UNIVERSITY OF CAMBRIDGE

Label-synchronous Neural Transducer

Keqi Deng

Supervisor: Prof. Phil Woodland

Engineering Department

Table of Contents

Background: Neural Transducer

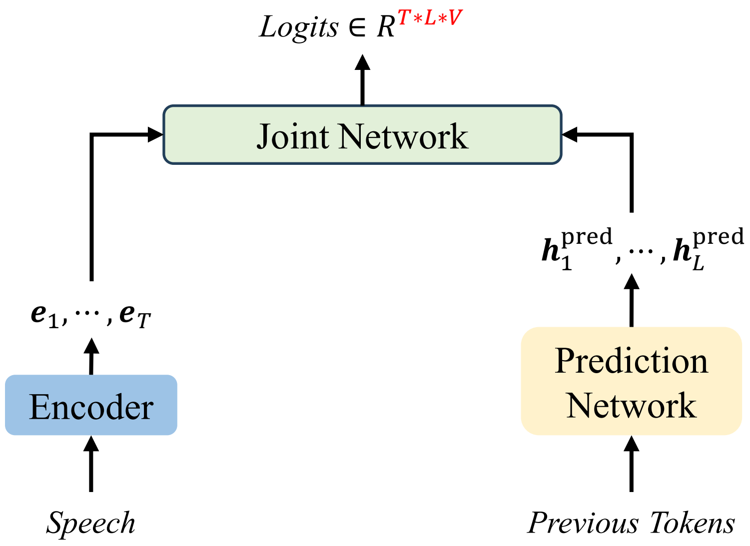
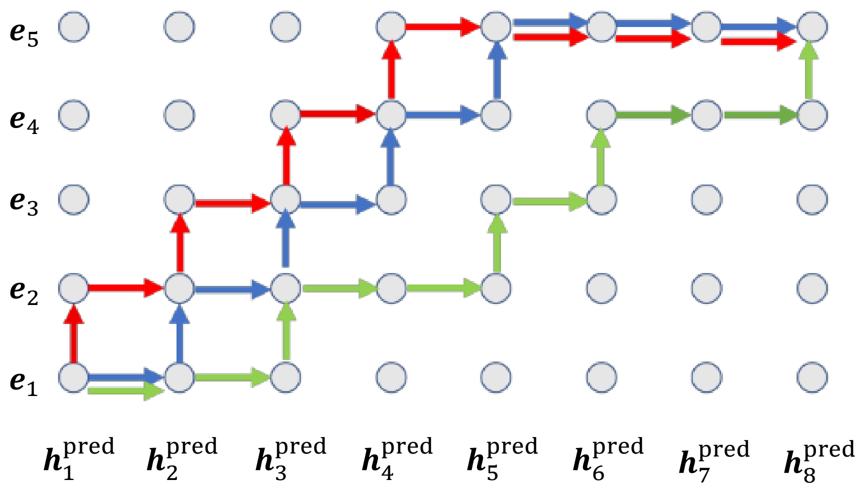


Fig. 1. Illustration of the standard neural transducer. The output logits is a three-dimensional tensor, where T and L are encoder output frame number and label length, and V denotes vocabulary size.



- Neural transducer is frame-synchronous, in which speech is decoded on a per-frame basis.
- Strengths:
  - Streaming Property
- Weaknesses:
  - Text-only Data Utilisation
- Attention-based encoder-decoder (AED) is label-synchronous, which is decoded on a per-label basis and not naturally equipped with streaming.

Background and Motivation

- Standard neural transducer is a frame-synchronous method, which combines encoder output and prediction network output at frame level. Output is a 3-dimensional tensor (R^{T*L*V}).

- Standard neural transducer needs the blank token to augment the output sequence. However, blank token generation means that prediction network cannot be viewed as an explicit LM due to inconsistency with LM task.

- **Motivation:** Want to combine encoder output and prediction network at label level, hence do not need blank tokens. Therefore, operation is label-synchronous and prediction network performs as a standard LM.

- **Challenge:** How to keep the valuable streaming property of standard neural transducer with label-synchronous operation?

UNIVERSITY OF CAMBRIDGE

Label-Synchronous Neural Transducer for Adaptable Online E2E Speech Recognition

Keqi Deng, Philip C. Woodland

IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2024

Background: Continuous Integrate-and-Fire (CIF)

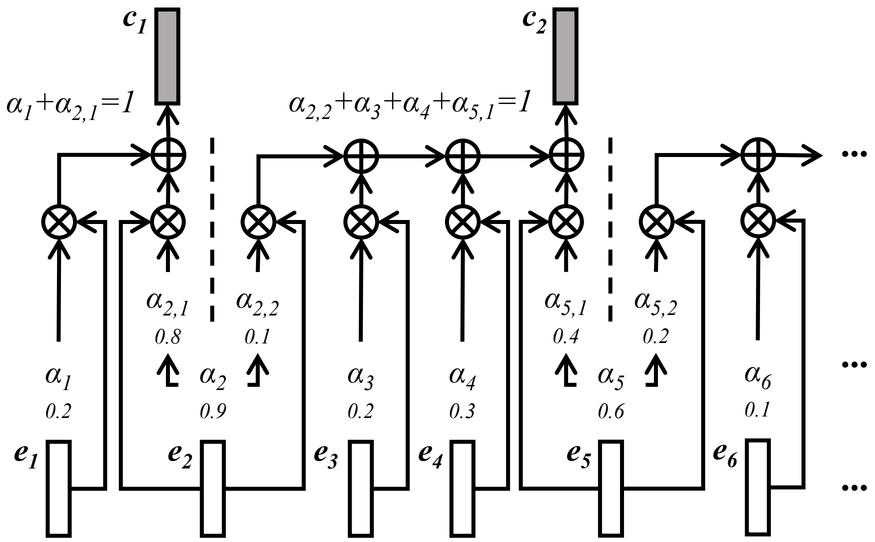


Fig. 2. Example of CIF [1]. $\oplus$ and $\otimes$ denote addition and multiplication. $\mathbf{E} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T)$ denotes encoder output and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T)$ represents predicted weights whose example values are (0.2, 0.9, 0.2, 0.3, 0.6, 0.1…) .

- CIF accumulates frame-level weights from left to right to locate boundaries.
- CIF is a non-autoregressive method.
- CIF estimates a monotonic alignment for streaming ASR.
- During training, a scaling strategy is used to ensure the integrated acoustic representations have the same length L as the target sequence, but this causes a mismatch between training and decoding.

[1] L. Dong and B. Xu, "CIF: Continuous Integrate-And-Fire for End-To-End Speech Recognition," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 6079-6083, doi: 10.1109/ICASSP40776.2020.9054250.

Label-synchronous Neural Transducer (LS-Transducer)

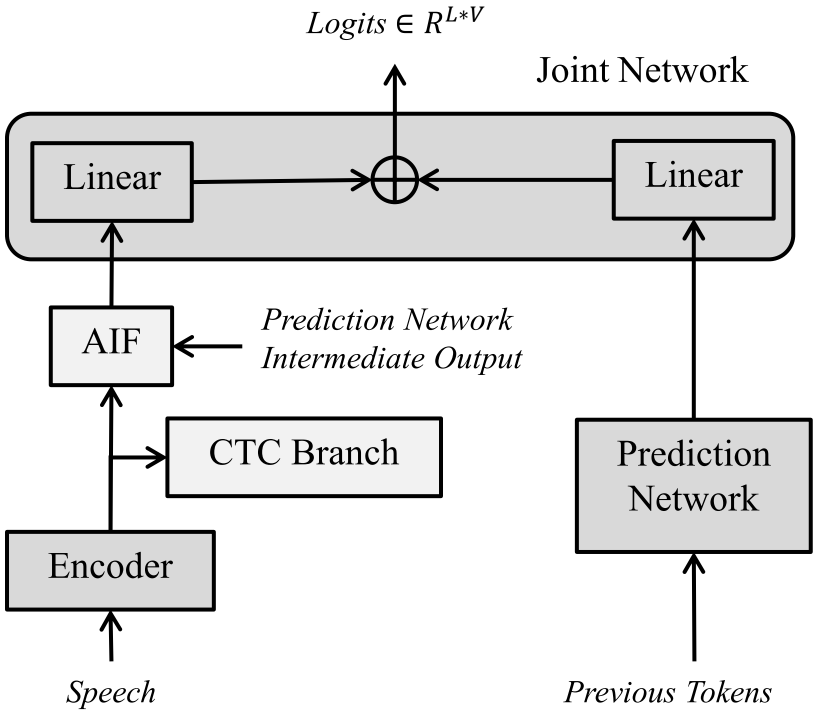


Fig. 3. Illustration of LS-Transducer. Linear denotes linear classifier. The output logits is a label-level two-dimensional matrix, where L and V are the label length and vocabulary size. ⊕ denotes addition.

