmax(Recurrency(\boldsymbol{s}_{i-1}, y_{i-1}, \boldsymbol{c}_i))$$

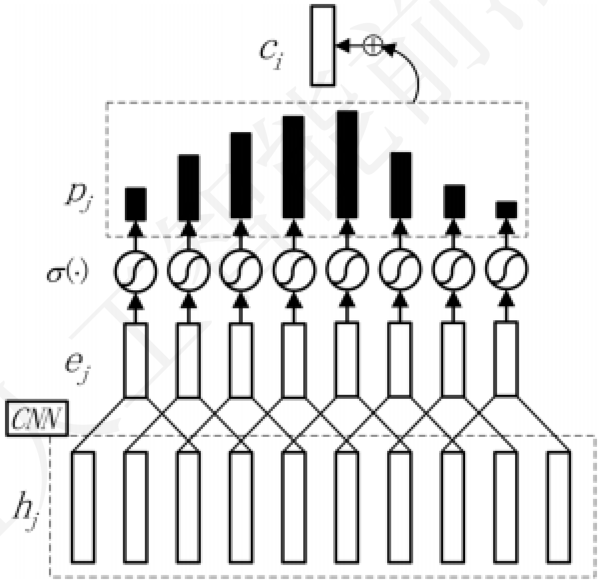


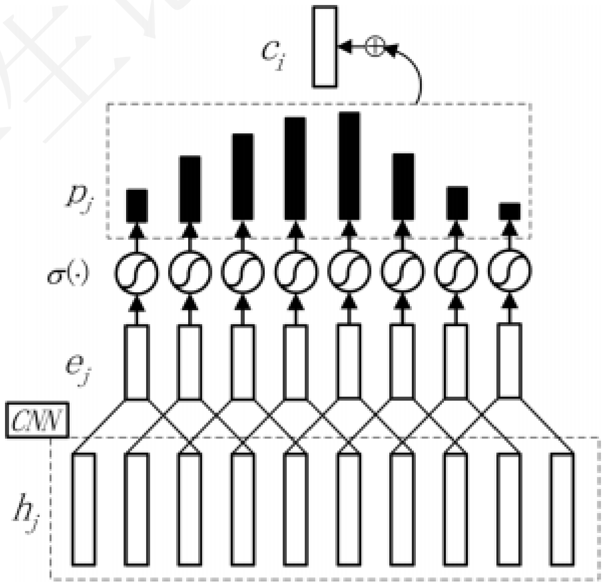
Fig. 2. The workflow of ACS algorithm. The halting probability is calculated out of the representations by the halting layer, which consists of a 1-D CNN layer and a sigmoidal unit.

◎ Adaptive Computation Steps

Compute energy vectors

$$\boldsymbol{e}_j = \text{Convolution1d}(\tilde{\boldsymbol{h}}_j)$$

Compute halting probability



$$a_j = \sigma(\boldsymbol{e}_j)$$

$$N_i = \min\left\{n: \sum_{j=1}^{n} a_j \geq 1 - \epsilon\right\} \qquad R_j = 1 - \sum_{j=1}^{N_i - 1} a_j$$

$$p_j = \begin{cases} R_i \ if \ j = N_i \\ a_j \ otherwise \end{cases} \qquad \boldsymbol{c}_i = \sum_{j=1}^{N_i} p_j \boldsymbol{h}_j$$

$$\mathrm{p}(y_i) = softmax(Recurrency(\boldsymbol{s}_{i-1}, y_{i-1}, \boldsymbol{c}_i))$$

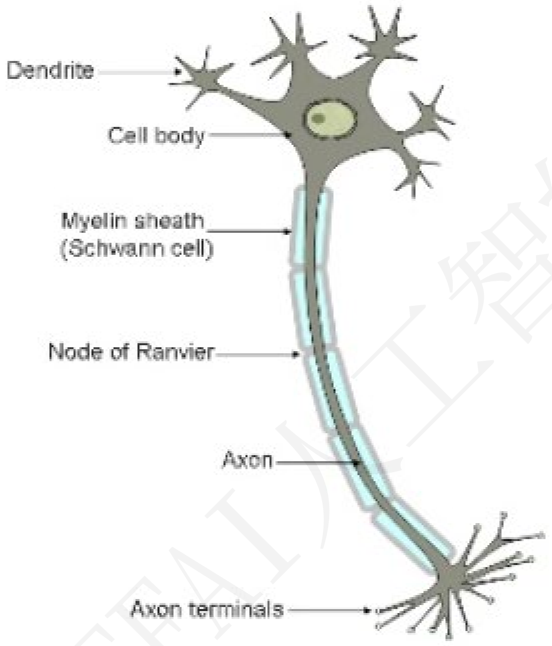
◎ Adaptive Computation Steps

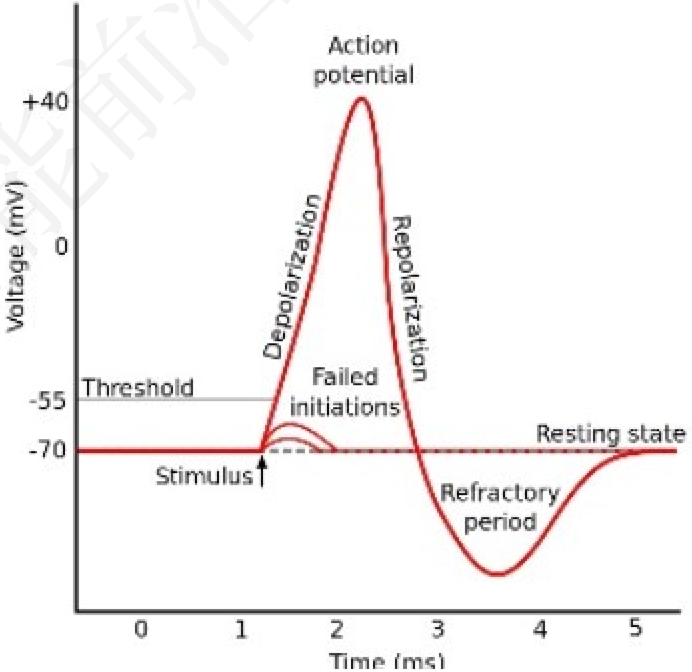
Table 1. Character Error Rate (CER) on HMM-DNN and end-to-end models. The results of attention-based and ACS models were decoded using beam search algorithm with the width of 8.

