Towards Streaming Speech Translation

Javier Iranzo-Sánchez

jairsan@upv.es

www.mllp.upv.es

Joint work with MLLP researchers

MLLP | Machine Learning and Language Processing

VRAIN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Contents

1 Introduction: Streaming Speech Translation 2

2 Introduction to Simultaneous MT 4

3 Streaming MT Evaluation 13

4 Streaming MT: Models & Baseline 24

1 Introduction: Streaming Speech Translation

Streaming Speech Translation

- Speech Translation for unbounded input audio streams
- Challenges
  - Processing and providing output in ***real-time***
  - ***Limited context*** to perform the recognition
- Realistic evaluation: Stream-level evaluation

Our approach

Cascade system

- Streaming ASR [Jorge et al., 2021]
  - Hybrid system: Chunk-based LSTM + Transformer LM

- Sliding-window RNN Segmenter [Iranzo-Sánchez et al., 2020]
  - EOS decision for every transcribed word

- ***MT (this talk)***
  - Evaluation [Iranzo-Sánchez et al., 2021]
  - Streaming-MT models [Iranzo-Sánchez et al., 2022]

2 Introduction to Simultaneous MT

(Sentence-level) Simultaneous Machine Translation

- Incrementally translate a sentence before it is fully available

- For every sentence pair $(\boldsymbol{x}, \boldsymbol{y})$,

$$\hat{y}_i = \arg\max_{y \in \mathcal{Y}} p\left(y \mid x_1^{g(i)}, y_1^{i-1}\right)$$

- Delay function $g(i)$: # src. words available for writing i-th word.

Simultaneous MT models

- A simultaneous MT model is characterized by its policy $g(i)$

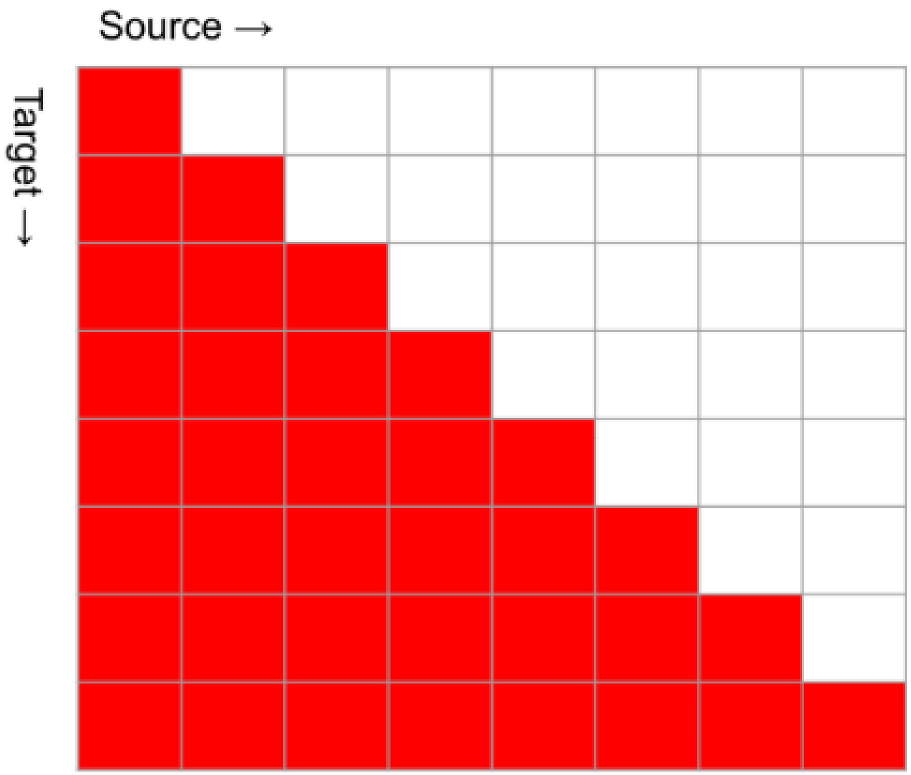
- At each timestep, the policy decides between 2 actions:
  - READ an input word (wait for more context)
  - WRITE an output word

- Baseline policy: Wait-k translation
  - First wait for k words to arrive (k READ),
  - then alternate between WRITE and READ

$$g(i) = \left\lfloor k + \frac{i-1}{\gamma_\theta} \right\rfloor$$

  - Length ratio:

$$\gamma = \frac{|\boldsymbol{y}|}{|\boldsymbol{x}|}$$



(Image source: [Huang et al., 2020])

Simultaneous MT Evaluation

Latency for the n-th sentence pair

$$L(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{1}{Z(\boldsymbol{x}, \hat{\boldsymbol{y}})} \sum_i C_i(\boldsymbol{x}, \hat{\boldsymbol{y}})$$

- Z: Normalization function for target positions
- C_i a cost function for each target position i

Latency for the evaluation set

- Average of the latencies of each sentence pair

MLLP

Cost function

$$C_i(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \begin{cases} g(i) & \text{AP} \\ g(i) - \frac{i-1}{\gamma} & \text{AL} \\ g'(i) - \frac{i-1}{\gamma} & \text{DAL} \end{cases} \tag{1}$$