- An Auto-regressive Integrate-and-Fire (AIF) is proposed to extract label-level encoder representation while maintaining streaming.
- The joint network adds the logits obtained from the AIF and prediction network.
- The output logits is a 2-dimensional tensor (R^{L*V}). Cross-entropy loss is used instead of RNN-T loss as the training objective.
- The prediction network works as a standard LM that can be flexibly fine-tuned on text-only data.

Auto-regressive Integrate-and-Fire (AIF)

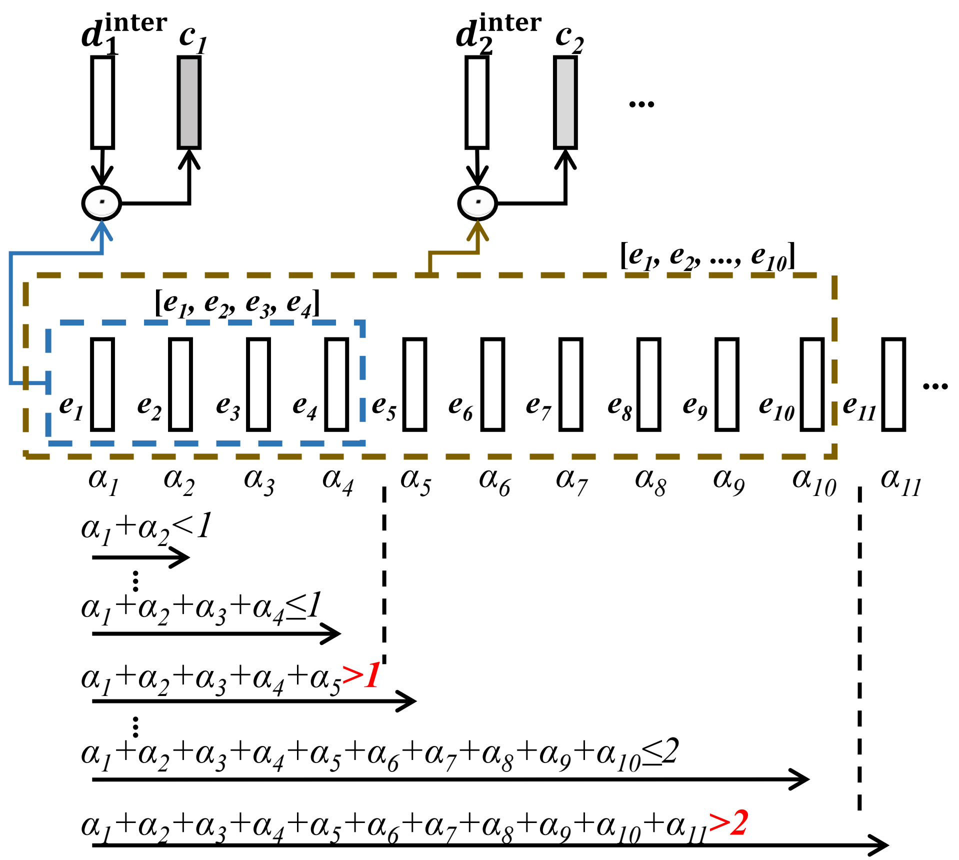


Fig. 4. Illustration of AIF. ⊙ denotes dot-product attention, whose $\boldsymbol{d}_j^{\text{inter}}$ is the intermediate output of the prediction network.

- Key difference with CIF: Additional input, i.e. prediction network, as the attention queries to generate representation auto-regressively.

- To extract label-level representation $\boldsymbol{c_j}$ where $j \in (1, L)$, AIF steps:
  1. Learn a frame-level weight α_i for each encoder frame.
  2. Accumulate α_i from left to right until it exceeds j, at which the time step is denoted as $T_j + 1$.
  3. Extract $\boldsymbol{c}_j$ via dot-product attention:

$$\boldsymbol{c}_j = \text{softmax}(\boldsymbol{d}_j^{\text{inter}} \cdot \mathbf{E}_{1:T_j}) \cdot \mathbf{E}_{1:T_j}$$

  where $\boldsymbol{d}_j^{\text{inter}}$ is prediction network intermediate output at the j-th step and $\mathbf{E} = (\boldsymbol{e}_1, \dots, \boldsymbol{e}_T)$ denotes encoder output.

- AIF uses accumulated frame-level weights to locate unit boundaries and dot-product to generate label encoder representation auto-regressively.

- Quantity loss is used to learn the alignment flat-start:

$$L_{qua} = ||L - \sum_{i=1}^{T} \alpha_i ||_1$$

Streaming Joint Decoding

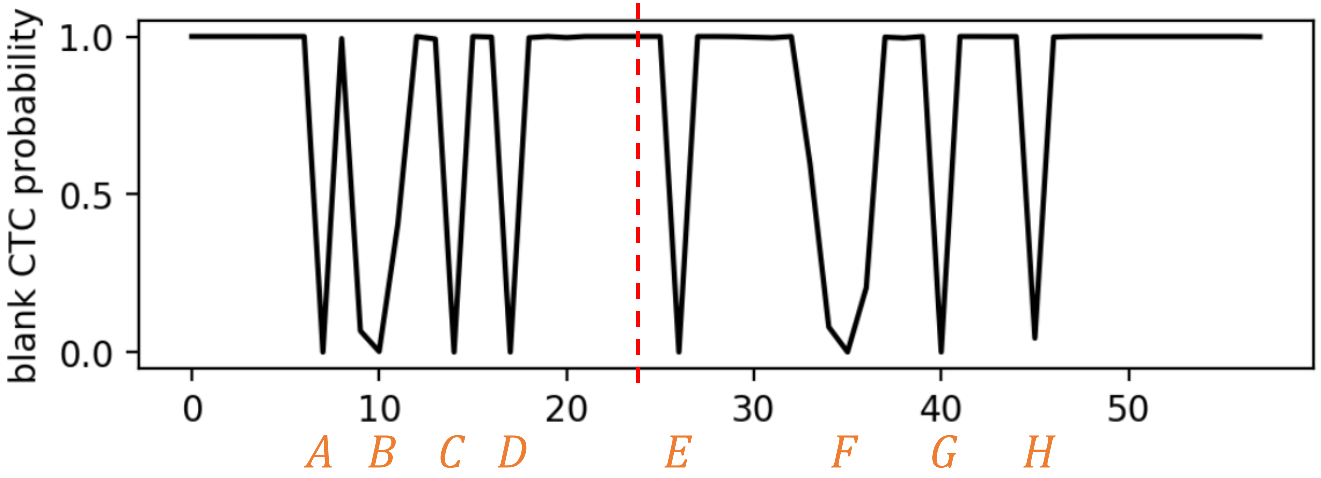
- Standard CTC prefix score requires complete speech utterance, i.e. offline:
  - Suppose g is a partial hypothesis, c is a token appended to g, and h is the new hypothesis such that $h = g \cdot c$
  - If c is a normal vocab token:

$$p_{ctc}(h, \ldots|X) = \sum_{v \in (U \cup [eos])} p_{ctc}(h \cdot v|X)$$

  - If c is end-of-sentence ([eos]):

$$p_{ctc}(h|X) = p_{ctc}(g|X) = \gamma_T^{(n)}(g) + \gamma_T^{(b)}(g)$$

- Online CTC prefix score $p_{ctc}(h|X_{1:t})$ was used to approximate $p_{ctc}(h|X)$.