Model	CER
HMM-Hybrid Models	
HMM-DNN [19]	8.5%
Online Character Models	
Attention	34.9%
Attention + RNN-LM	32.4%
ACS	35.2%
ACS + Bidirectional Contexts (*w*=1)	33.5%
ACS + Bidirectional Contexts (*w*=1) + RNN-LM	**31.2%**
Offline Character Models	
Attention	23.2%
Attention + RNN-LM	22.0%
ACS	21.6%
ACS + Bidirectional Contexts (*w*=1)	19.8%
ACS + Bidirectional Contexts (*w*=1) + RNN-LM	**18.7%**

◎ Continuous Integrate-and-Fire

- Integrate-And-Fire





◎ Continuous Integrate-and-Fire

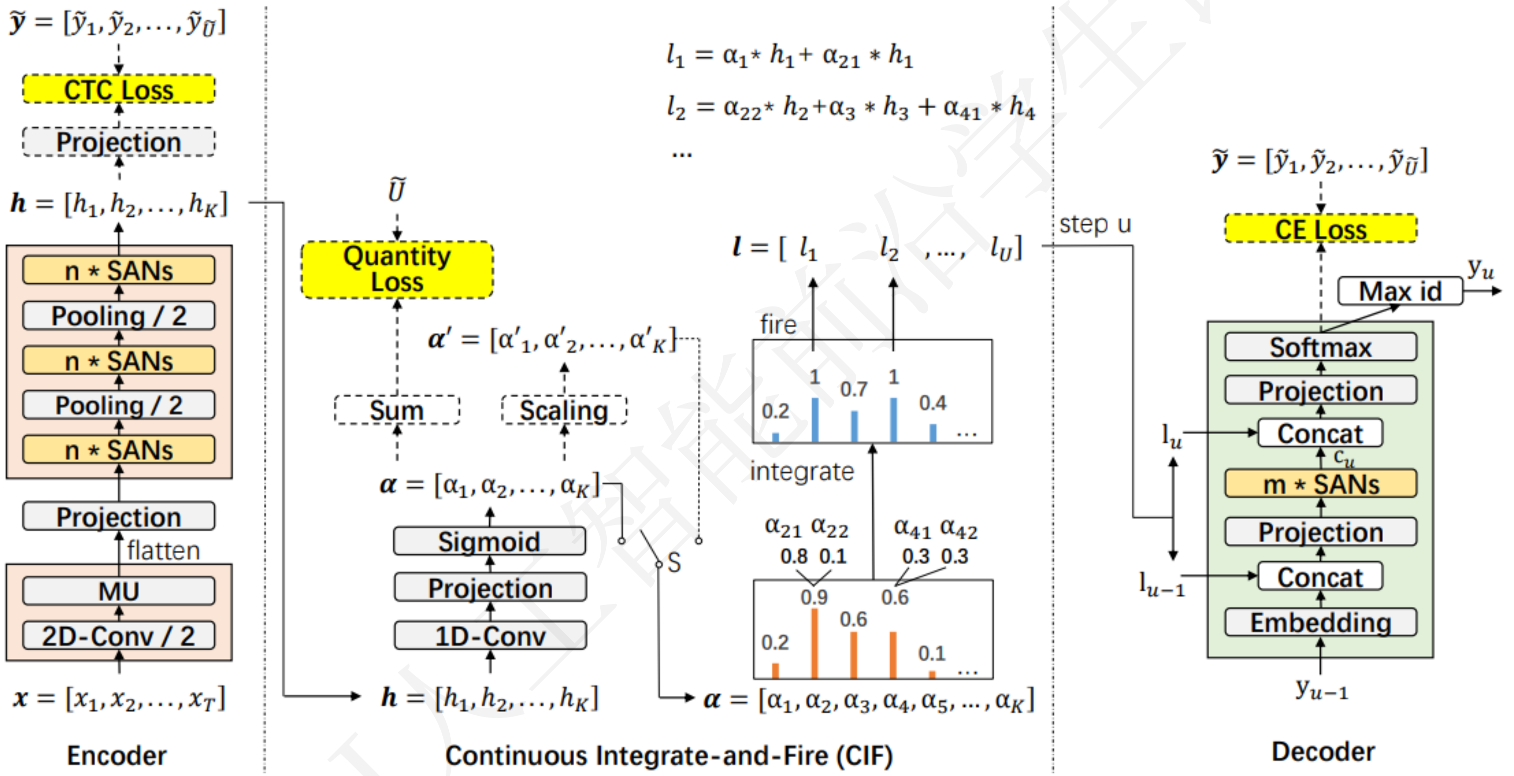


Figure 2: The architecture of our CIF-based encoder-decoder model. Operations in the dashed rectangles are only applied in the training stage, and the switch (S) in the CIF part connects the right in the training stage and the left in the inference stage.

CIF: Continuous Integrate-and-Fire for End-to-End Speech Recognition

◎ Continuous Integrate-and-Fire

- Scaling

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_K)$$

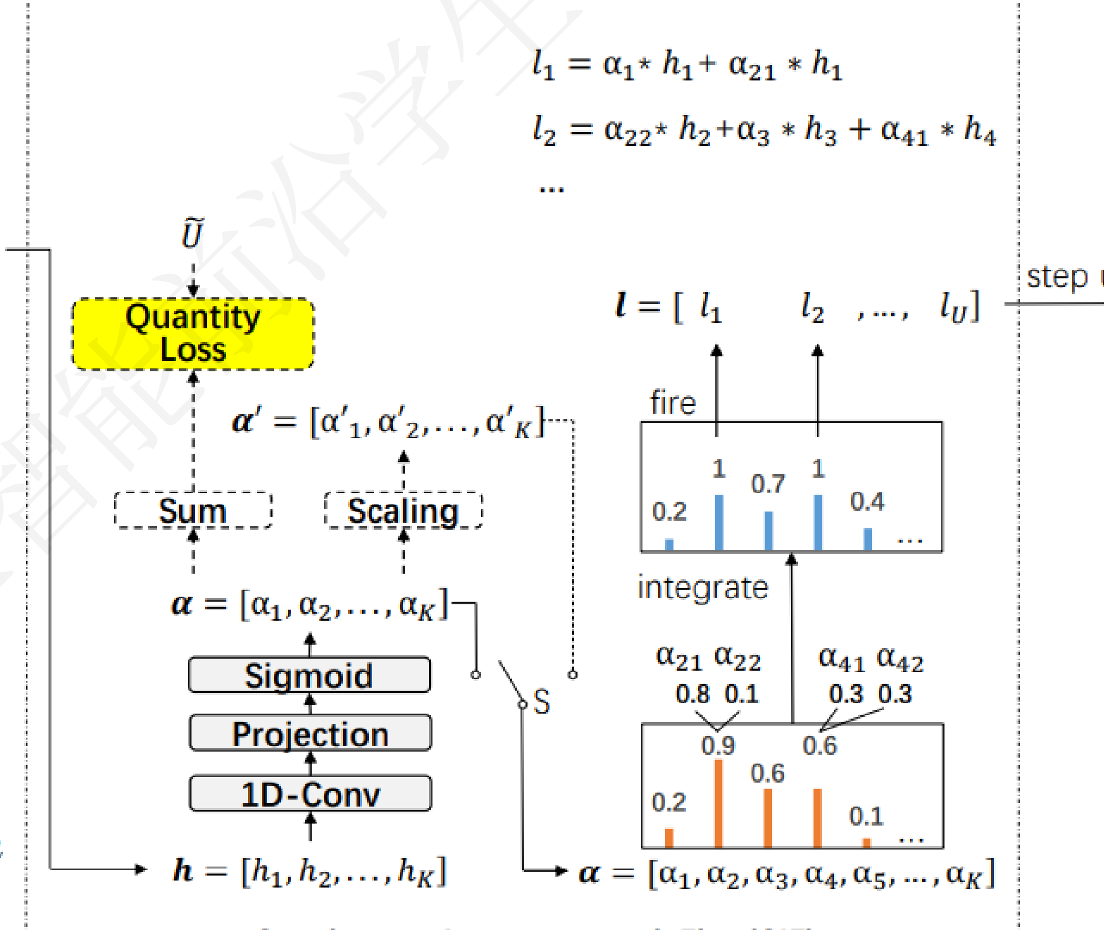
$$\Big\downarrow \frac{\tilde{U}}{\sum_{k=1}^{K} \alpha_k}$$