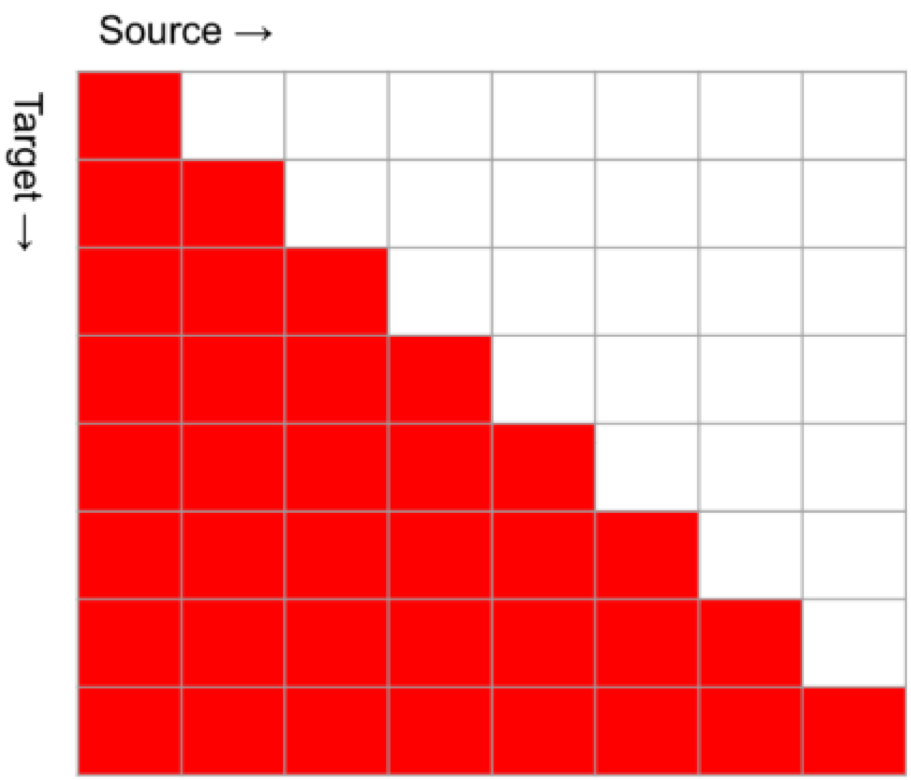
Normalization function

$$Z(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \begin{cases} |\boldsymbol{x}| \cdot |\hat{\boldsymbol{y}}| & \text{AP} \\ \arg\min\limits_{i: g(i) = |\boldsymbol{x}|} i & \text{AL} \\ |\hat{\boldsymbol{y}}| & \text{DAL} \end{cases} \tag{2}$$

MLLP

Average Proportion (AP)

$$L(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{1}{|\boldsymbol{x}| \cdot |\hat{\boldsymbol{y}}|} \sum_i {\color{red} g(i)}$$



$$L(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{36}{64} = 0.56$$

(Image source: [Huang et al., 2020])

Average Lagging (AL)

AL $\simeq$ Difference between model and a wait-0 oracle

$$L(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{1}{\underset{i:g(i)=|\boldsymbol{x}|}{\arg\min}\ i} \sum_i g(i) - \frac{i-1}{\gamma}$$

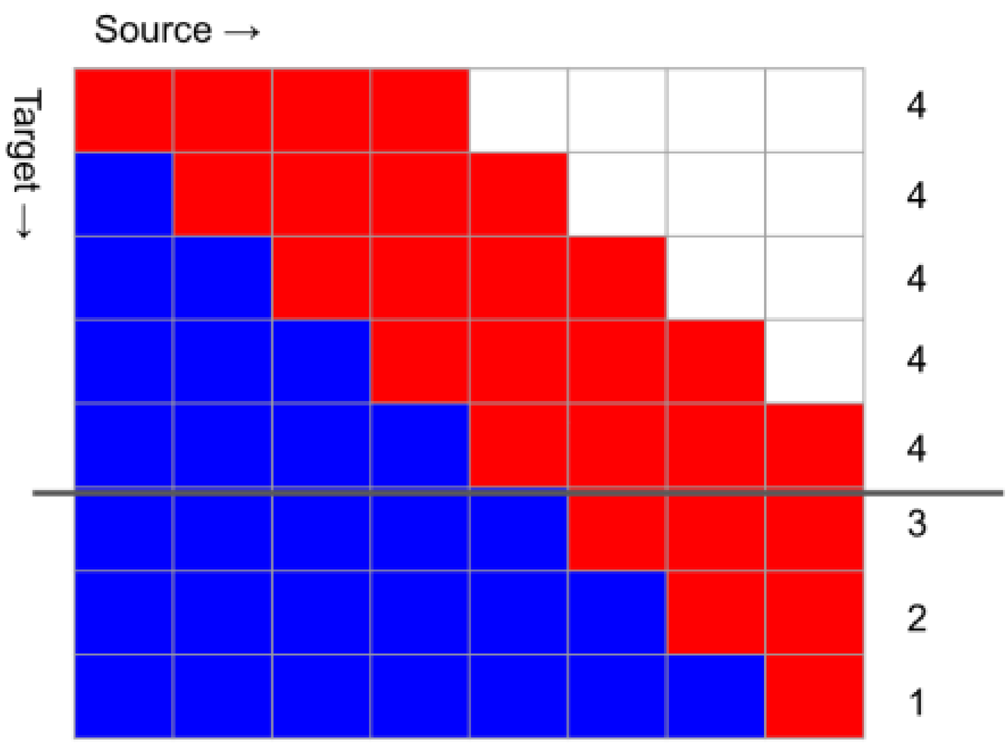
- $g(i)$: Policy of the model being evaluated
- $\frac{i-1}{\gamma}$: Policy of a wait-0 oracle
- $\underset{i:g(i)=|\boldsymbol{x}|}{\arg\min}\ i$: Stop when we have read $|x|$ tokens

MLLP

Average Lagging (AL)

$$L(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{1}{\underset{i:g(i)=|\boldsymbol{x}|}{\arg\min}\; i} \sum_i g(i) - \frac{i-1}{\gamma}$$

- Wait-0 oracle
- Model to be evaluated (wait-4 policy)



(Image source: [Huang et al., 2020])