- Proposed Modified online CTC prefix scores for [eos]:

$$p_{ctc}(h|X_{1:t}) = \begin{cases} p_{ctc}(h, \ldots|X_{1:t}), t < T \\ \gamma_T^{(n)}(g) + \gamma_T^{(b)}(g), t = T \end{cases}$$

- Where $h = g \cdot [\text{eos}]$

Initial Experiments

- Online ASR models were trained on LibriSpeech-100h.
- Offline AED had the same streaming encoder but was trained and decoded in an offline way.
- HAT and Factorised T-T are two variants of the neural transducer to achieve internal LM estimate and adaptation.

Online Model	Test-clean	Test-other	Dev-clean	Dev-other
Offline AED **Topline** model	4.4	11.3	4.2	11.3
HAT [2]	5.4	12.2	5.1	12.1
Factorised T-T [3]	5.4	12.4	5.3	12.4
LSTM-Prediction Network T-T	5.3	12.5	5.1	12.5
Stateless-Prediction Network T-T	5.6	12.6	5.5	12.6
Transformer-Prediction Network T-T	5.1	12.0	4.9	12.0
LS-Transducer	4.4	11.0	4.1	10.8

The HAT and Factorised T-T results were obtained based on our implementation.

- **Prediction network of the LS-Transducer performs as an explicit LM.**
- **Initialising it by a trained source-domain LM is very effective to boost ASR performance.**

[2] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (HAT)," in Proc. ICASSP, 2020.
[3] X. Chen, Z. Meng, S. Parthasarathy, and J. Li, "Factorized neural transducer for efficient language model adaptation," in Proc. ICASSP, 2022.

Experiments on LibriSpeech-960

Intra-domain Experiments:

- Online ASR models were trained on LibriSpeech-960h
- External LM shallow fusion was used.

Online Model	Test-clean	Test-other	Dev-clean	Dev-other
Transformer-Prediction Network T-T	3.1	7.7	2.9	7.5
LS-Transducer	2.7	6.8	2.6	6.7

Cross-domain Experiments:

- Directly decode the ASR models on cross-domain sets: TED-LIUM 2 and AESRC2020 test sets.

Online Model	LibriSpeech -> TED-LIUM 2		LibriSpeech -> AESRC2020	
	Test	Dev	Dev	Test
Transformer-Prediction Network T-T	12.7	13.1	19.0	18.7
+Target-domain LM Shallow Fusion	11.9	12.2	16.7	16.2
Label-Synchronous Transducer	11.7	12.0	18.2	17.8
+ Adapted prediction net (internal LM)	10.0	10.3	14.9	14.1
++Target-domain LM Shallow Fusion	9.1	9.6	13.6	12.6

Ablation Studies on AIF:

Online Model	Train Data	Test-clean	Test-other	Dev-clean	Dev-other
Transformer-Prediction Network T-T	LS100	5.1	12.0	4.9	12.0
LS-Transducer w/ AIF	LS100	4.4	11.0	4.1	10.8
LS-Transducer w/ CIF	LS100	7.0	13.3	6.5	13.2
Transformer-Prediction Network T-T	LS960	3.2	8.0	3.0	7.8
LS-Transducer w/ AIF	LS960	3.0	7.2	2.9	7.4
LS-Transducer w/ CIF	LS960	4.6	9.0	4.3	8.7

Ablation Studies on Prediction Network Initialisation:

External LM shallow fusion was not used.

Online ASR Model on LS960	Test-clean	Test-other	Dev-clean	Dev-other
Transformer-Prediction Network T-T	3.2	8.0	3.0	7.8
w/ pre-trained prediction network	4.5	9.6	4.2	9.5
LS-Transducer	3.0	7.2	2.9	7.4
w/o pre-trained prediction network	3.5	8.1	3.5	8.0

Ablation Studies on Streaming Joint Decoding:

Online Model	Test-clean	Test-other	Dev-clean	Dev-other
Transformer-Prediction Network T-T	3.2	8.0	3.0	7.8
Proposed LS-Transducer:				
w/ streaming joint decoding	3.0	7.2	2.9	7.4
w/o modification for [eos]	6.8	9.9	6.0	10.0
w/o streaming joint decoding	3.9	7.9	3.5	7.8

- Directly using online CTC prefix scores damages the performance.
- The modification to the [eos] is simple but effective.

Visual Examples:

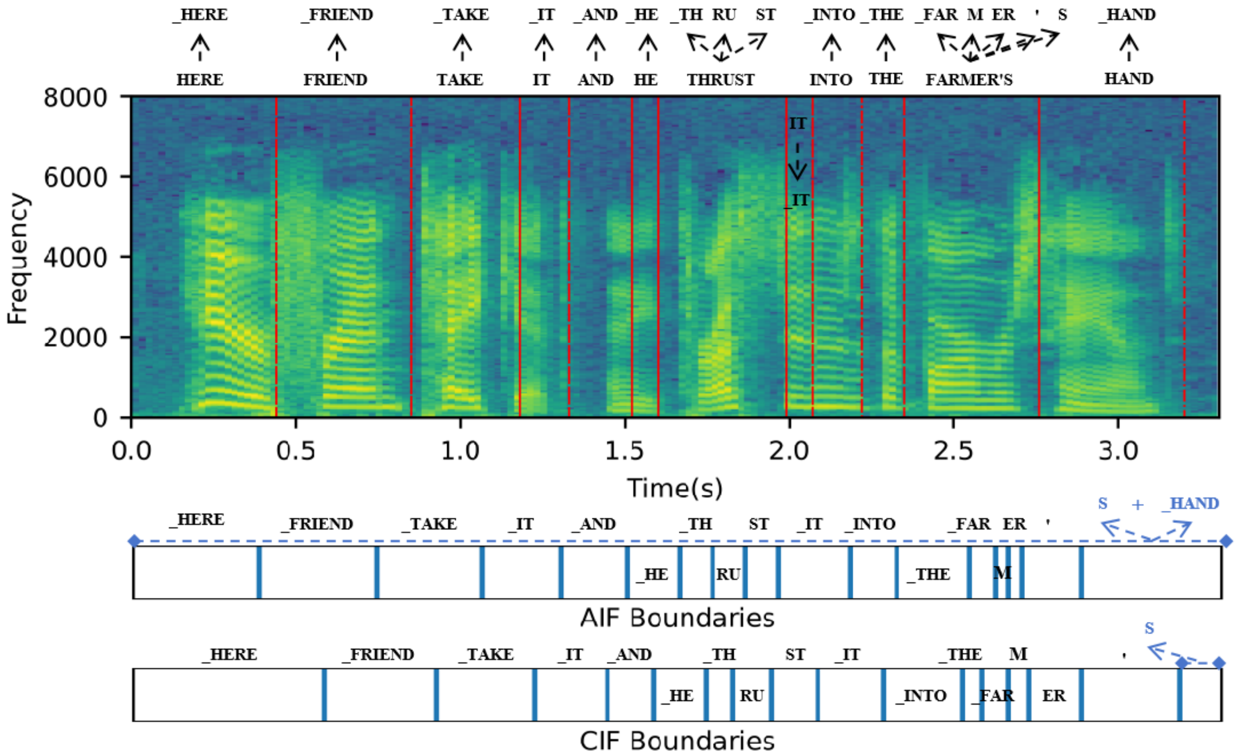


Fig. 5. An example of AIF not accurately locating the BPE boundaries but still predicting the correct transcript.

