# Average Token Delay: A Latency Metric for Simultaneous Translation

*Yasumasa Kano*[1]*, Katsuhito Sudoh*[1]*, Satoshi Nakamura*[1]

[1]Nara Institute of Science and Technology, Japan

{kano.yasumasa.kw4, sudoh, s-nakamura}@is.naist.jp

## Abstract

In simultaneous translation, translation begins before the speaker has finished speaking. In its evaluation, we have to consider the latency of the translation in addition to the quality, with latency preferably as small as possible for users to comprehend what the speaker says with a small delay. Existing latency metrics focus on when the translation starts but do not consider adequately when the translation ends. This means such metrics do not penalize the latency caused by a long translation output, which delays user comprehension. In this work, we propose a novel latency evaluation metric called *Average Token Delay* (ATD) that focuses on the end timings of partial translations in simultaneous translation. We discuss the advantage of ATD using simulated examples and investigate the differences between ATD and Average Lagging with simultaneous translation experiments.

**Index Terms**: simultaneous translation, latency evaluation

## 1. Introduction

Machine translation (MT) has evolved rapidly using recent neural network techniques and is now widely used both for written and spoken languages. MT for speech is a very attractive application for translating various conversations, lecture talks, etc. For smooth real-time speech communication across languages, speech MT should run in real-time and incrementally without waiting for the end of an input utterance as in consecutive interpretation. While the translation quality can improve by waiting for later inputs as the context, it should result in longer latency. This quality-latency trade-off is the most important issue in simultaneous MT.

Most recent simultaneous MT studies use BLEU [1] for evaluating the quality and Average Lagging [2] for the latency. AL is based on the number of input words that have become available when starting a translation and measures its average over all the timings of generating partial translations. It is very suitable for wait-$k$ [2] that waits for $k$ input tokens before starting the translation and then repeats to read and write one token alternately.

This work focuses on a problem of AL. AL does not sufficiently consider the cases where chunk-level outputs are generated at a time and give smaller latency values for them than one-by-one cases like wait-$k$. In addition, AL has a critical flaw that AL can be negative when the chunk-level outputs become longer, although the latency should not be negative.

Suppose we have a pair of seven input tokens and seven output tokens and apply two different simultaneous policies to this pair as shown in Figure 1: wait-3 and *chunk-3* that writes three tokens after reading three tokens. Figure 2 illustrates a situation where one input or output token requires one unit time to speak, ignoring computation time for simplicity. These policies are equivalent to each other, from the viewpoint of latency in this situation. However, AL for wait-3 is $\frac{15}{5} = 3$ and that for chunk-3 is $\frac{13}{7} \approx 1.857$; the equation of AL is explained in Section 2. AL tends to give a small latency value for such a long chunk-level output. However, a long output delays the start of the translation of later parts and makes the listener feel the delay. This problem becomes more severe when we need speech outputs because the speech outputs must be sequential. Therefore, the length of a translation output has a large effect on delay and should be included in the latency measurement. This observation suggests the need of another latency metric to cope with such situations.

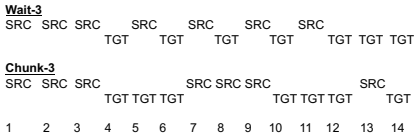



Figure 1: *wait-3 and chunk-3 in a step-wise view.*

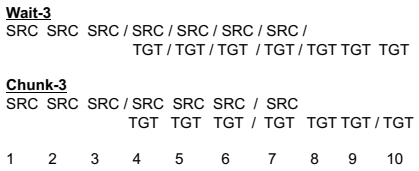



Figure 2: *wait-3 and chunk-3 in a time-synchronous view.*

We propose a novel latency metric called *Average Token Delay* (ATD) [1] that focuses on the end timings of partial translations. ATD generalizes the latency measurement for simultaneous translation both with speech and text outputs and works intuitively for chunk-based outputs that are not properly handled by AL as presented above. We present some simulation results to show the characteristics of ATD clearly and also demonstrate its effectiveness through simultaneous translation experiments.