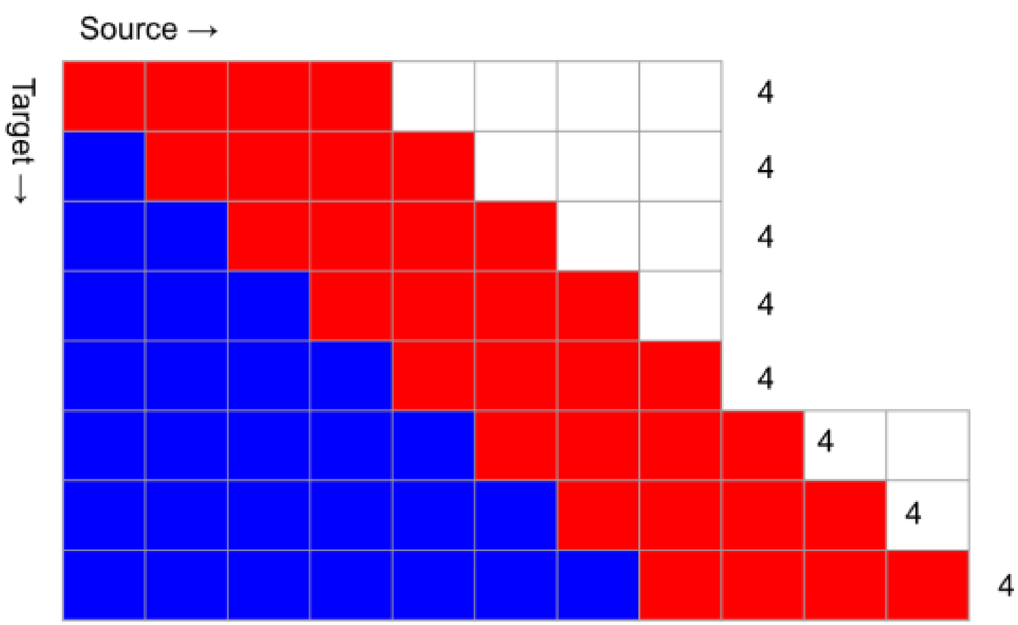
MLLP

Differentiable Average Lagging (DAL)

DAL $\simeq$ AL but write operations incur an additional $\frac{1}{\gamma}$ delay

$$L(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \frac{1}{|\hat{\boldsymbol{y}}|} \sum_i g'(i) - \frac{i-1}{\gamma}$$

$$g'(i) = \max \begin{cases} g(i) \\ g'(i-1) + \frac{1}{\gamma} \end{cases} \tag{3}$$



3 Streaming MT Evaluation

Are current practices for simultaneous MT evaluation realistic?

Some problematic aspects

- Sentences are evaluated in isolation
  - Delays do have an effect on follow-up sentences
- Fixed segmentation must be used to compare systems
- Evaluated with short segments (MuST-C $\simeq$ 4.8s segments)
  - Is simultaneous MT even required for this scenario?

Proposed approach

- Evaluate latency of the entire stream to be translated

MLLP

Simultaneous Translation Evaluation: Previous work

Concat-1[Schneider and Waibel, 2020]

- Concat all text into a single sentence, translate & evaluate

Drawbacks

- This assumes a constant writing rate (γ) for the entire stream
- Is this realistic?

Concat 1 - Example

- Translate two sentences with a wait-1 system with $\gamma = \gamma_n$
- $|\boldsymbol{x}_1| = 2$, $|\hat{y}_1| = 2$, $\gamma_1 = 1$

i	1	2
$g(i)$	1	2

- $|\boldsymbol{x}_2| = 2$, $|\hat{y}_2| = 4$, $\gamma_2 = 2$

i	1	2	3	4
$g(i)$	1	1	2	2

- Compute:
  - Standard sentence-level metrics ($\gamma_1 = 1$, $\gamma_2 = 2$)
  - Concat-1 metrics ($\gamma = \frac{3}{2}$)
- Expectation: AL/DAL $\simeq$ 1

MLLP

Concat 1 - Example

- $|\boldsymbol{x}_1| = 2$, $|\hat{y}_1| = 2$, $|\gamma_1| = 1$
- $|\boldsymbol{x}_2| = 2$, $|\hat{y}_2| = 4$, $|\gamma_2| = 2$

									L
Ind. Sent.		i	1	2	1	2	3	4	
		$g(i)$	1	2	1	1	2	2	
		$\frac{i-1}{\gamma}$	0.0	1.0	0.0	0.5	1.0	1.5	
		AP	1	2	1	1	2	2	**0.8**
	C_i	AL	1	1	1	0.5	1	-	**0.9**
		DAL	1	1	1	1	1	1	**1.0**
Concat-1		i	1	2	3	4	5	6	
		$g(i)$	1	2	3	3	4	4	
		$\frac{i-1}{\gamma}$	0	0.6	1.3	2.0	2.6	3.3	
		AP	1	2	3	3	4	4	**0.7**
	C_i	AL	1	1.3	1.6	1	1.3	-	**1.2**
		DAL	1	1.3	1.6	1.6	1.6	1.6	**1.5**

Concat 1 - Cont.

When Concat-1 is computed for standard evaluation sets:

- AP $\rightarrow 0.5$

- AL and DAL do not reflect real behaviour of the model
  - Oracle writing speed is always under/over-estimated

- DAL grows larger and larger due to accumulating write delays

- System ranking is altered, not interpretable

Wait-k	1	2	3	4	5
AL	-9.7	-12.0	-45.2	-23.7	-8.5

- Streaming evaluation is unfeasible with a single, fixed oracle γ

MLLP

Our proposal

Stream-level Latency Evaluation for Simultaneous Machine Translation [Iranzo-Sánchez et al., 2021]

- Key idea: Need local (sentence-like) estimation of γ, γ_n
- Keep track of latency with a global delay, $G(i')$
  - Like in Concat-1
- Convert $G(i')$ to local representation and check with local oracle
- Accurate metrics if we obtain good local representation & oracle

Our proposal

$G(i')$: # stream src words available for writing i'-th tgt stream word

$$C_i(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n) = \begin{cases} g_n(i) & \text{AP} \\ g_n(i) - \frac{i-1}{\gamma_n} & \text{AL} \\ g'_n(i) - \frac{i-1}{\gamma_n} & \text{DAL} \end{cases}$$