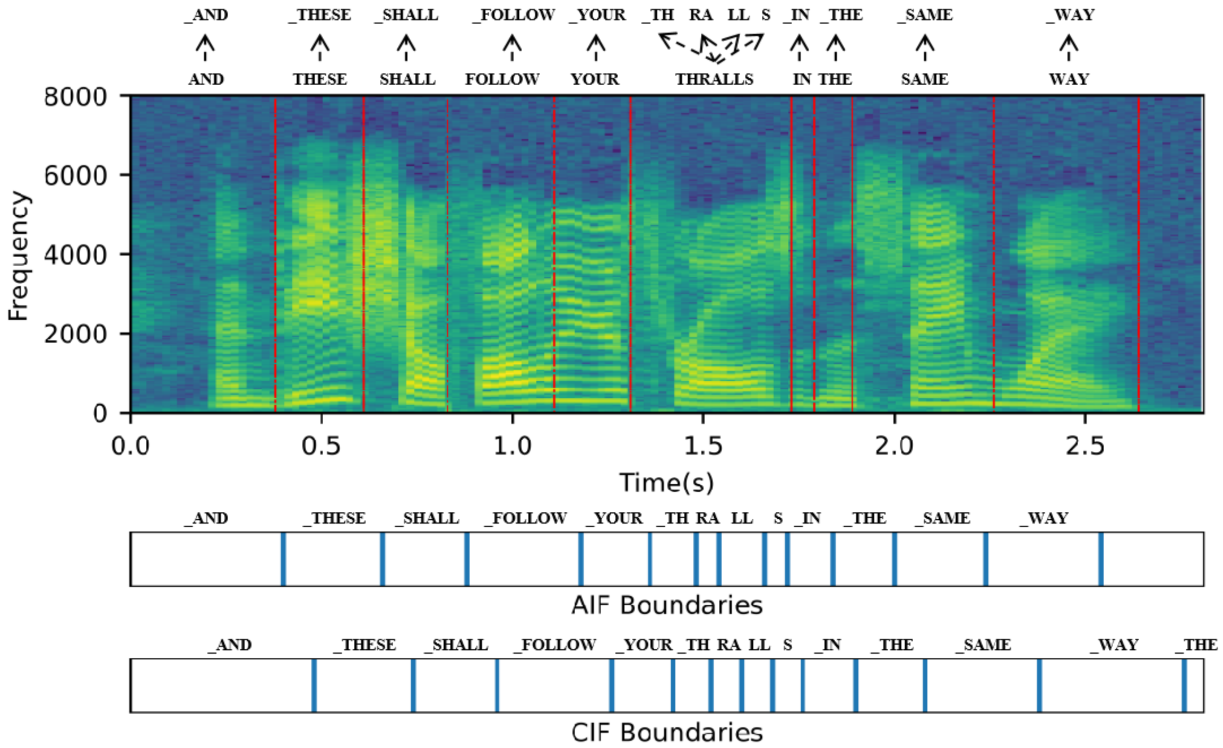


Fig. 6. An example of AIF accurately locating the BPE boundaries but still predicting the correct transcript.

- **Red dashed line: right-side word boundary**.
- **Black dotted arrow:** points from the ground truth word to the **corresponding BPE units**.
- **Solid blue line: right-side BPE boundary located by AIF or CIF**.

UNIVERSITY OF CAMBRIDGE

Label-Synchronous Neural Transducer for E2E Simultaneous Speech Translation

Keqi Deng, Philip C. Woodland

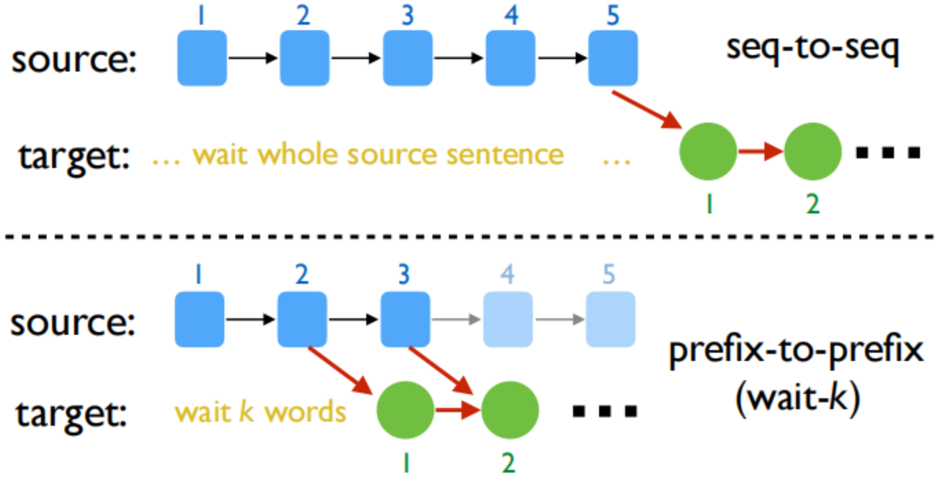
ACL, 2024

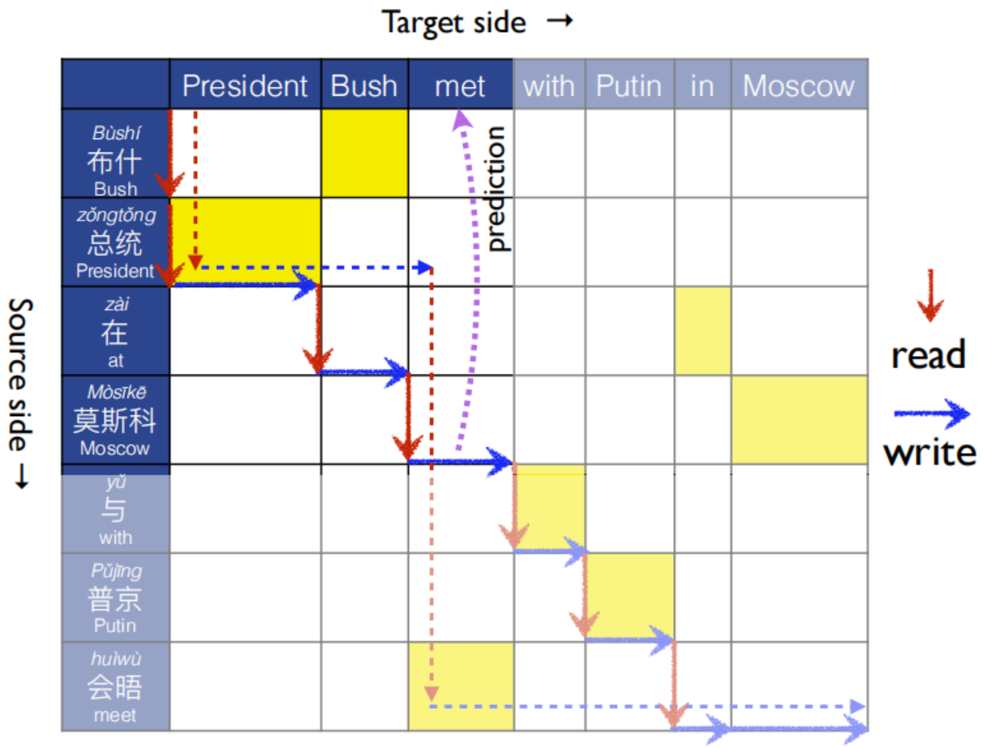
Background and Motivation

- Simultaneous speech translation (SST) harder than streaming ASR since needs both online and re-ordering capabilities.
- Neural transducer widely used in streaming ASR, but with limited re-ordering abilities due to monotonic property.
- AED can be modified for streaming ASR but always with higher latency as the attention alignment is too flexible and hard to control.
- LS-Transducer uses accumulated weights to decide when to emit tokens and uses an attention mechanism to extract label-level representation, thus it has an obvious potential to be naturally equipped with both streaming and re-ordering capabilities.
- **Motivation:** A framework for SST that can naturally handle streaming and re-ordering. Since the prediction network of LS-Transducer can effectively utilise monolingual text-only data, the E2E SST data sparsity issue can be alleviated.
- **Challenge:** How to **adapt AIF for SST**?
  How to flexibly **control the quality-latency trade-off**?
  How to achieve **incremental decoding**?

Background: Wait-k – Fixed Policy

- Wait-k policy [1] from simultaneous machine translation:





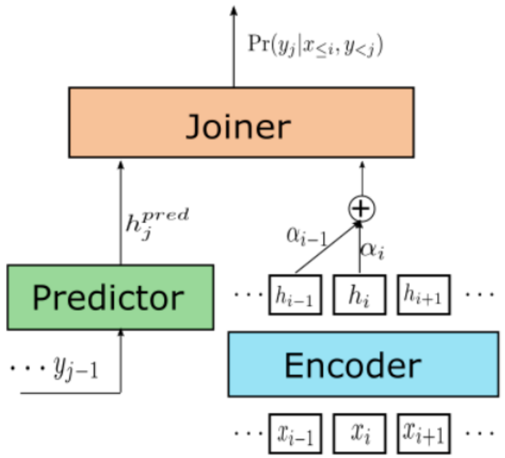
- Wait-*k* policy in SST [2] : Wait-*k* policy was adapted to simultaneous speech translation:
  - Source text has clear word or BPE boundaries, while speech does not explicitly have it.
  - [2] designs a fixed pre-decision policy to treat every fixed number Δt of frames as one step.