## 2. Existing Latency Metric for Simultaneous Translation

Gu et al.[3] proposed a latency metric called Consecutive Wait (CW) to measure the local delay. Since CW is not originally an average measure of the latency, Ma et al.[2] used a step-wise CW. The average CW result in the same value for the follow-

[1]ATD will be implemented in https://github.com/facebookresearch/SimulEval

ing two extreme cases: (1) all the output tokens are generated after reading all input tokens and (2) all the output tokens are generated after reading only the first input token. Cho et al.[4] proposed Average Proportion (AP). AP differs with the change of the sequence length for the same simultaneous translation policy and is not intuitive for the latency metric.

Ma et al.[2] proposed Average Lagging (AL). AL is denoted as follows:

$$AL_g(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\tau_g(|\boldsymbol{x}|)} \sum_{\tau=1}^{\tau_g(|\boldsymbol{x}|)} \left( g(\tau) - \frac{\tau - 1}{r} \right). \quad (1)$$

$\boldsymbol{x}$ is the input sentence and $\boldsymbol{y}$ is its translation. $g(\tau)$ is a monotonic non-decreasing function representing the number of source words read to predict the $\tau$-th target word. r is the length ratio defined as $|\boldsymbol{y}|/|\boldsymbol{x}|$. $\tau_g(|\boldsymbol{x}|)$ is the cut-off step defined as follows:

$$\tau_g(|\boldsymbol{x}|) = \min\{\tau \mid g(\tau) = |\boldsymbol{x}|\} \quad (2)$$

meaning the index of the output token predicted right after the observation of the entire source sentence.

AL solves the problem of AP mentioned above by focusing on the difference from the *ideal* policy based on the length ratio. However, AL still suffers from unintuitive latency measurement because AL can be negative when the model finishes the translation before reading the entire input due to the subtraction term. Ma et al.[5] modified AL by changing the calculation of the length ratio $r$ based on the length of the reference translation. Papi et al.[6] proposed Length-Adaptive Average Lagging (LAAL) that uses the longer one between the reference and output. These modifications do not completely avoid negative values when the model generates a long partial output faster than the ideal policy. Arivazhagan et al.[7] proposed another variant, Differentiable Average Lagging (DAL), for optimizing simultaneous translation model. Iranzo-Sánchez et al.[8] proposed a method to calculate AL for a streaming input in a segmentation-free manner.

## 3. Proposed Metric: Average Token Delay

We propose a novel latency metric called Average Token Delay (ATD). We start from the latency measurement using ATD in case of simultaneous speech-to-speech MT and then generalize it for speech-to-text and text-to-text cases.

### 3.1. ATD for simultaneous speech-to-speech translation

Figure 3 illustrates a step-by-step workflow of simultaneous speech-to-speech MT. The white boxes represent fixed-length speech segments, the orange ones represent the processing time to encode prefix inputs and to judge whether prefix translations can be predicted there, and the blue ones represent decoding time. Intuitively, ATD is the average time difference between the points of the same color (black, red, and white).

Suppose we have an input $\boldsymbol{x}$ and an output $\boldsymbol{y}$ segmented into $C$ chunks $\boldsymbol{x} = \boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^C$ and $\boldsymbol{y} = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^C$, respectively. An input chunk $\boldsymbol{x}^c$ is what is observed after the previous judgment for the prefix translation $\boldsymbol{y}^{c-1}$ and is used to predict $\boldsymbol{y}^c$. In the case of text, each chunk consists of sub-segments with words, subwords, or characters. In the case of speech, we divide each chunk into sub-segments of 0.3 seconds long from the beginning of the chunk, assuming one word is uttered in 0.3 seconds. Through this segmentation, the input and output sentences can be represented as a series of sub-segments $\boldsymbol{x} = x_1, ..., x_{|\boldsymbol{x}|}$, $\boldsymbol{y} = y_1, ..., y_{|\boldsymbol{y}|}$ respectively.