- What is the global index of the i-th word of the n-th sentence?
  - $G(i + |\hat{\boldsymbol{y}}_1^{n-1}|)$

$$\underbrace{g_n(i)}_{\text{Local delay}} = \underbrace{G(i + |\hat{\boldsymbol{y}}_1^{n-1}|)}_{\text{Global delay}} - \underbrace{|\boldsymbol{x}_1^{n-1}|}_{\text{Local operator}}$$

Segmentation

- We need sentence-level alignment for the evaluation
  - For local operator
  - For local oracle
  - For computing the metrics
- Do as for quality evaluation: Re-align sentences with the ref.
  - Minimum edit distance: MWER [Matusov et al., 2005]
- After re-alignment, we obtain pairs $(\boldsymbol{x}_n, \hat{\boldsymbol{y}}_n)$
  - Then, compute local variables

AL Results

- Train data: IWSLT2020 En$\leftrightarrow$De except OpenSubtitles
- Eval data: IWSLT2010 De$\rightarrow$En
- 1 system + 3 oracles:
  - Real: DS segmenter + Wait-k system
  - + In. Seg: Use ref. input segmentation instead of DS
  - + Out. Seg: Use ref. output segmentation instead of MWER
  - + Policy : Use oracle γ_n for each sentence

AL Results

Concat-1

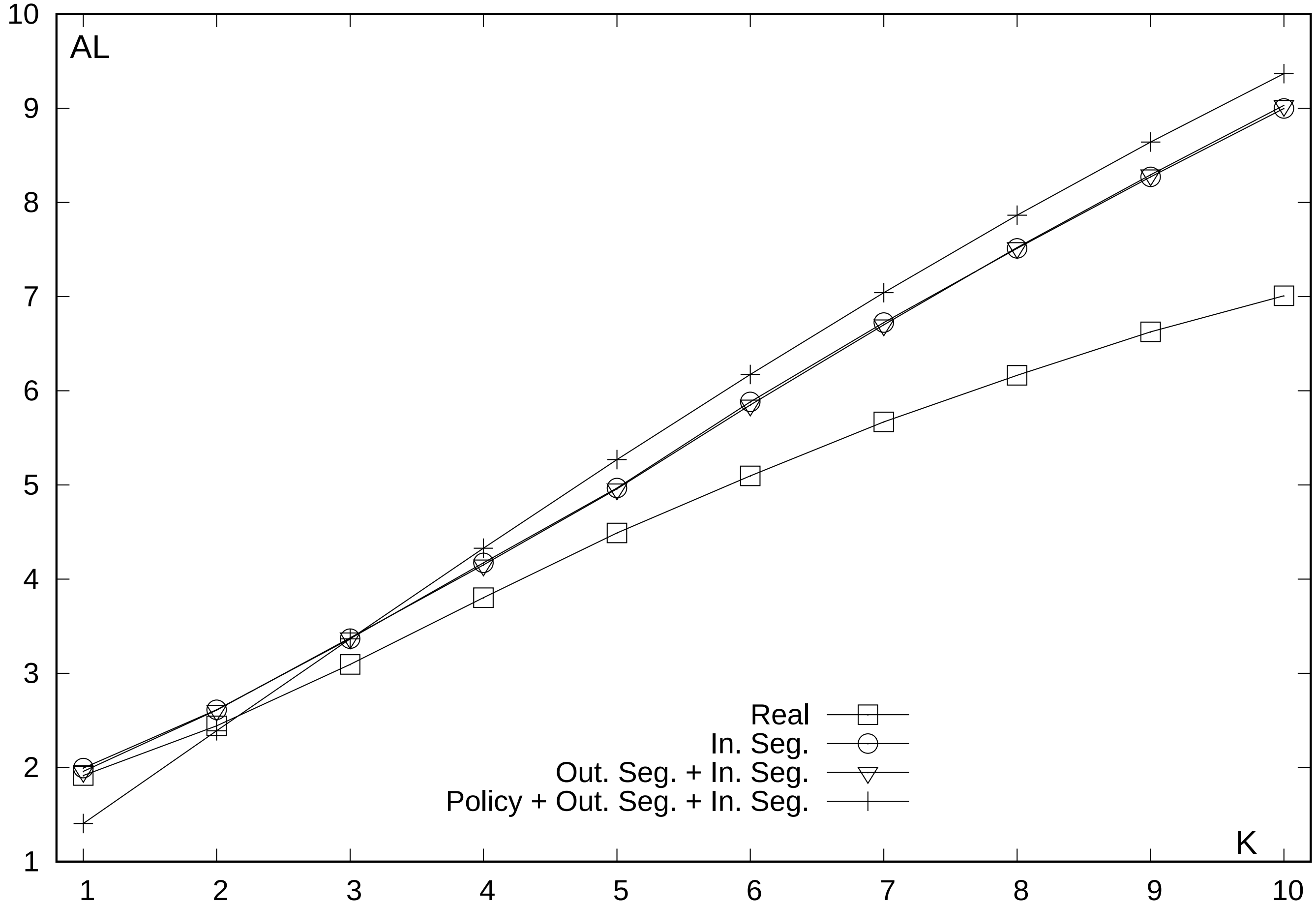
	Wait-k				
System	1	2	3	4	5
Real	-9.7	-12.0	-45.2	-23.7	-8.5
+In. Seg.	-42.9	-29.0	17.4	-10.1	25.5
+ Policy	14.2	15.1	16.0	16.8	17.6

- The ranking of the systems is altered
- The results are not interpretable

MLLP

AL Results(cont.)