$$\boldsymbol{\alpha}' = (\alpha_1', \alpha_2', ..., \alpha_K')$$

- Loss

$$\mathcal{L}_{Qua} = \left|\sum_{k=1}^{K} \alpha_k - \tilde{U}\right|$$

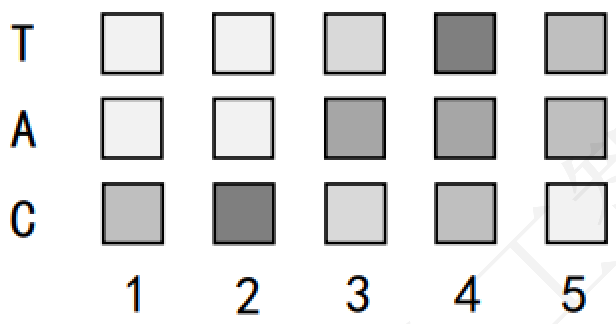
$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{CTC} + \lambda_2 \mathcal{L}_{Qu}$$



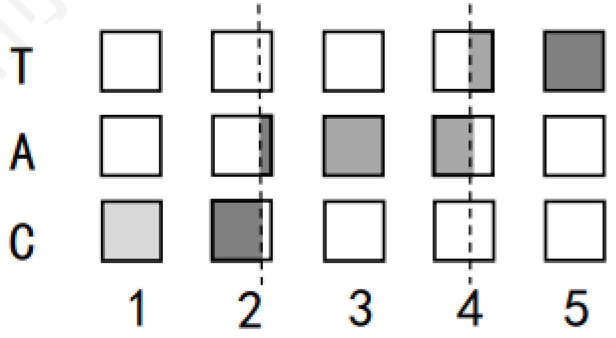
Continuous Integrate-and-Fire (CIF)

◎ Continuous Integrate-and-Fire

- Compare CIF with attention mechanism



(b) Attention



(c) CIF

◎ Continuous Integrate-and-Fire

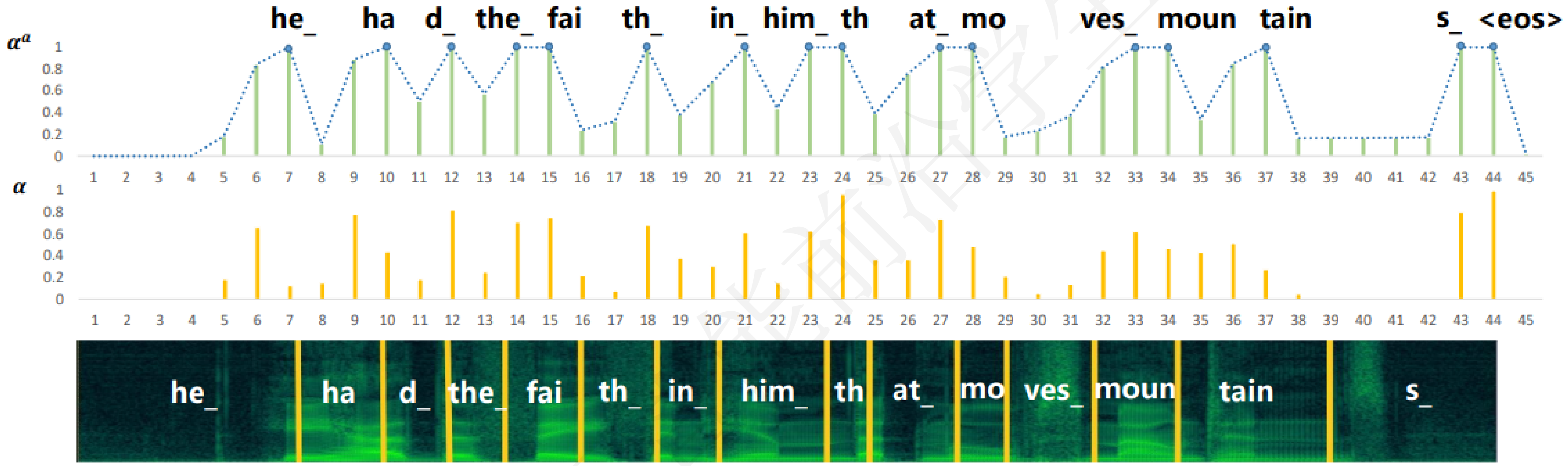


Figure 3: Token boundary positioning by CIF on an English utterance in Librispeech test-clean where "_" represents the space. The boundary in the spectrogram is marked by two humans. The middle part shows the calculated weights α for each encoded representations, and the upper part shows the accumulated weights α^a at different steps. When α^a reaches the threshold, a firing happens and a token boundary is located. We find the located boundaries are roughly accurate and the token with more stable and clear pronunciations are more prone to be located ahead of time by CIF.