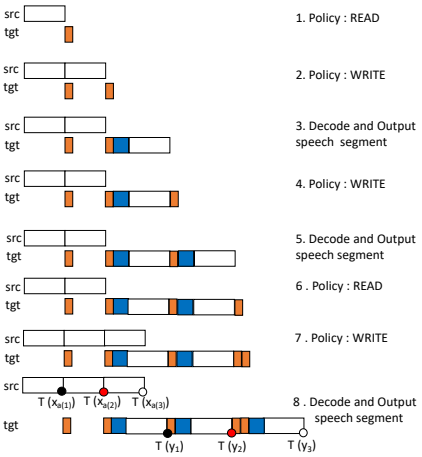



Figure 3: *Step-by-step example of simultaneous speech-to-speech MT*

ATD is defined as follows:

$$\mathrm{ATD}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{|\boldsymbol{y}|} \sum_{t=1}^{|\boldsymbol{y}|} \left( T(y_t) - T(x_{a(t)}) \right) \quad (3)$$

where

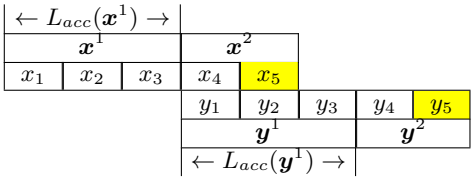
$$a(t) = \begin{cases} s(t) & \text{if } s(t) \leq L_{acc}(\boldsymbol{x}^{c(t)}) \\ L_{acc}(\boldsymbol{x}^{c(t)}) & \text{otherwise} \end{cases} \quad (4)$$

$$s(t) = t - \max(L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}), 0) \quad (5)$$

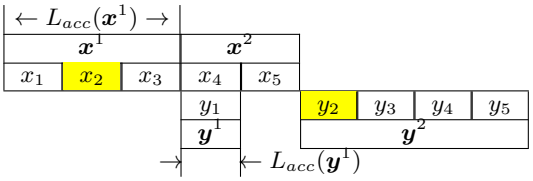
$T(\cdot)$ in Eq. (3) represents the ending time of each token, which is shown as colored points in Figure 3. $a(t)$ represents the index of the input token corresponding to $y_t$. $L_{acc}(\boldsymbol{x}^c) = \sum_{j=1}^{c} |\boldsymbol{x}^j|$ is the cumulative length up to the $c$-th chunk, and $L_{acc}(\boldsymbol{x}^0) = 0$. $L_{acc}(\boldsymbol{y}^c)$ is defined similarly. $c(t)$ denotes the chunk number $c$ to which $y_t$ belongs. As shown in Eq. (5), if the previous translation prefix is longer than the previous input prefix, $s(t)$ becomes smaller than the output index $t$, which means the previous long output makes the time difference between the input token and the corresponding output token larger. ATD is guaranteed not to become negative in any conditions due to the nature of these equations.

ATD is the average delay of output sub-segments against their corresponding input sub-segments, considering the latency required for inputs and outputs. Although the input-output correspondence does not necessarily mean semantic equivalence, especially for language pairs with large differences in their word order and the numbers of tokens, we use this simplified formulation for the latency measurement as same as AL.

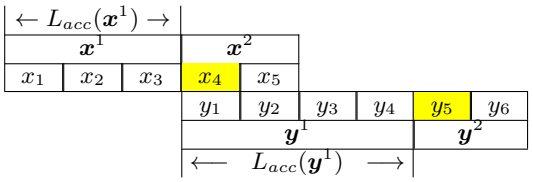
Figure 4 shows examples to explain the term in max operator in Eq. (5). In Figure 4a, Suppose we measure the token delay on $y_5$. $\boldsymbol{y}_5$ is in the second output chunk, so $c(5) = 2$. Since $L_{acc}(\boldsymbol{y}^1) = L_{acc}(\boldsymbol{x}^1) = 3$, we obtain $a(5) = s(5) = 5 - 0 = 5 \leq L_{acc}(\boldsymbol{x}^2) = 5$, and therefore $\boldsymbol{y}_5$ corresponds to $\boldsymbol{x}_5$. In Figure 4b, Suppose we measure the token delay on $y_2$. $y_2$ is in the second output chunk, so c(2) = 2. Since $L_{acc}(\boldsymbol{y}^1) - L_{acc}(\boldsymbol{x}^1) = 1 - 3 < 0$, we obtain $a(2) = s(2) = 2 - 0 = 2 \leq L_{acc}(x^2) = 5$, therefore $y_2$ corresponds to $x_2$. In Figure 4c, the first output chunk is longer: $L_{acc}(\boldsymbol{y}^1) = 4$. This results in $a(5) = s(5) = 5 - 1 = 4 \leq L_{acc}(\boldsymbol{x}^2) = 5$, so $\boldsymbol{y}_5$ corresponds to $\boldsymbol{x}_4$.