Proposed approach



- Results ranked by increasing order of k
- Interpretable and accurate results

4 Streaming MT: Models & Baseline

From Simultaneous to Streaming Machine Translation by Leveraging Streaming History [Iranzo-Sánchez et al., 2022]

- Stream(ing) MT consists in:
  - Real-time translation
  - Translation of an unbounded stream

- Translate an input stream X into a target stream Y

Streaming MT

- Global delay $G(i)$

$$\hat{Y}_i = \arg\max_{y \in \mathcal{Y}} p\left(y \mid X_1^{G(i)}, Y_1^{i-1}\right)$$

- For efficiency, we introduce the history function $H(i)$

$$\hat{Y}_i = \arg\max_{y \in \mathcal{Y}} p\left(y \mid X_{G(i)-H(i)+1}^{G(i)}, Y_{i-H(i)}^{i-1}\right)$$

- Translate using sliding windows defined by $G(i)$ and $H(i)$

MLLP

Streaming MT Baseline

Segmentation

- a_n: Starting position of n-th source sentence
- b_n: Starting position of n-th target sentence
- $|\mathbf{a}| = |\mathbf{b}| = N$

MLLP

Streaming MT Baseline

Policy

- Simultaneous (sentence-level) wait-k:

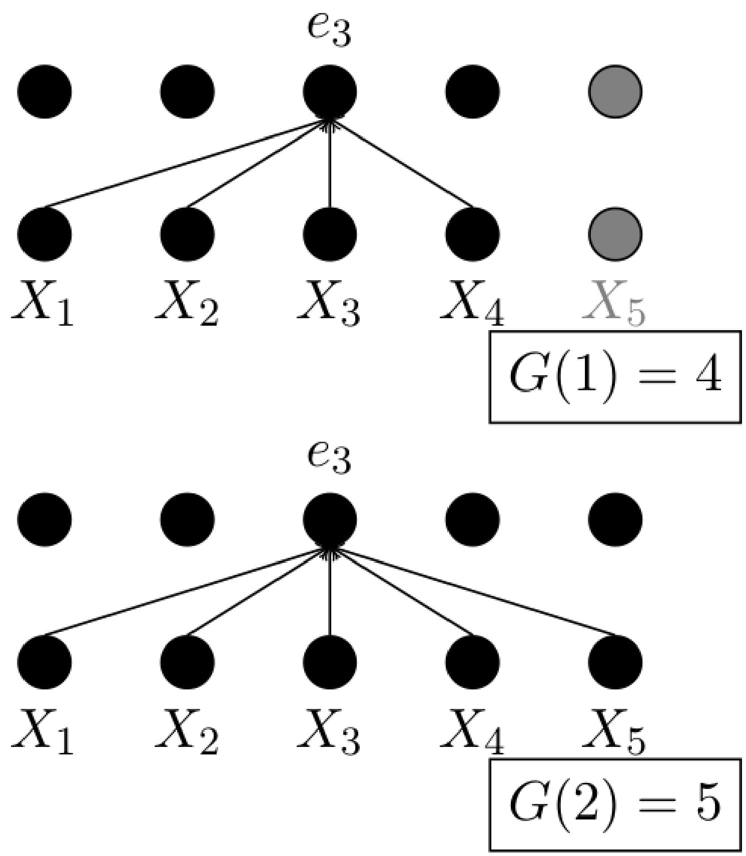
$$g(i) = \left\lfloor k + \frac{i-1}{\gamma} \right\rfloor$$

- Streaming MT wait-k:

$$G(i) = \left\lfloor k + \frac{i-b_n}{\gamma} \right\rfloor + a_n - 1$$

- $b_n \leq i < b_{n+1}$

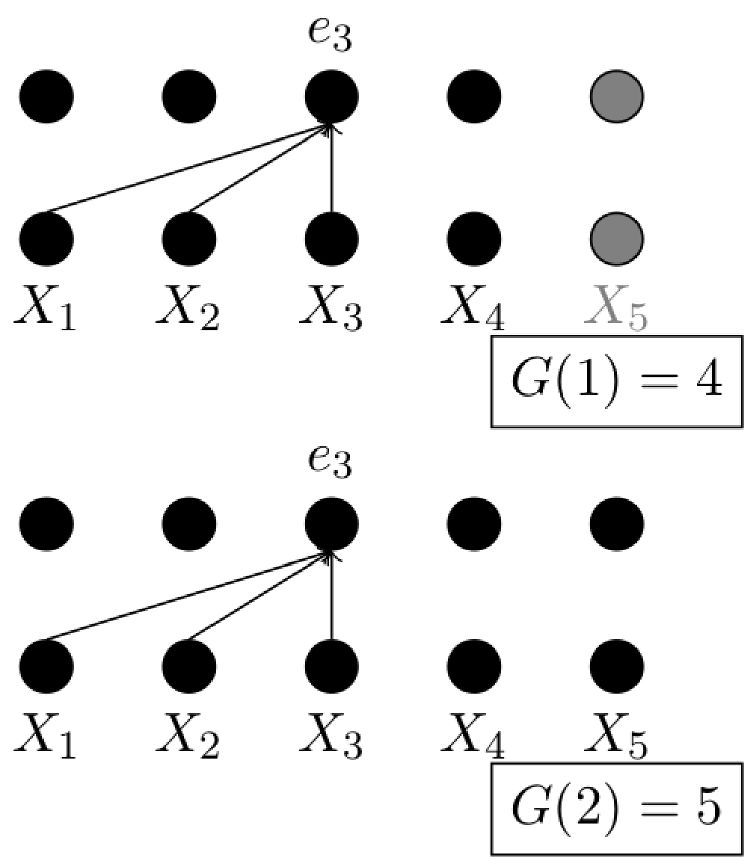
Streaming MT Baseline: Encoders



Bidirectional - Standard MT

$$e_j^{(l)} = \mathrm{Enc}\left(e^{(l-1)}_{G(i)-H(i)+1:G(i)}\right)$$

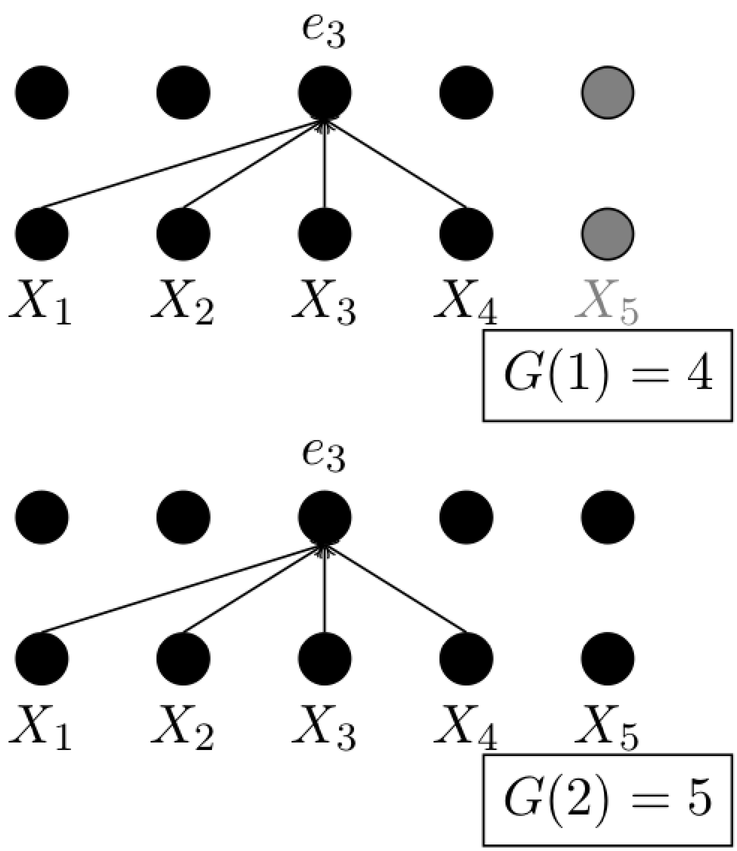
Streaming MT Baseline: Encoders



- Unidirectional - [Ma et al., 2019, Elbayad et al., 2020]

$$e_j^{(l)} = \mathrm{Enc}\left(e_{G(i)-H(i)+1:j}^{(l-1)}\right)$$

Streaming MT Baseline: Encoders



Partial Bidirectional Encoder (PBE) - This work

$$e_j^{(l)} = \mathrm{Enc}\left(e^{(l-1)}_{G(i)-H(i)+1:\max(G(i)-H(i)+k,j)}\right)$$

Streaming MT Baseline: System training

Sentence pair	Source	Target
1	$x_{1,1}\ x_{1,2}$	$y_{1,1}\ y_{1,2}$
2	$x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}$	$y_{2,1}\ y_{2,2}\ y_{2,3}$
3	$x_{3,1}\ x_{3,2}\ x_{3,3}$	$y_{3,1}\ y_{3,2}\ y_{3,3}$

Sample	Source
1	[DOC] $x_{1,1}\ x_{1,2}$ [BRK]
2	[DOC] $x_{1,1}\ x_{1,2}$ [SEP] $x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}$ [BRK]
3	[CONT] $x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}$ [SEP] $x_{3,1}\ x_{3,2}\ x_{3,3}$ [BRK]

Sample	Target
1	[DOC] $y_{1,1}\ y_{1,2}$ [BRK]
2	[DOC] $y_{1,1}\ y_{1,2}$ [SEP] $y_{2,1}\ y_{2,2}\ y_{2,3}$ [BRK]
3	[CONT] $y_{2,1}\ y_{2,2}\ y_{2,3}$ [SEP] $y_{3,1}\ y_{3,2}\ y_{3,3}$ [BRK]