◎ Continuous Integrate-and-Fire

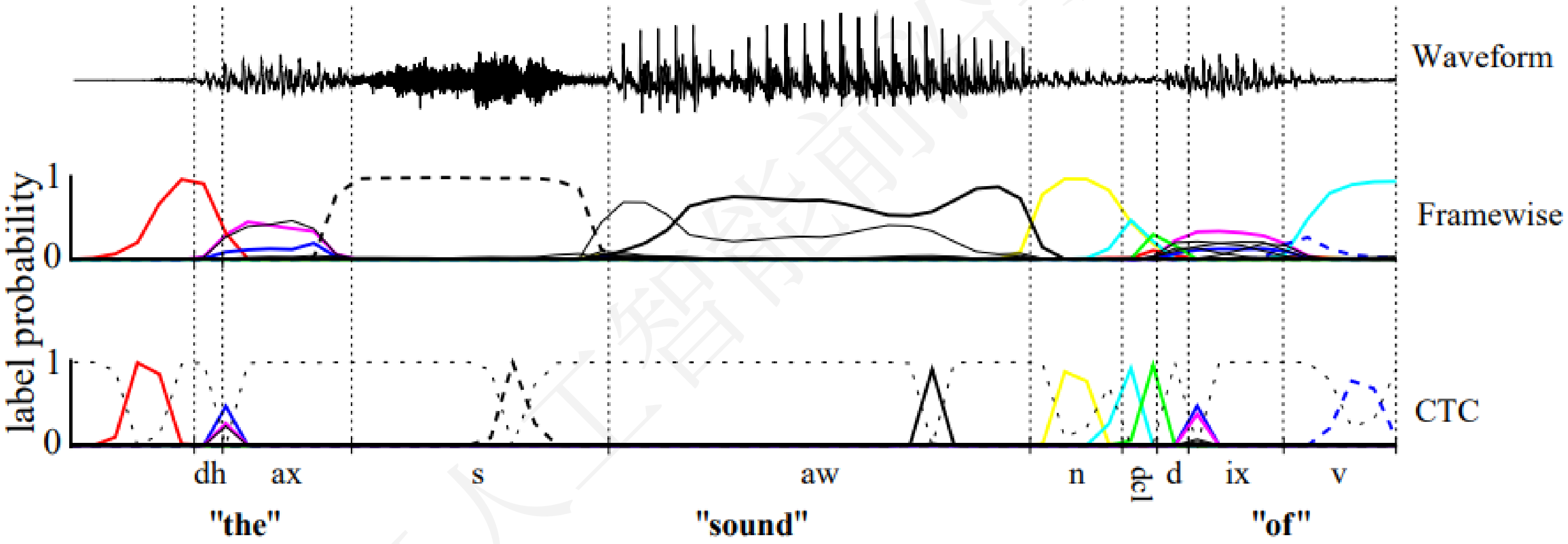
Table 1: Comparison with other published models on the AISHELL-2, CER (%)

Model	End-to-End	test_android	test_ios	test_mic
Chain-TDNN [33]	No	9.59	8.81	10.87
CIF-based model	Yes	7.25 ± 0.06	6.69 ± 0.02	7.47 ± 0.06

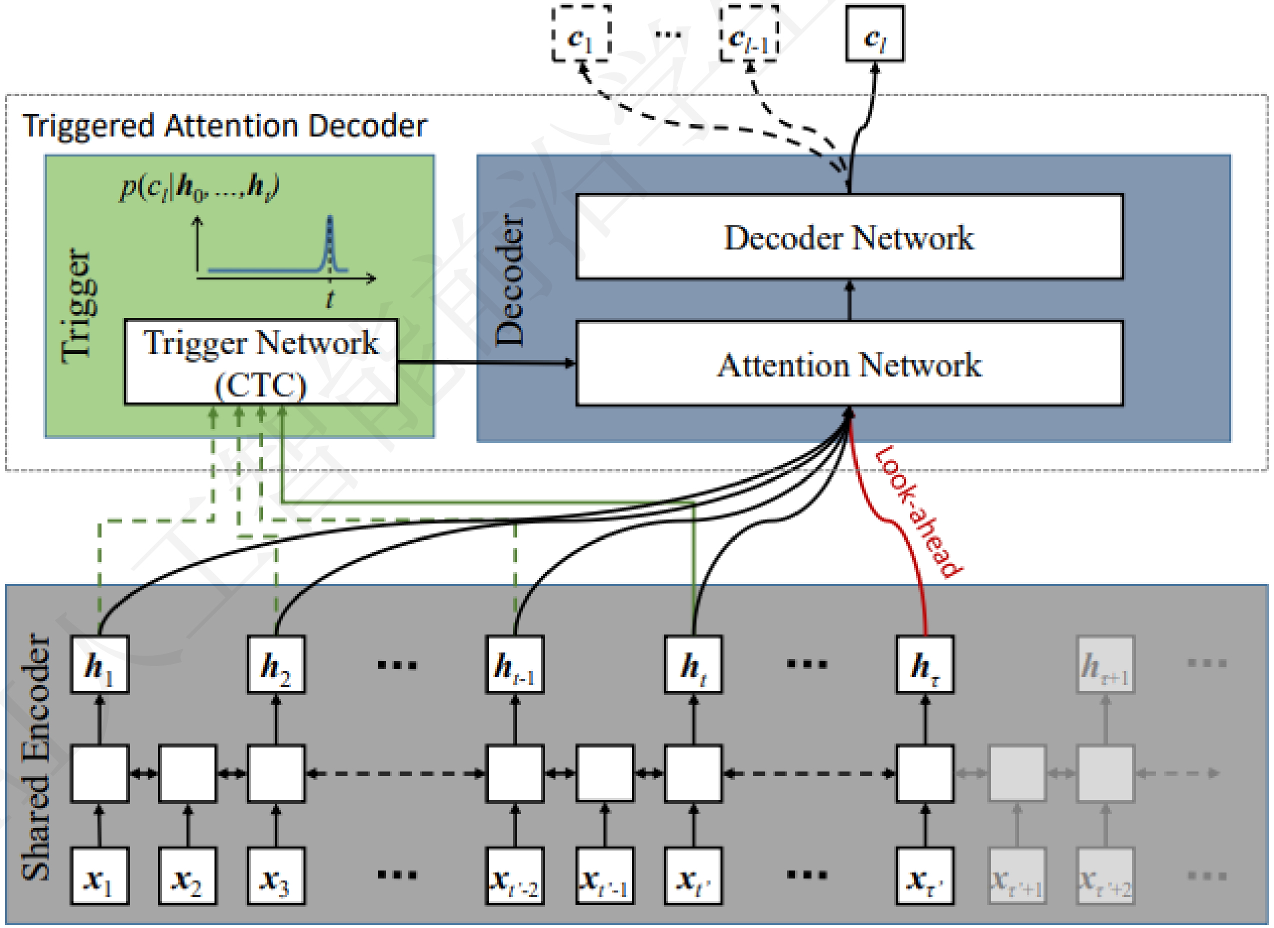
Table 2: Comparison with other end-to-end models on the Librispeech dataset, WER (%)