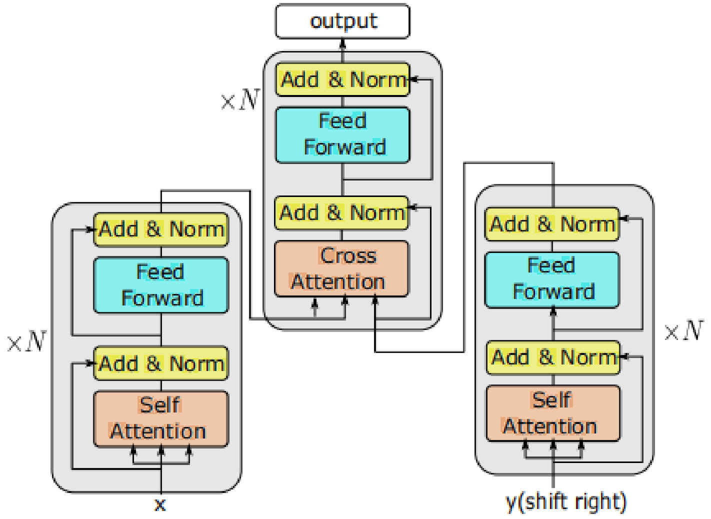
[4] Ma, M., Huang, L., Xiong, H., Zheng, R., Liu, K., Zheng, B., Zhang, C., He, Z., Liu, H., Li, X., Wu, H., & Wang, H. (2018). STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework. Annual Meeting of the Association for Computational Linguistics.

[5] Ma, X., Pino, J.M., & Koehn, P. (2020). SimulMT to SimulST: Adapting Simultaneous Text Translation to End-to-End Simultaneous Speech Translation. AACL.

Background: CAAT – Flexible Policy

- Cross Attention Augmented Transducer (CAAT) [6]:





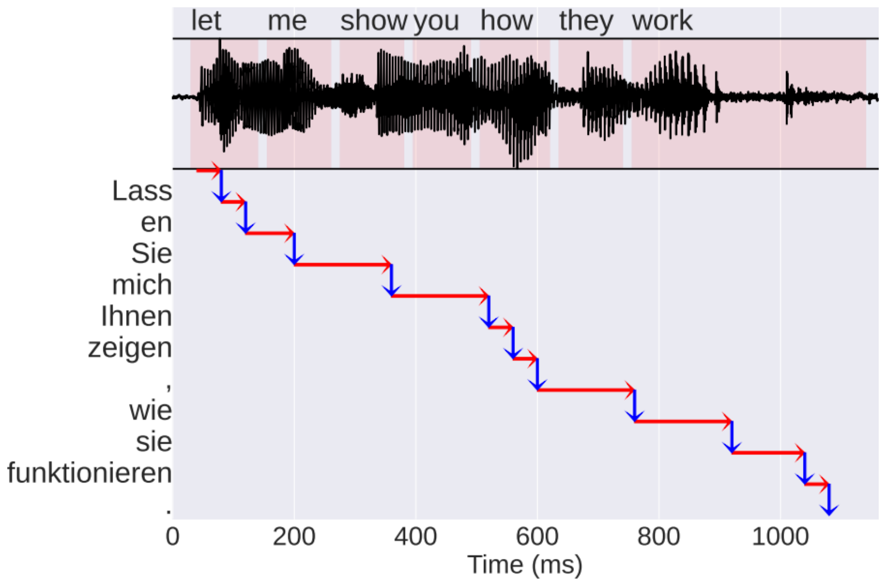
$$s_{i,j} = Att(Q,K,V) = Att(h_j^{pred}, h_{\leq i}^{enc}, h_{\leq i}^{enc})$$

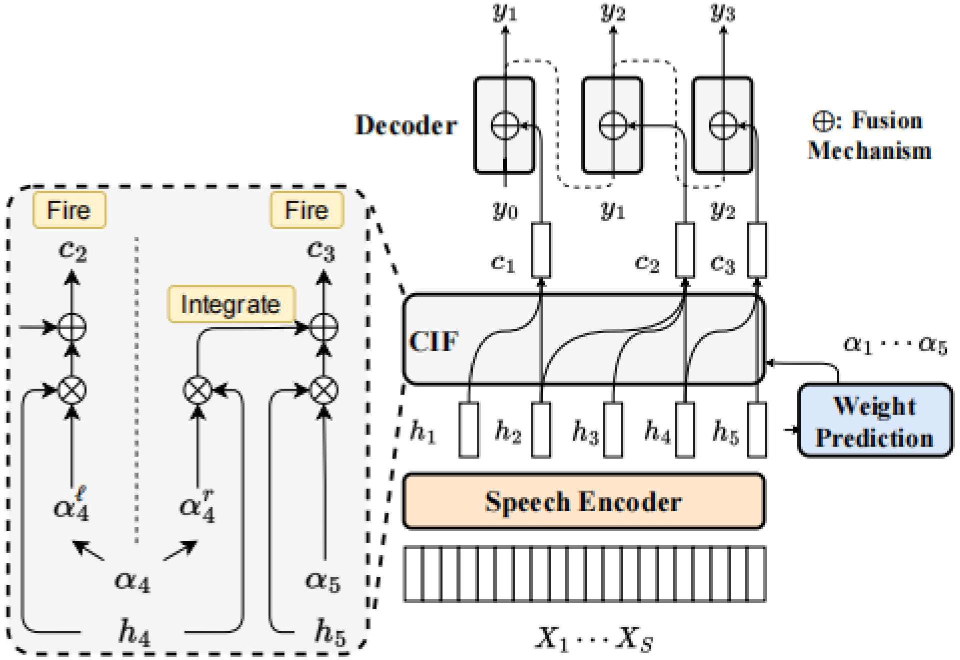
- CAAT is more expensive than a standard neural transducer:
  - The complexity of joiner is $O(|x| \cdot |y|)$ at training.
  - For standard neural transducer, joiner is more efficient as it only simply adds hidden states, while CAAT involves many attention operations. The complexity of CAAT can be about $|x|$ times higher than a normal AED.

[6] Liu, D., Du, M., Li, X., Li, Y., & Chen, E. (2021, November). Cross attention augmented transducer networks for simultaneous translation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 39-55).

Background: CIF-IL – Flexible Policy

- CIF with infinite lookback decoder (CIF-IL) [7]:





- It was shown that CIF-IL is suitable for handling SST in low and medium-latency scenarios.

[7] Chih-Chiang Chang and Hung-yi Lee. 2022. Exploring continuous integrate-and-fire for adaptive simultaneous speech translation. In Proc. Interspeech,

Label-synchronous Neural Transducer for SST (LS-Transducer-SST)

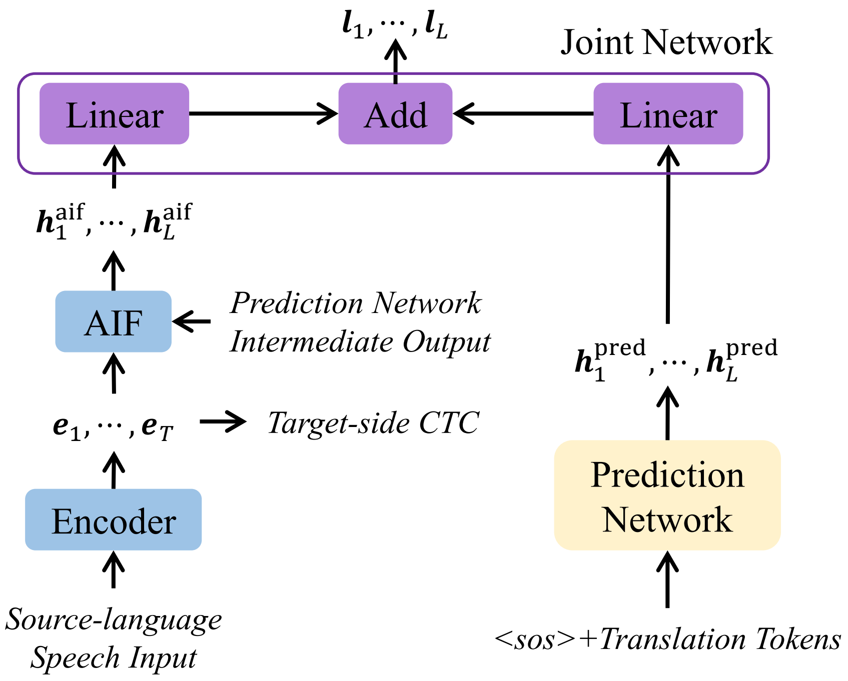


Fig. 7. Illustration of LS-Transducer-SST. Linear denotes linear classifier. Target-side CTC uses translations in the training objective computation.