(a) $L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}) = 0$ $(t = 5)$

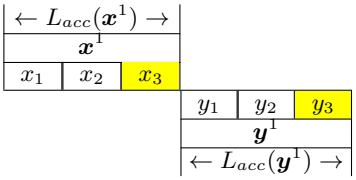



(b) $L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}) < 0$ $(t = 2)$

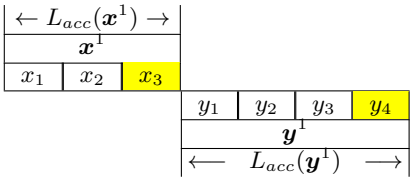



(c) $L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}) > 0$ $(t = 5)$

Figure 4: *Examples for the explanation of Eq. (5)*




(a) $s(t) \leq L_{acc}(\boldsymbol{y}^{c(t)})$ $(t = 3)$




(b) $s(t) > L_{acc}(\boldsymbol{y}^{c(t)})$ $(t = 4)$

Figure 5: *Example for the explanation of Eq. (4)*

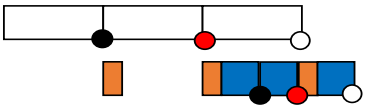


Figure 6: *Summary view for latency measurement for simultaneous speech-to-text translation*

Figure 5 shows examples to explain Eq. (4). In Figure 5a, we measure the token delay on $y_3$. Here, we obtain $s(t) = 3 \leq L_{acc}(\boldsymbol{x}^1) = 3$, so $y_3$ corresponds to $x_3$. Figure 5b, we measure the token delay on $y_4$. Here, we obtain $s(t) = 4 > L_{acc}(\boldsymbol{x}^1) = 3$, so $y_4$ corresponds to $x_3$.

### 3.2. ATD for simultaneous {speech,text}-to-text translation

Figure 6 illustrates the latency measurement for speech-to-text simultaneous translation, where the output duration can be ignored. Figure 7 is the one for text-to-text simultaneous translation. We reserve input duration here because input for text-to-text simultaneous translation comes from speech via automatic speech recognition (ASR), in most cases. The input duration reflects ASR computation time.

### 3.3. Non-computation-aware ATD

We sometimes use the latency measurement independent of the computation time for estimating ideal situations that are not influenced by the performance of computers and the efficiency of implementations. In Figures 3, 6, and 7, we remove the orange, blue and yellow parts and only include the duration of speech segments to calculate delay. However, this means all the terms in text-to-text translation become 0. We follow the conventional step-wise latency measurement as CW and AP by letting each input and output word spend one step as shown in Figure 8. Also, we assume the model can read the next input and output the partial translation in parallel as shown in Figure 2.

## 4. Simulation

Before presenting the latency measurement experiments using real data, we show simulation results comparing AL and ATD in different conditions in simultaneous text-to-text translation.

### 4.1. Comparison of Wait-k and Chunk-k

We assume that the numbers of tokens of the input and output are both 40. We compare the latency measurement by AL and ATD where the hyperparameter $k$ for wait-$k$ and chunk-$k$ varies from 1 to 40. Here, for simplicity, we assume the length of input and output chunks are the same for chunk-$k$ until the prediction of the end-of-sentence token.