		text-clean		text-other	
Model	**Params**	**w/o LM**	**w/ LM**	**w/o LM**	**w/ LM**
LAS + SpecAugment [34]	-	2.8	2.5	6.8	5.8
Jasper [35]	333 M	3.86	2.95	11.95	8.79
wav2letter++ [36]	-	-	3.44	-	11.24
LAS + Deep bLSTM [37]	150 M	4.87	3.82	15.39	12.76
ASG + Gated ConvNet [38]	208 M	6.7	4.8	20.8	14.5
CTC + policy learning [39]	75 M	-	5.42	-	14.70
CTC + i-SRU 1D-Conv [40]	36 M	-	5.73	-	15.96
'Soft' and 'monotonic':					
ACS [15]	67M	16.72 ± 0.07	16.11 ± 0.03	24.09 ± 0.25	22.66 ± 0.30
Triggered Attention [14]	-	7.4	5.7	19.2	16.1
CIF-based model	67M	4.48 ± 0.09	3.70 ± 0.10	12.62 ± 0.09	10.90 ± 0.16

2.3. Triggered Attention



◎ Triggered attention system architecture



◎ Triggered attention system architecture

Details:

t =	1	2	3	4	5	6	7	8	9
Z =	<b>	<b>	d	d	<b>	o	g	g	<b>
$p(Z\|H)$ =	0.9	0.7	0.4	0.7	0.7	0.8	0.9	0.6	0.5
Z' =	<b>	<b>	d	<b>	<b>	o	g	<b>	<b>
			↑			↑	↑		

Fig. 2. Conversion of the CTC sequence Z into the trigger sequence Z', using an example with the word "dog". The red dashed boxes and the arrows indicate the frame position of a trigger event.

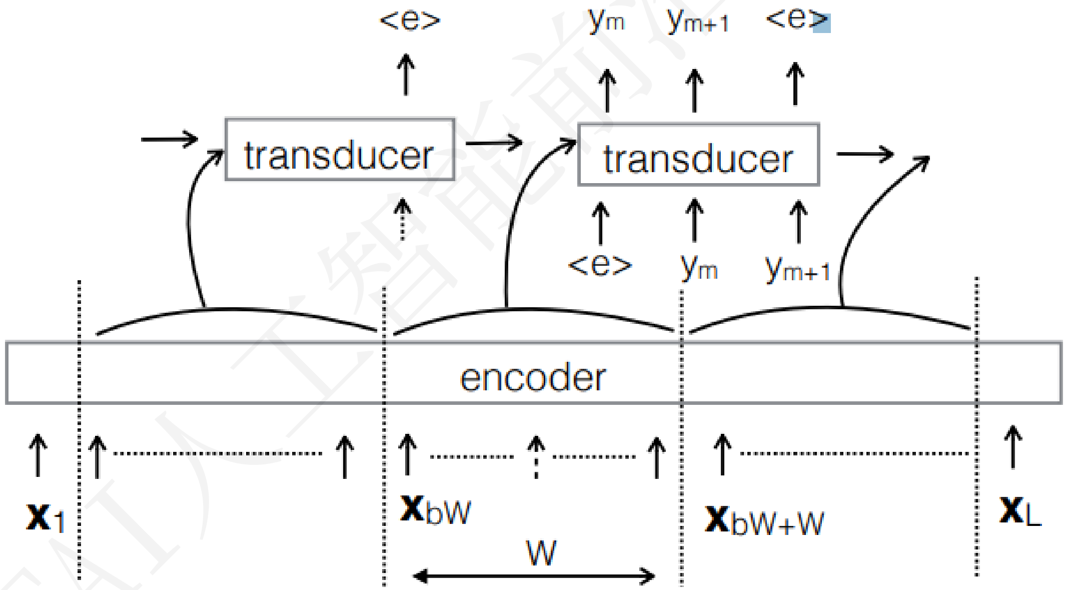
$$a_{lt} = \begin{cases} \text{DotProductAttention}(\boldsymbol{q}_{l-1}, \boldsymbol{h}_t) \\ \text{ContentAttention}(\boldsymbol{q}_{l-1}, \boldsymbol{h}_t) \\ \text{LocationAttention}(\{a_{l-1}\}_{t=1}^{T_l}, \boldsymbol{q}_{l-1}, \boldsymbol{h}_t) \end{cases}$$

Table 4. Character error rates (CER) and word error rates (WER) of the LibriSpeech ASR task.

System Settings		CER [%]		WER [%]	
Model	LM	dev [clean/other]	test [clean/other]	dev [clean/other]	test [clean/other]
Attention(dot)	✗	14.8 / 29.3	16.0 / 30.5	12.4 / 25.2	13.9 / 26.3
Attention(cont)	✗	9.6 / 24.4	8.8 / 23.8	7.4 / 21.3	7.5 / 20.6
Attention(loc)	✗	**7.1** / 22.1	**7.3** / 23.0	**5.8** / 19.2	**6.1** / 20.0
TA(dot, $\varepsilon = 2$)	✗	10.3 / 23.2	10.2 / 23.9	9.2 / 21.0	9.3 / 21.6
TA(cont, $\varepsilon = 2$)	✗	8.2 / **20.3**	8.1 / **21.3**	7.4 / **18.4**	7.4 / **19.2**
TA(loc, $\varepsilon = 20$)	✗	8.0 / 20.7	8.1 / 22.0	7.3 / 19.1	7.4 / 20.0
Attention(dot)	✓	12.6 / 28.3	14.7 / 29.6	10.1 / 22.9	12.5 / 24.3
Attention(cont)	✓	9.8 / 21.8	9.0 / 21.0	7.4 / 18.0	7.8 / 17.0
Attention(loc)	✓	**6.6** / 19.2	**6.7** / 20.0	**5.3** / **15.4**	**5.4** / **16.1**
TA(dot, $\varepsilon = 2$)	✓	9.2 / 21.3	9.1 / 22.5	7.8 / 18.7	8.0 / 19.8
TA(cont, $\varepsilon = 2$)	✓	6.9 / **18.3**	**6.7** / **19.3**	5.8 / 15.8	5.7 / 16.7
TA(loc, $\varepsilon = 20$)	✓	7.1 / 19.1	7.2 / 20.5	6.2 / 17.0	6.3 / 18.3