- AIF is directly used to dynamically decide when to emit translation tokens.
- Quantity loss is also used with target translation sequence length as the objective.
- To help AIF learn this cross-lingual speech-text alignment, a target-side CTC is computed to encourage the Transformer encoder to re-order the output according to the target translation sequence, as found by [8].

[8] Shun-Po Chuang, Yung-Sung Chuang, Chih-Chiang Chang, and Hung-yi Lee. 2021. Investigating the re-ordering capability in CTC-based non-autoregressive end-to-end speech translation. In Proc. ACL/IJCNLP (Findings), Online.

Latency-controllable AIF

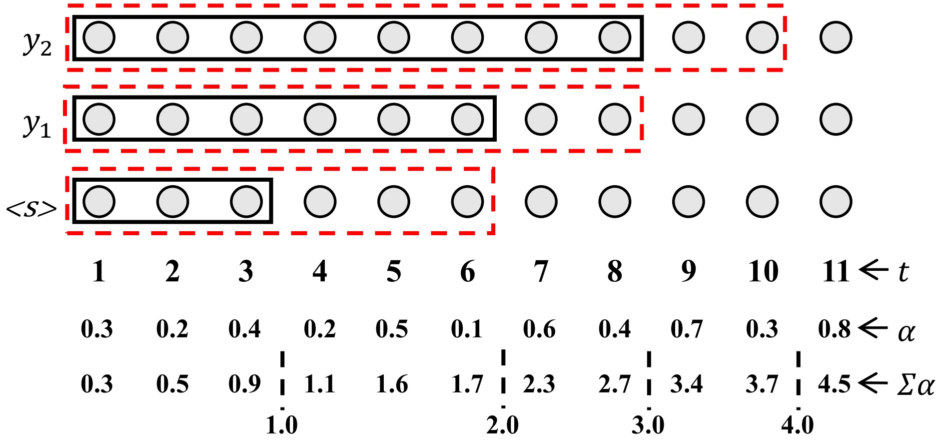


Fig. 8. Illustration of latency-controllable AIF. t denotes the time step. α is the frame-level weight. The black solid line shows when the tokens are emitted under standard AIF; the red dotted line illustrates the case when the AIF decision threshold is increased by 1.

- Originally, decision threshold of i-th translation token is i. By adding a hyper-parameter ϵ into decision threshold, i.e. $i+\epsilon$, quality-latency trade-off can be controlled.

- Latency-controllable AIF allows quality-latency trade-off to be controlled not only at training but also at decoding.

- Advantages of Latency-controllable AIF:
  - One hyper-parameter ϵ to achieve fine-grained latency control and can meet any latency requirements.
  - Typical flexible policy uses latency loss to control quality-latency trade-off at training. Latency-controllable AIF can control latency only at decoding.

Chunk-based Incremental Joint Decoding

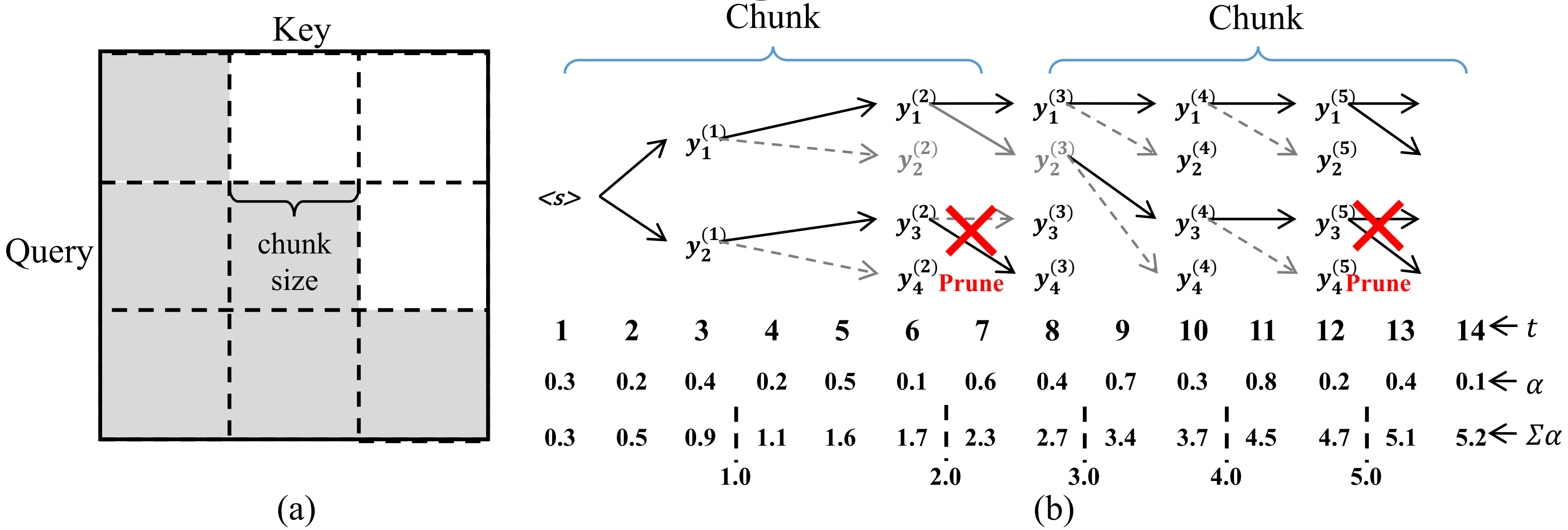


Fig. 9. Illustration of chunk-based incremental joint decoding. (a) an illustration of the chunk-based mask; (b) an example of the chunk-based incremental pruning according to the accumulated AIF weights, in which the chunk size is 7, the beam size is 2 within a chunk, the decision threshold of the i-th output is i.

- Chunk-based streaming strategy (as shown in Fig.7 (a)) is used for the Transformer encoder.
- SST normally requires that the translation prediction is not changed after being output. Hence, chunk-based incremental joint decoding is designed to prune the hypotheses to the same prefix within a chunk.
- The key point is to know in advance if the speech input required for the next token will exceed the range of this chunk: comparing the accumulation of frame-level weights up to the current chunk with the decision threshold of the next translation output token.
- The score of the target-side CTC branch is also considered during translation decoding to enhance translation quality, as found by [9].

[9] Keqi Deng, Shinji Watanabe, Jiatong Shi, and Siddhant Arora. 2022. Blockwise streaming transformer for spoken language understanding and simultaneous speech translation. In Proc. Interspeech, Incheon, Korea.

