CA*: Addressing Evaluation Pitfalls in Computation-Aware Latency for Simultaneous Speech Translation

Anonymous ACL submission

Abstract

Simultaneous speech translation (SimulST) systems must balance translation quality with response time, making latency measurement crucial for evaluating their real-world performance. However, there has been a longstanding belief that current metrics yield unrealistically high latency measurements in unsegmented streaming settings. In this paper, we investigate this phenomenon, revealing its root cause in a fundamental misconception underlying existing latency evaluation approaches. We demonstrate that this issue affects not only streaming but also segment-level latency evaluation across different metrics. Furthermore, we propose a modification to correctly measure computation-aware latency for SimulST systems, addressing the limitations present in existing metrics.

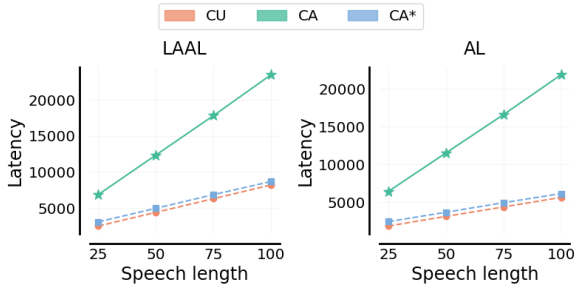

Figure 1: Computation-aware latency metrics tend to produce unrealistically high scores as speech duration increases. We segmented the original tst-COMMON dataset into 25s speech segments, then duplicated and concatenated them to create 50s, 75s, and 100s speech durations. The system used for evaluation is a wait-k-stride-n model, where n=3 and k=4, with each speech segment spanning 250 ms.

1 Introduction

Simultaneous speech-to-text translation (SimulST) (Ma et al., 2020b) focuses on a real-time, low-latency scenario where the model starts generating the textual translation before the entire audio input is processed. Achieving high-quality translations with minimal latency is the primary objective of SimulST systems, with time constraints varying by scenario(Anastasopoulos et al., 2022; Agarwal et al., 2023; Ahmad et al., 2024). These constraints are typically quantified as latency, often defined as the average time delay between when a word is spoken and when its translation is generated. Accurate latency measurement is thus critical for evaluating system performance.

Various metrics have been introduced to measure SimulST system's latency, including Average Proportion (AP) (Cho and Esipova, 2016), Average Lagging (AL) (Ma et al., 2019), Differentiable Average Lagging (DAL) (Cherry and Foster, 2019), and Length Adaptive Average Lagging (LAAL)(Papi et al., 2022) and Average Token Delay (ATD)(Kano et al., 2022). Recent studies (Papi et al., 2023a,b; Ahmad et al., 2024; Xu et al., 2024) have emphasized computation-aware latency as a more realistic way to evaluate SimulST performance in real-time scenarios where computation time cannot be ignored.

As the performance of SimulST systems improves, researchers are motivated to apply them to unsegmented streaming long speech(Polák, 2023; Ouyang et al., 2024; Papi et al., 2024; Iranzo-Sánchez et al., 2021), which better represents real-world scenarios such as interpreting and lecture transcription. However, computation-aware latency scores reported by existing metrics have been observed to be unrealistically high, hindering progress in this area.

As illustrated in Figure 1, the computation-aware AL and LAAL metrics show drastic increases as the length of streaming speech increases. Specifically, the computation-aware LAAL increases from 6850 ms when the speech length is 25 seconds to 23460 ms when the speech length increased to 100 seconds. In contrast, the theoretical computation unaware LAAL remains mostly consistent, at 8181

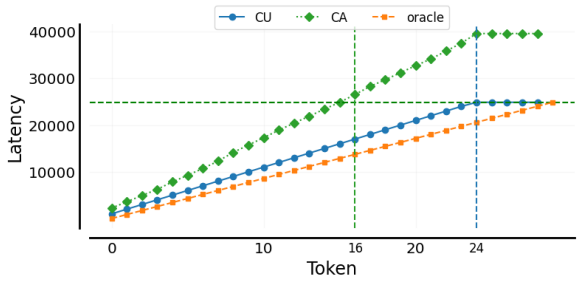

Figure 2: Computation unaware and aware d_i with the corresponding oracle delay d^*, where the intersection represents $\tau'(|\mathbf{X}|)$. After conversion[1], the latency AL_CA only considers the first 46 outputs against the oracle, resulting from the unreliable calculation of computation elapsed, while AL_CU considers all outputs until the last speech segment.