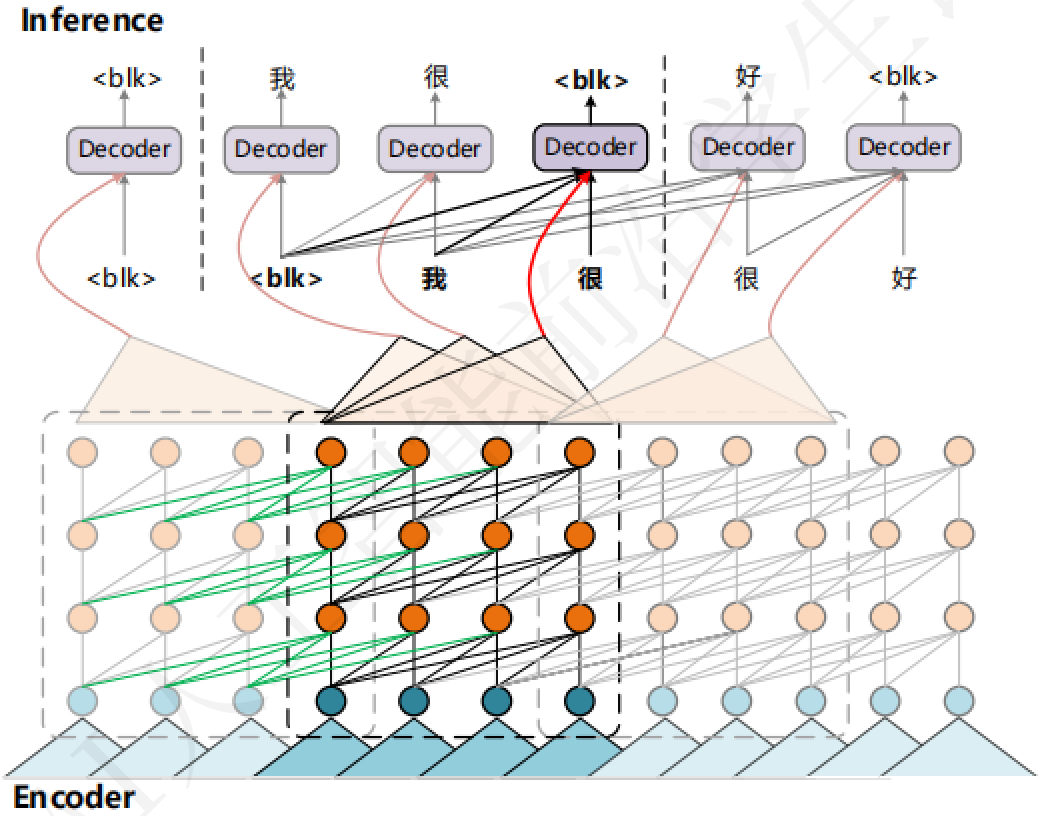
2.4 Chunk-Wise

◎ Neural Transducer



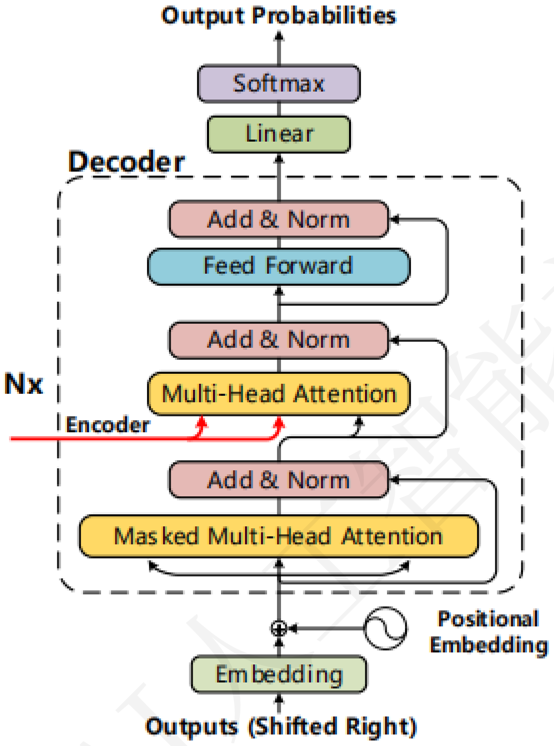
(b) Neural Transducer

◎ Synchronous Transformer

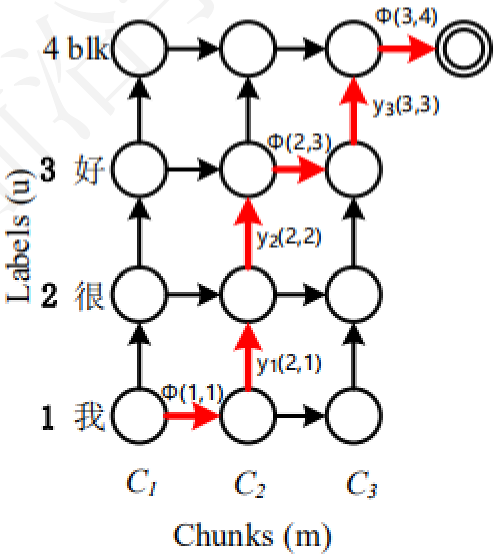


(a) The Structure of Synchronous Transformer and Inference Process

◎ Synchronous Transformer



(b) The Structure of Decoder



(c) Output Probability Lattice

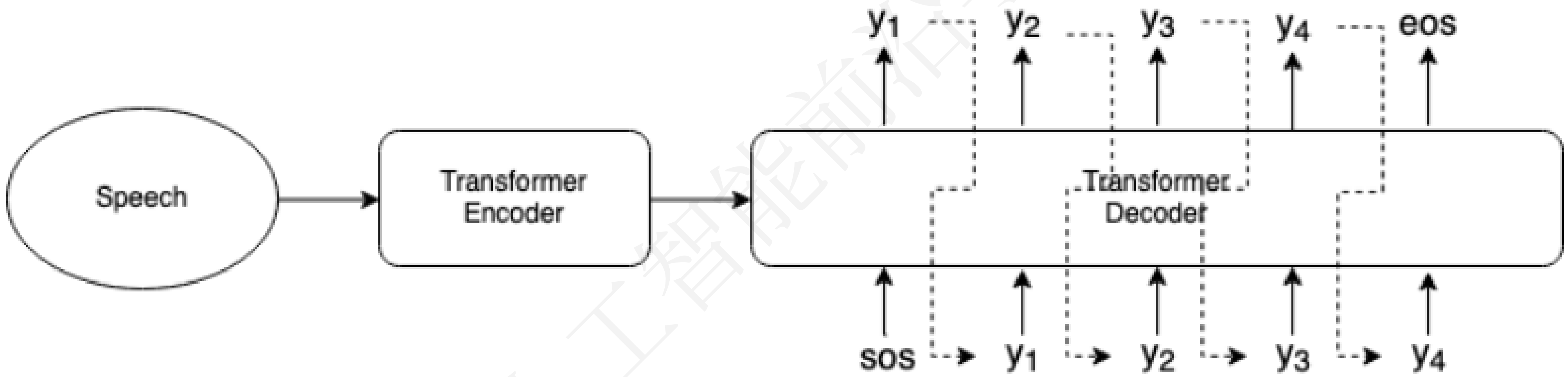
◎ Synchronous Transformer

Table 3. Comparisons with other models (CER %).