Experiments

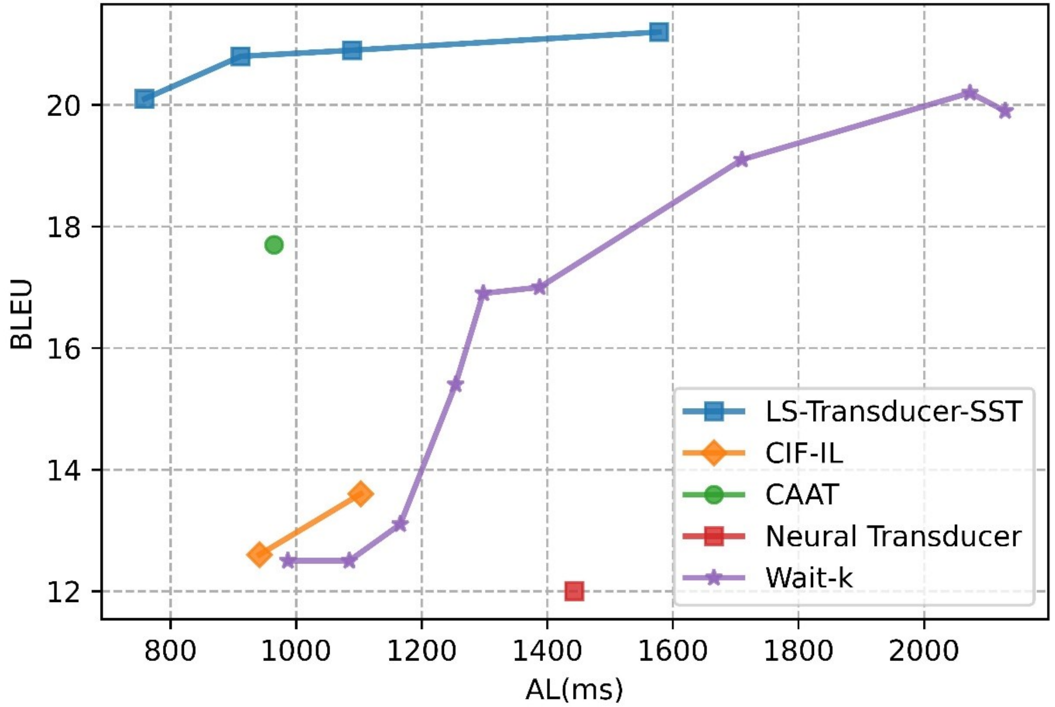


Fig. 10. Quality-latency trade-off curves on Fisher-CallHome Spanish CallHome-evltest set.

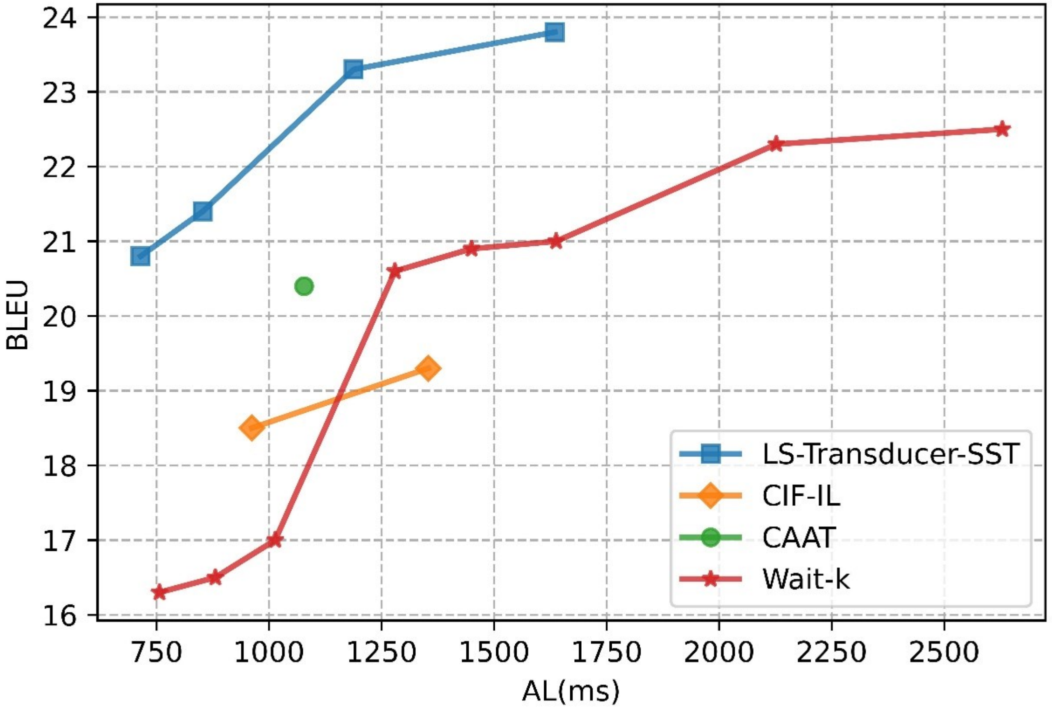


Fig. 11. Quality-latency trade-off curves on MuST-C En-De tst-COMMON set.

- CAAT [10] is a variant of the neural transducer, CIF-IL [11] is a CIF-based SST method, and Wait-k [12] is a typical AED-based SST method.
- We focus on the low and medium latency scenarios, i.e. AL < 1 s and 1 s < AL < 2 s

[10] Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In Proc. EMNLP, Online and Punta Cana, Dominican Republic.
[11] Chih-Chiang Chang and Hung-yi Lee. 2022. Exploring continuous integrate-and-fire for adaptive simultaneous speech translation. In Proc. Interspeech, Incheon, Korea.
[12] Xutai Ma, Juan Miguel Pino, and Philipp Koehn. 2020c. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In Proc. AACL/IJCNLP, Suzhou, China..

Cross-domain Experiments:

SST Models	Fisher-CallHome Spanish (FCS)		FCS -> Europarl-ST Es-En		
	Evltest (BLEU)	AL(s)	Test (BLEU)	Dev (BLEU)	AL(s)
Wait-5 with 360ms pre-decision	19.9	2.129	9.4	10.6	4.603
+Density Ratio	---	---	10.8	12.3	4.565
LS-Transducer-SST	20.1	0.759	10.4	11.7	0.915
+ Adapted prediction net (internal LM)	---	---	12.5	13.8	0.863
++Target-domain LM Shallow Fusion	---	---	12.8	14.3	0.931

- In cross-domain, the number of BPE units for target translations tends to be relatively longer than for the source domain as the BPE model is trained on source-domain text.

Ablation Studies on Prediction Network Initialisation:

SST Models on MuST-C En-De	tst-COMMON	tst-HE	AL (s)
CAAT	20.4	18.9	1.078
w/ pre-trained prediction network	18.1	16.5	1.068
LS-Transducer-SST	20.8	19.3	0.715
w/o pre-trained prediction network	19.3	18.3	0.704

- Pre-training the prediction network did not help for CAAT, which inherits the frame-synchronous property from the standard neural transducer.

Analysis of Latency-controllable AIF:

- Latency-controllable AIF with consistent training and decoding performed slightly better than the test-time-only one.
- Adjusting ε only in the decoding stage achieves similar results.

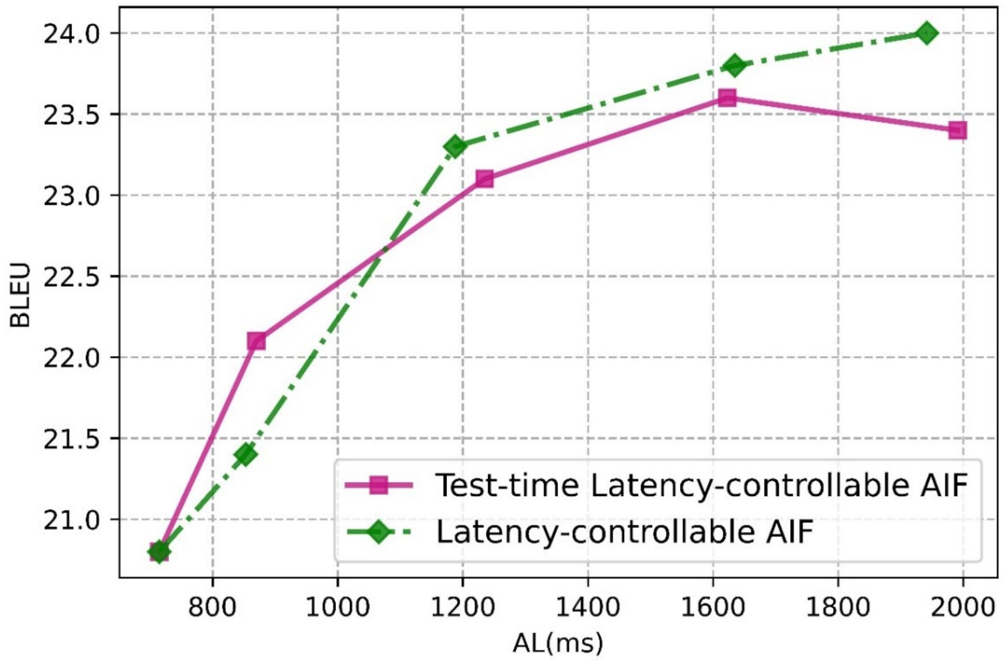


Fig. 12. Quality-latency trade-off of LS-Transducer- SST on MuST-C En-De tst-COMMON set. The 5 dots for the latency-controllable AIF are $\varepsilon \in \{0, 1, 3, 5, 7\}$.