Figure 9 indicates the gap between wait-$k$ and chunk-$k$ by AL mentioned in section 1, while ATD results in the same values for them as shown in Figure 10. One serious problem raised here is the large jump in AL at $k = 40$. For example, in the case of chunk-39, AL uses $r = 1$, $\tau_{g_{\text{chunk-39}}}(|\boldsymbol{x}|) = 40$, $g_{\text{chunk-39}}(\tau) = 39$ $(1 \leq \tau \leq 39)$, and $g_{\text{chunk-39}}(\tau) = 40$ $(\tau = 40)$. Then the AL value becomes $\frac{1}{40}\left\{\left(\sum_{\tau=1}^{39} \tau\right) + 1\right\} = \frac{781}{40} = 19.525$. However, in the case of chunk-40, $g_{\text{chunk-40}}(\tau) = 40$ for all $\tau$ and $\tau_{g_{\text{chunk-40}}}(|\boldsymbol{x}|) = 1$ according to Equation 2. As a result, the AL value becomes 40. This phenomenon comes from the definition of the cut-off step described in Section 4.2 by Ma et al.[2], where they assume later outputs derive no further delays.

### 4.2. Translation example

Figure 11 shows examples of chunk-based simultaneous translation for the input *I bought a pen.* by three different models.

Model 1 waits for the whole input sentence and results in the largest delay and the highest translation quality.

Model 2 has a smaller delay than Model 1 because it can segment the input after observing *I*. The segmentation enables the model to generate a partial translation but causes quality degradation due to the lack of context.

Model 3 works similarly to Model 2 for the input segmentation but outputs a longer segment for *I*. This causes large quality degradation by over-translation.

Regarding the latency of these three models, AL is the

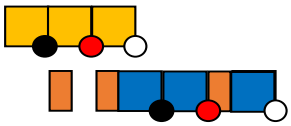


Figure 7: *Summary view for latency measurement for simultaneous text-to-text translation*

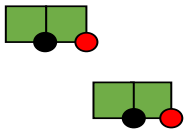


Figure 8: *Summary view for non-computation-aware latency measurement for simultaneous text-to-text translation*

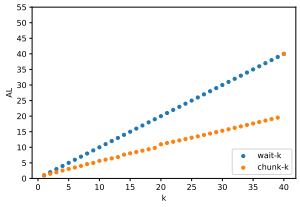



Figure 9: *Latency measurement by AL*

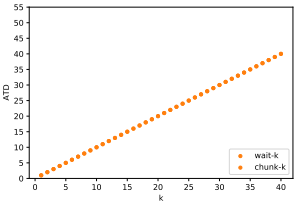



Figure 10: *Latency measurement by ATD*

smallest for Model 3 and decreases largely from Model 1 to Model 2; they are not intuitive as discussed so far. In contrast, ATD results in reasonable latency values considering the delay caused by the outputs.

## 5. Analyses

We conducted analyses on actual simultaneous translation results to investigate the effectiveness of ATD. Note that the analyses here were conducted on simultaneous text-to-text translation for simplicity.

### 5.1. Data

We used the data from the IWSLT evaluation campaign for English-to-Geman simultaneous translation. We used WMT 2014 training set (4.5 M sentence pairs) for pre-training and IWSLT 2017 training set (206 K sentence pairs) for fine-tuning. The development set consists of dev2010, tst2010, tst2011 and tst2012 (5,589 sentence pairs in total), and the evaluation set is tst2015 (1,080 sentence pairs).

We compared wait-$k$ [2], Meaningful Unit [9], Incremental Constitute Label Prediction [10], and Prefix Alignment [11], following the the experimental settings in the literature [11].

### 5.2. Results

As shown in Figures 12 and 13, ATD demonstrated clear differences in delay among models compared to AL. MU and ICLP were affected by the change in the latency metric. We analyzed their results in detail and found this degradation was due to over-translation as suggested by the observations of length ratio results shown in Figure 14. This phenomenon is the same as what happened with Model 3 in subsection 4.2. MU and ICLP generated long translations exceeding the length ratio of 1.0 when they worked with small latency. One interesting finding here is the correlation between BLEU and ATD by MU; larger latency did not always result in better BLEU. It is because over-translation increases ATD, but decreases BLEU at the same time. In contrast, wait-$k$ is a strategy that generates