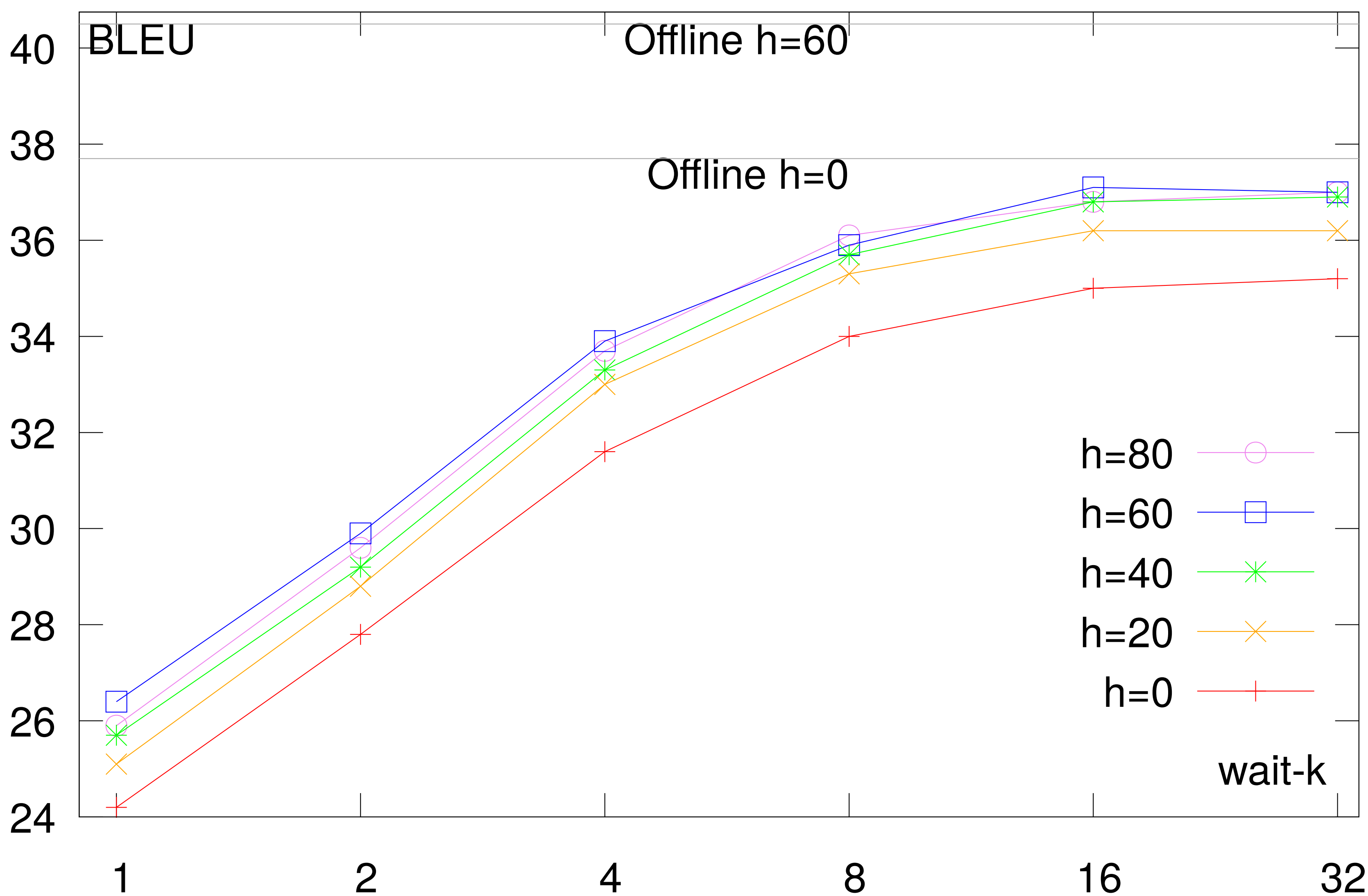
Experiments: Setup

- Train data: IWSLT2020 En$\leftrightarrow$De except OpenSubtitles
- Eval data
  - IWSLT2010 De$\rightarrow$En
  - IWSLT2020 En$\rightarrow$De
- Finetune on MuST-C train
  - Same setup as [Schneider and Waibel, 2020]
- Transformer Big model, 40k BPE subwords
- Multi-path wait-k policy [Elbayad et al., 2020]

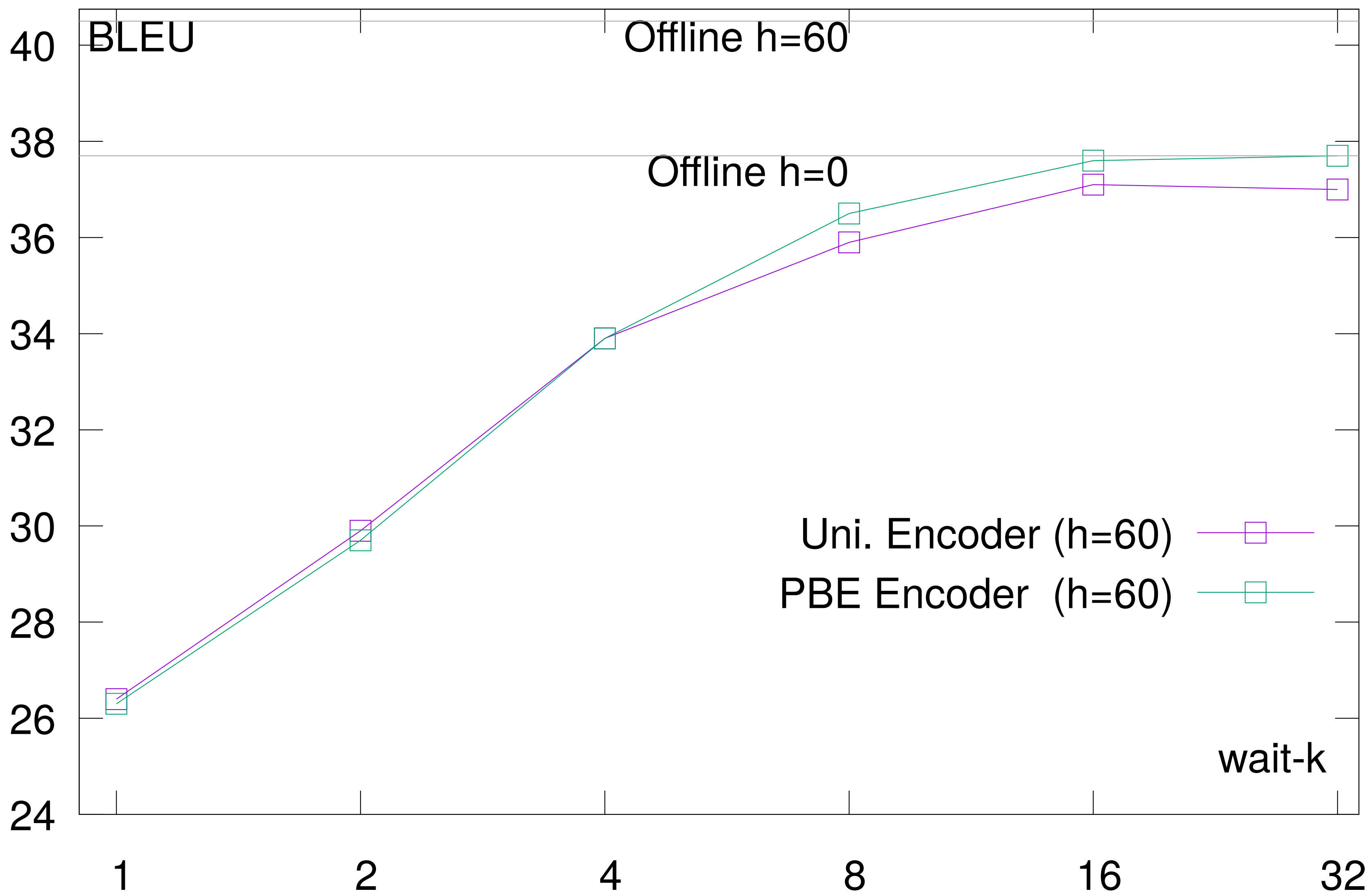
MLLP

Experiments: Streaming history size

IWSLT 2010 Dev (De $\rightarrow$ En)