Ablation Studies on Chunk-based Incremental Decoding:

SST Models on MuST-C En-De	tst-COMMON	tst-HE	AL (s)
LS-Transducer-SST	20.8	19.3	0.715
w/ tail beam search	18.5	17.4	0.761
w/ greedy search	17.8	16.8	0.760

- Beam search within each chunk expands search space.

Comparison with Recent Work (Literature Results)

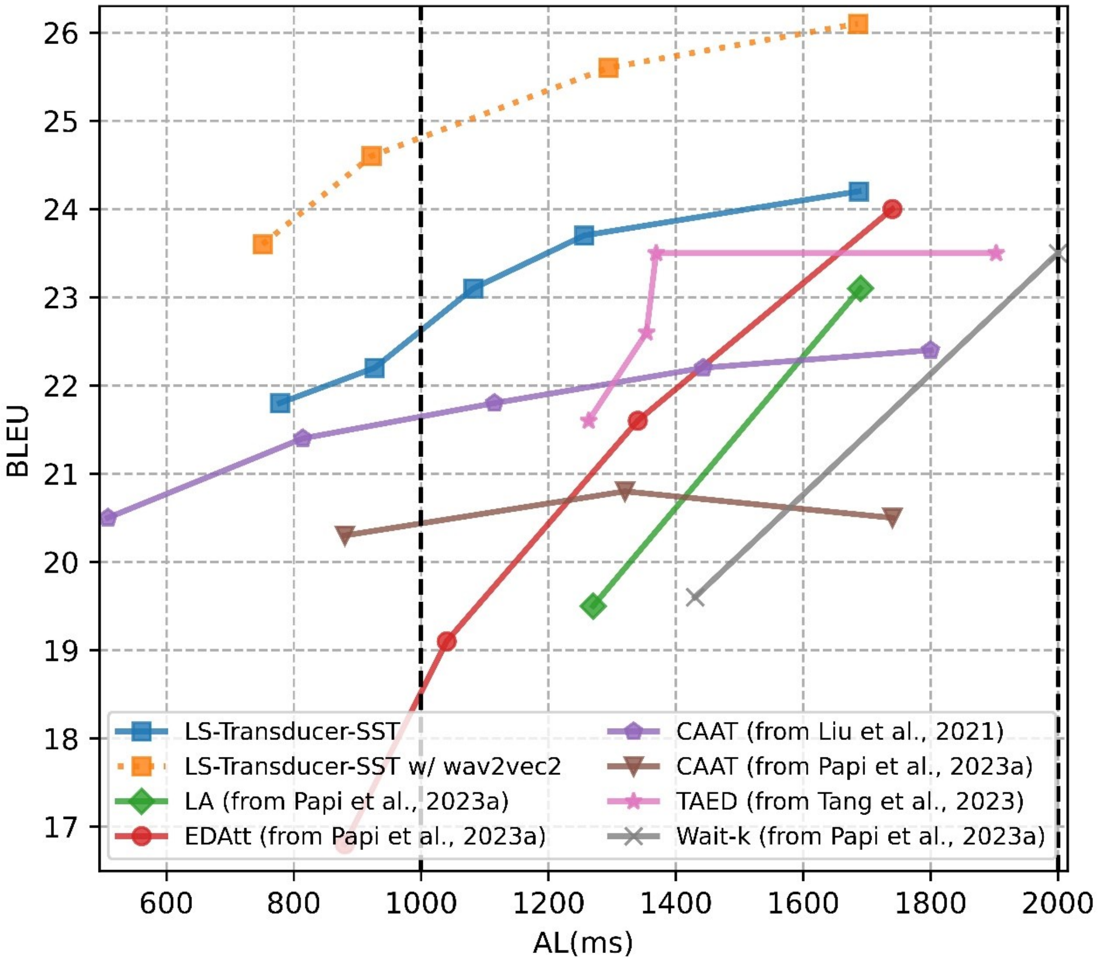


Fig. 13. Quality-latency trade-off curves on MuST-C En-De tst-COMMON set. Solid lines are comparable with technique results from literature. Dotted line indicates wav2vec2.0. All results use sequence-level KD

- LS-Transducer-SST (solid blue) outperforms other models in both low and medium-latency regions (up to an AL of about 1.7 s).
- Note in low and medium-latency scenarios, we use the same constant chunk size for the encoder for simplicity.
- When the latency approaches the end of the medium-latency region or even enters the high-latency region, the chunk size we are currently using is no longer suitable and increases the chunk size can greatly improve translation quality.

UNIVERSITY OF CAMBRIDGE

Conclusions

- LS-Transducer provides an alternative approach to the standard neural transducer.
- Streaming property has been maintained.
- Adaptation capability has been enhanced.
- Output is a 2-dimensional matrix, which is simpler than a 3-dimensional tensor in the standard neural transducer.
- LS-Transducer exceeds standard neural transducers with 12.9% and 24.6% relative WER reductions in intra-domain and cross-domain scenarios respectively.
- LS-Transducer-SST, naturally equipped with streaming and reordering abilities, is a natural solution for SST.
- LS-Transducer-SST gives a 3.1/2.9 point BLEU increase (Es-En/En-De) relative to CAAT at a similar latency and a 1.4 s reduction in average lagging latency with similar BLEU scores relative to Wait-k.

PETERHOUSE
UNIVERSITY OF CAMBRIDGE

Thanks for watching!

[github: https://github.com/D-Keqi/LS-Transducer-SST](https://github.com/D-Keqi/LS-Transducer-SST)

UNIVERSITY OF
CAMBRIDGE
Department of Engineering

Appendix: Continuous Integrate-and-Fire (CIF)

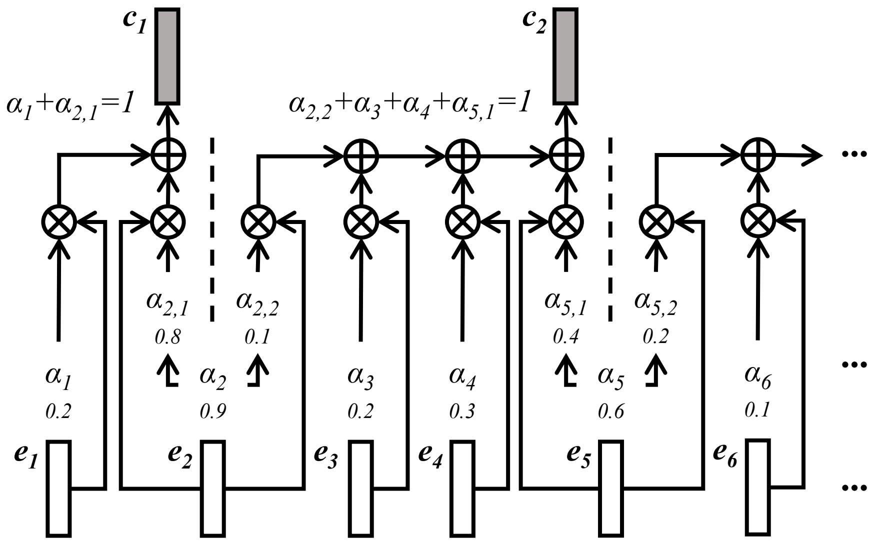


Fig. 2. Example of CIF [1]. $\oplus$ and $\otimes$ denote addition and multiplication. $\mathbf{E} = (\boldsymbol{e}_1, \dots, \boldsymbol{e}_T)$ denotes encoder output and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_T)$ represents predicted weights whose example values are (0.2, 0.9, 0.2, 0.3, 0.6, 0.1…) .

- During training, a scaling strategy is used to ensure the integrated acoustic representations have the same length L as the target sequence:

$$\hat{\alpha}_t = \alpha_t * \frac{L}{\sum_{i=1}^{T} \alpha_i}$$

but this causes a mismatch between training and decoding.

- Quantity loss is used to learn the alignment flat-start:

$$L_{qua} = ||L - \sum_{i=1}^{T} \alpha_i \,||_1$$

[1] L. Dong and B. Xu, "CIF: Continuous Integrate-And-Fire for End-To-End Speech Recognition," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 6079-6083, doi: 10.1109/ICASSP40776.2020.9054250.