**Model 1**
ATD:5.4, AL:5.0, Quality: **1st**
I bought a pen . /
私 は ペン を 買っ た 。
1 2 3 4 5 6 7 8 9 10 11 12

**Model 2**
ATD:**3.4**, AL:1.6, Quality: 2nd
I / bought a pen . /
私 。 / ペン を 買っ た 。
1 2 3 4 5 6 7 8 9 10

**Model 3**
ATD:4.1, AL:**0.8**, Quality: 3rd
I / bought a pen . /
私 で ござい ます 。/ペン を 買っ た 。
1 2 3 4 5 6 7 8 9 10 11

Figure 11: *Translation example*

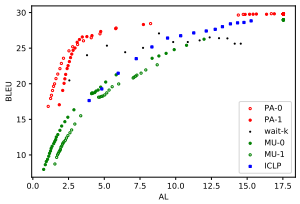



Figure 12: *Latency measurement by AL*

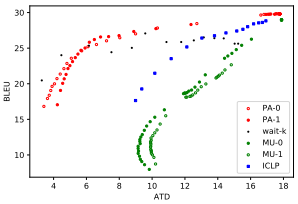



Figure 13: *Latency measurement by ATD*

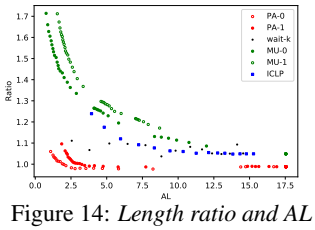



Figure 14: *Length ratio and AL*

one output token at a time and can avoid such an issue. PA also worked well with the latency measurement by ATD because it fine-tunes the translation model to avoid over-translation.

## 6. Conclusion

We proposed a novel latency metric ATD for simultaneous machine translation, which addresses the problem in the latency evaluation for a chunk-based model by taking the output length into account. ATD gives a large latency value to a long output based on the assumption that the output also causes a delay, different from AL. We revealed the effectiveness of ATD through the analyses of simulation and actual translation results compared with AL.

Future work includes studies on semantics-oriented latency measurement not just focusing on timing information without any consideration about the delivery of contents.

## 7. Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP21H03500.

# 8. References


[1] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[2] M. Ma, L. Huang, H. Xiong, R. Zheng, K. Liu, B. Zheng, C. Zhang, Z. He, H. Liu, X. Li, H. Wu, and H. Wang, "STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3025–3036. [Online]. Available: https://aclanthology.org/P19-1289

[3] J. Gu, G. Neubig, K. Cho, and V. O. Li, "Learning to translate in real-time with neural machine translation," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 1053–1062. [Online]. Available: https://aclanthology.org/E17-1099

[4] K. Cho and M. Esipova, "Can neural machine translation do simultaneous translation?" *arXiv preprint arXiv:1606.02012.*, 2016.

[5] X. Ma, M. J. Dousti, C. Wang, J. Gu, and J. Pino, "SIMULEVAL: An evaluation toolkit for simultaneous translation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 144–150. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.19

[6] S. Papi, M. Gaido, M. Negri, and M. Turchi, "Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation," in *Proceedings of the Third Workshop on Automatic Simultaneous Translation*. Online: Association for Computational Linguistics, Jul. 2022, pp. 12–17. [Online]. Available: https://aclanthology.org/2022.autosimtrans-1.2

[7] N. Arivazhagan, C. Cherry, W. Macherey, C.-C. Chiu, S. Yavuz, R. Pang, W. Li, and C. Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1313–1323. [Online]. Available: https://aclanthology.org/P19-1126

[8] J. Iranzo-Sánchez, J. Civera Saiz, and A. Juan, "Stream-level latency evaluation for simultaneous machine translation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 664–670. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.58

[9] R. Zhang, C. Zhang, Z. He, H. Wu, and H. Wang, "Learning adaptive segmentation policy for simultaneous translation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2280–2289. [Online]. Available: https://aclanthology.org/2020.emnlp-main.178

[10] Y. Kano, K. Sudoh, and S. Nakamura, "Simultaneous neural machine translation with constituent label prediction," in *Proceedings of the Sixth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2021, pp. 1124–1134. [Online]. Available: https://aclanthology.org/2021.wmt-1.120

[11] ——, "Simultaneous neural machine translation with prefix alignment," in *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Dublin, Ireland (in-person and online): Association for Computational Linguistics, May 2022, pp. 22–31. [Online]. Available: https://aclanthology.org/2022.iwslt-1.3