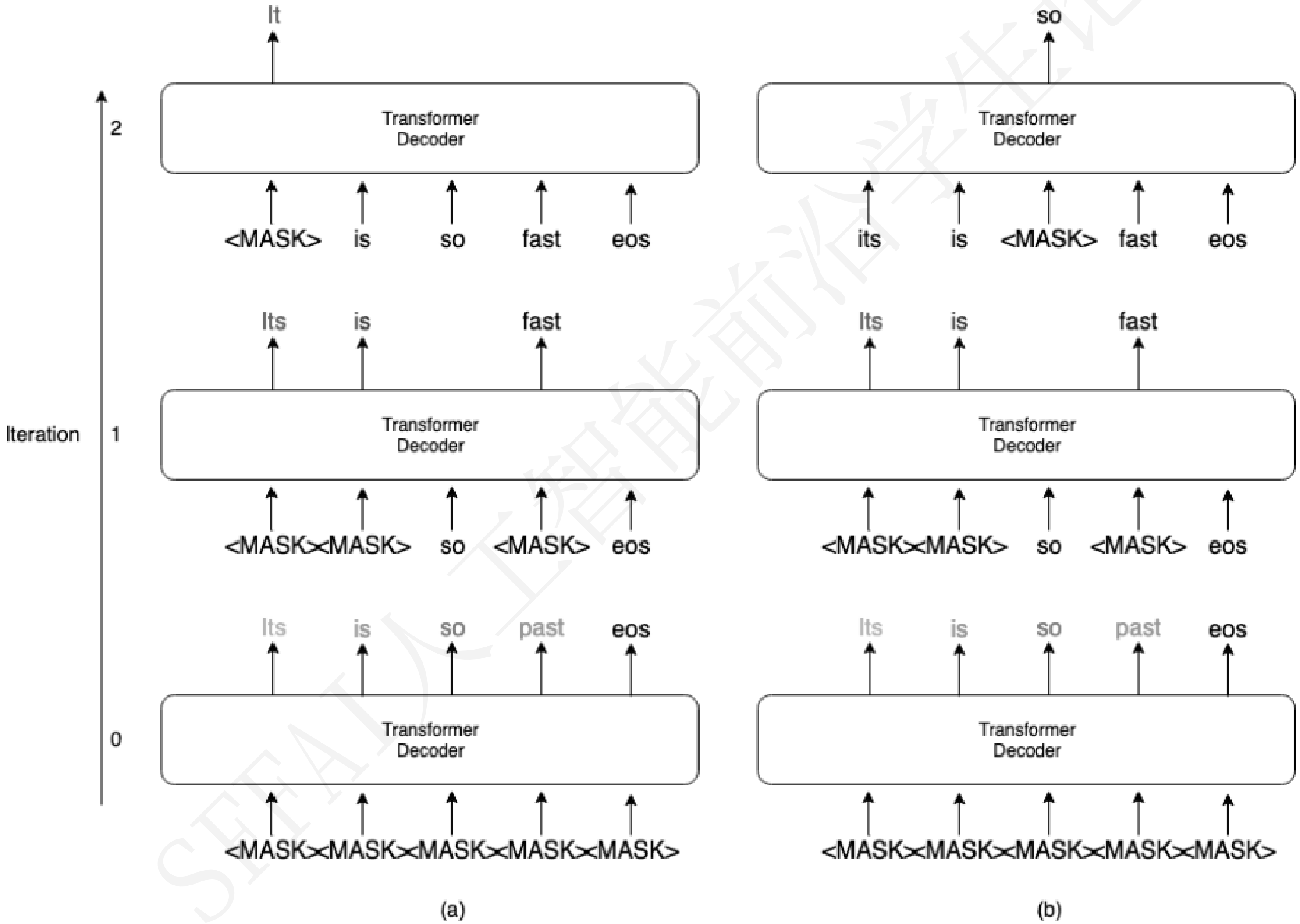
Model	**Online**	**Steps**	**Dev**	**Test**
LAS [20]	No	U	-	10.56
Transformer	No	U	7.80	8.64
RNN-T [10]	No	T	10.13	11.82
SA-T [10]	No	T	8.30	9.30
Chunk-Flow SA-T [10]	Yes	T	8.58	9.80
Sync-Transformer	Yes	U+M	7.91	8.91