Experiments: PBE Encoder



Experiments: Comparison with SoTA

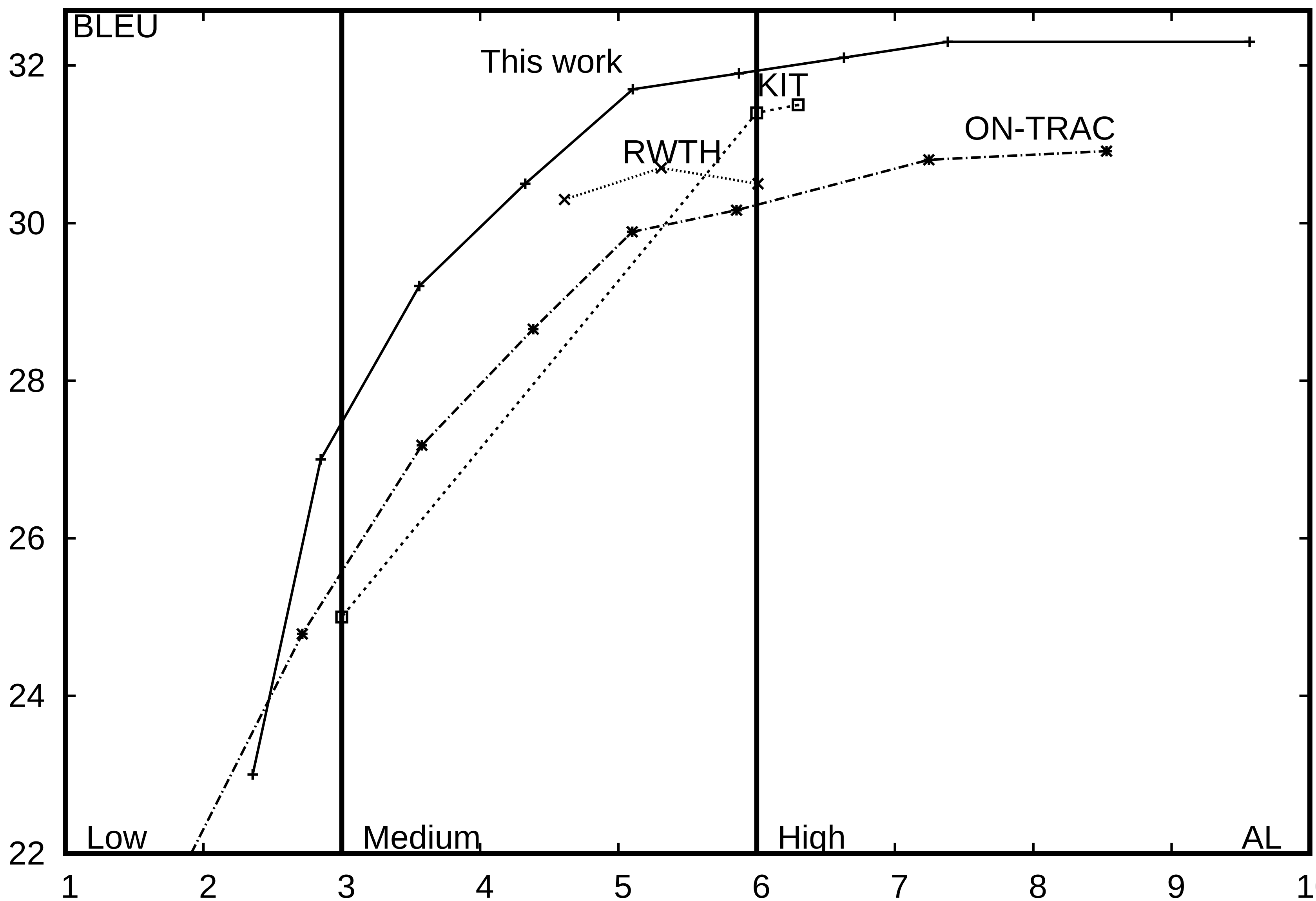
Streaming MT, IWSLT 2010

Model	BLEU	AP	AL	DAL
ACT [Schneider and Waibel, 2020]	30.3	10.3	100.1	101.8
This work	29.5	1.2	11.2	17.8

- Latency measured with streaming AP/AL/DAL [Iranzo-Sánchez et al., 2021]
- Similar performance with a fraction of the latency
- Adaptive policy of ACT falls behind (no catch-up mechanism)
- Wait-k + segmenter ensure model keeps up with the speaker

Experiments: Comparison with SoTA

IWSLT 2020 En→De: MuST-C tst-COMMON



Conclusions

- Need for realistic Simultaneous MT evaluation
- Proposed techniques for Streaming evaluation
- Proposed Streaming MT system has significant quality gains

Thanks for your attention!

Full details available in the papers

Code for segmenter/MT: `https://github.com/jairsan`

References

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. of Interspeech*, pages 1461–1465, 2020.

Liang Huang, Colin Cherry, Mingbo Ma, Naveen Arivazhagan, and Zhongjun He. Simultaneous translation. In *Proc. of EMNLP: Tutorial Abstracts*, pages 34–36. Association for Computational Linguistics, 2020.

Javier Iranzo-Sánchez, Adrià Giménez, Joan Albert Silvestre-Cerdà, Pau Baquero, Jorge Civera, and Alfons Juan. Direct Segmentation Models for Streaming Speech Translation. In *Proc. of EMNLP*, pages 2599–2611. ACL, 2020.

Javier Iranzo-Sánchez, Jorge Civera Saiz, and Alfons Juan. Stream-level latency evaluation for simultaneous machine translation. In *Findings of ACL: EMNLP*, pages 664–670. ACL, 2021.

Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. From simultaneous to streaming machine translation by leveraging streaming history. In *Proc. of ACL*, 2022.

Javier Jorge, Adrià Giménez, Joan Albert Silvestre-Cerdà, Jorge Civera, Albert Sanchis, and Juan Alfons. Live streaming speech recognition using deep bidirectional lstm acoustic models and interpolated language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:148–161, 2021. doi: 10.1109/TASLP.2021.3133216.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proc. of ACL*, pages 3025–3036. ACL, 2019.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. Evaluating machine translation output with automatic sentence segmentation. In *Proc. of IWSLT*. ISCA, 2005.

Felix Schneider and Alexander Waibel. Towards stream translation: Adaptive computation time for simultaneous machine translation. In *Proc. of IWSLT*, pages 228–236. ACL, 2020.