3. Non-Autoregressive Transformer

- Autoregressive Model

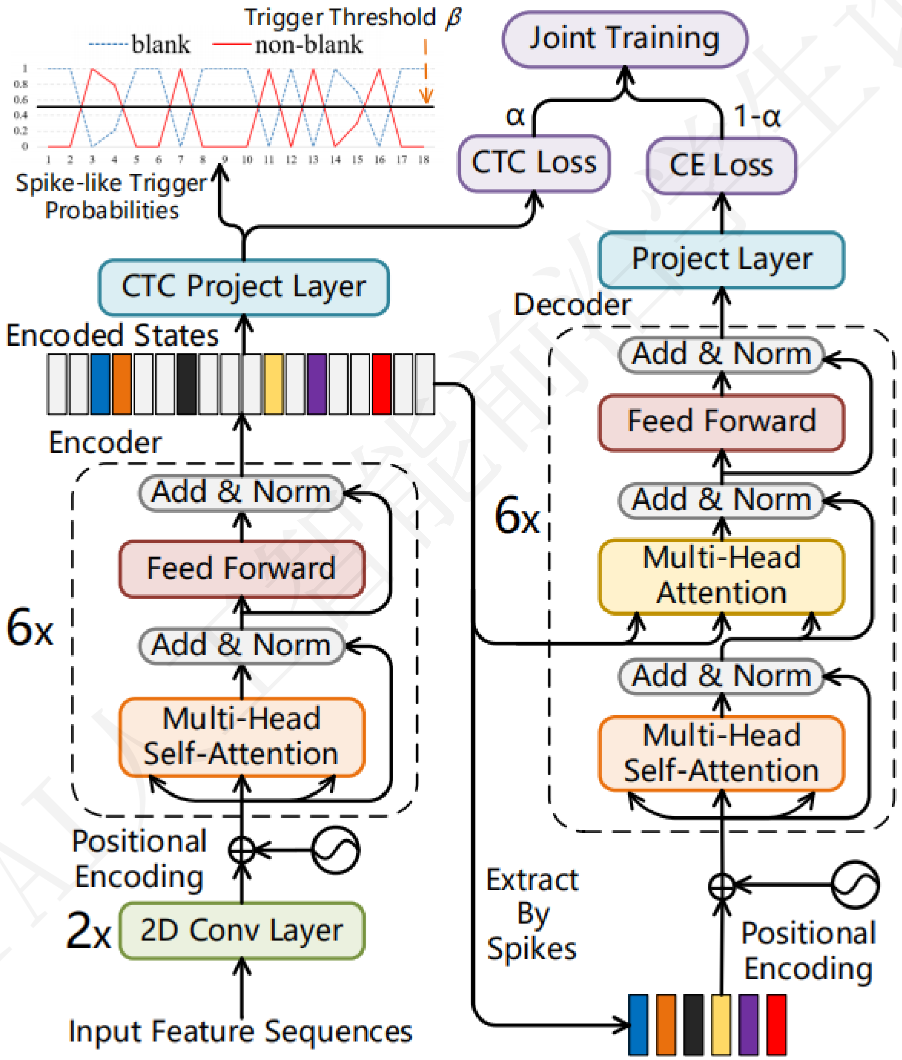


◎Inference Procedure



System	Dev CER	Test CER	Real Time Factor
Baseline(Transformer)	**6.0**	**6.7**	1.44
Baseline(Kaldi nnet3)	-	8.6	-
Baseline(Kaldi chain)	-	7.5	-
An et al. (2019)	-	6.3	-
Fan et al. (2019)	-	6.7	-
Easy first(K=1)	6.8	7.6	0.22
Easy first(K=3)	6.4	7.1	0.22
Mask-predict(K=1)	6.8	7.6	0.22
Mask-predict(K=3)	6.4	7.2	0.24
A-FMLM(K=1)	**6.2**	**6.7**	0.28
A-FMLM(K=2)	**6.2**	6.8	0.22

◎Spike-Triggered Non-Autoregressive Transformer



◎Spike-Triggered Non-Autoregressive Transformer

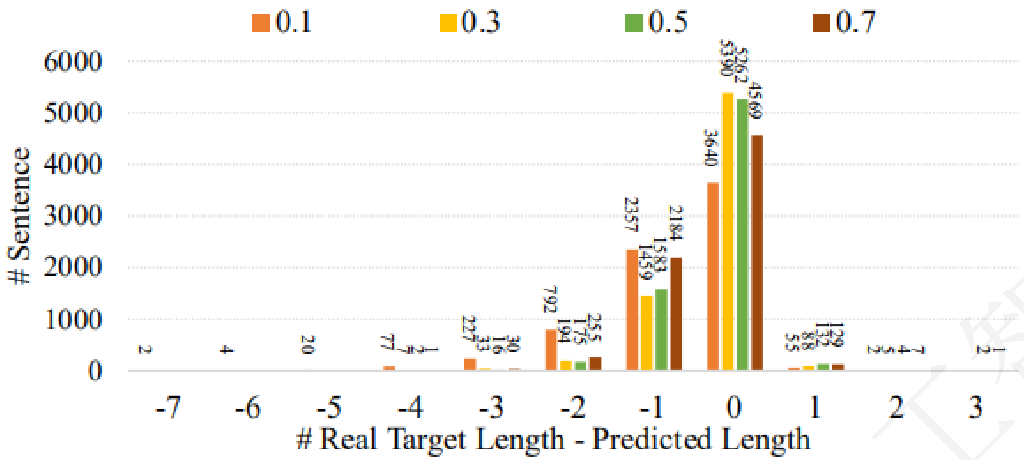
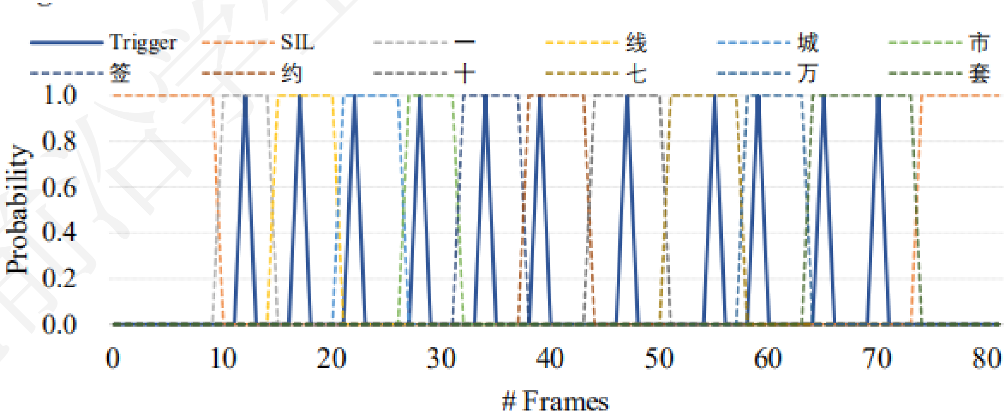
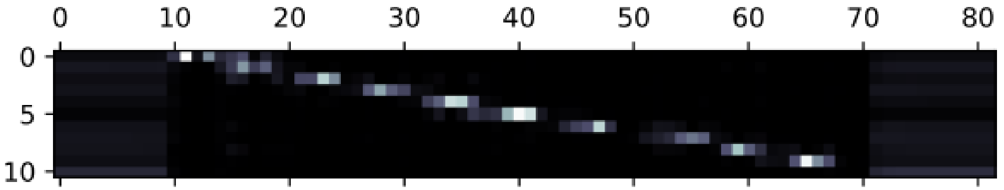


Figure 2: *The analysis of the predicted length. The histogram shows the difference between the target length and the predicted length.*



(a) The realtionship bettween trigger and word boundaries



(b) Attention mechanism visualization

◎Spike-Triggered Non-Autoregressive Transformer

Table 3: *Compare with other models in performance and real-time factor.*

Model	DEV	TEST	RTF
TDNN-Chain (Kaldi) [21]	-	7.45	-
LAS[22]	-	10.56	-
Speech-Transformer *	6.57	7.37	0.0504
SA-Transducer † [16]	8.30	9.30	0.1536
SAN-CTC * [23]	7.83	8.74	0.0168
Sync-Transformer † [24]	7.91	8.91	0.1183
NAT-MASKED * [11]	7.16	8.03	0.0058
ST-NAT(ours)	6.88	7.67	**0.0056**
ST-NAT+LM(ours)	**6.39**	**7.02**	0.0292

* These models are re-implemented by ourselves according to the papers.
† We supplement the RTF of our previous two models.

4. Conclusion

- There is still a lot of room to improve for the streaming end-to-end models.

- Non-Autoregressive Transformers can achieve a comparable performance with the autoregressive transformer.

Thanks