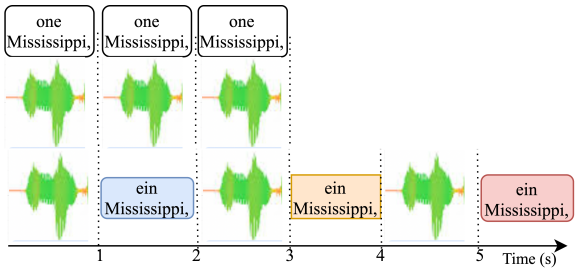

Figure 3: In practice, a SimulST system alternates between reading and writing actions while receiving speech input in a continuous stream. The existing approach implicitly assumes that the time spent on generating text and processing streaming speech occurs sequentially. As a result, this can lead to unreliable accumulation of delay calculations.

ms.

This paper aims to uncover the root causes of the unrealistic computation aware latency measurements and to present our proposed solution for correcting them. Section 2 describes the current latency metrics in detail, while Section 3 outlines the misconceptions underlying these inconsistencies. Our proposed solution is introduced in Section 4, where we address the limitations of existing evaluation methods and provide a refined approach to computation-aware latency measurement. In Section 5, we exemplify our proposed calculation aligned with a real-time setting, and show that this miscalculation also impacts pre-segmented speech.

2 Latency Metrics

An evaluation corpus for a speech translation task contains one or more instances, each consisting of a source speech sequence $\mathbf{X} = [x_1, ..., x_{|\mathbf{X}|}]$ and a reference text sequence $\mathbf{Y}^* = [y_1^*, ..., y_{|\mathbf{Y}^*|}^*]$. The system to be evaluated takes $\mathbf{X}$ as input and generates $\mathbf{Y} = [y_1, ..., y_{|\mathbf{Y}|}]$ as the target language's text translation incrementally.

In the simultaneous speech translation task, the system starts generating a hypothesis with only partial input. It alternates between reading a new source speech segment or writing a new target text segment. Assuming $\mathbf{X}_1 = [x_1, ..., x_j]$, $j < |\mathbf{X}|$ has been read when generating y_i, where x_j represents a raw audio segment of duration T_j. The Computation Aware (CA) and Computation Unaware (CU) delay of a token y_i are defined as:

$$d_i = \begin{cases} \sum_{k=1}^{j} T_k, & \text{CU} \\ \sum_{k=1}^{j} T_k + C_i, & \text{CA} \end{cases} \quad (1)$$

where C_i is the elapsed time when generating the i-th token, as recorded by SimulEval (Ma et al., 2020a) after generating this token.

The latency metrics are calculated using a normalization function, which takes the sequence of delays, $\mathbf{D} = [d_1, \ldots, d_{|\mathbf{Y}|}]$, from SimulEval, along with a corresponding set of oracle delays, $\mathbf{D}^* = [d_1^*, \ldots, d_{|\mathbf{Y}|}^*]$. This process can be formalized as follows:

$$Latency = \frac{1}{\tau'(|\mathbf{X}|)} \sum_{i=1}^{\tau'(|\mathbf{X}|)} (d_i - d_i^*), \quad (2)$$

where $\tau'(|\mathbf{X}|) = \min\{i | d_i = \sum_{j=1}^{|\mathbf{X}|} T_j\}$ and d_i^* represents the oracle delays. Ma et al. (2020a) suggests using $d_i^* = (i-1) \cdot \frac{\sum_{j=1}^{|\mathbf{X}|} T_j}{|\mathbf{Y}^*|}$ to mitigate potential under-generation in Average Lagging (AL) measures. To correct the bias towards over-generation, Papi et al. (2022) recommends substituting $|\mathbf{Y}^*|$ with the larger value between $|\mathbf{Y}^*|$ and $|\mathbf{Y}|$ when computing oracle delays.

3 The Problem of current CA

In practice, the system receives new speech input while generating text translations. However, as shown in Figure 3, the CA delay $\sum_{k=1}^{j} T_k + C_i$ treats a parallel reading and writing process sequentially, assuming the system alternates between reading and writing actions. As a result, the CA delay

[1]For illustration purposes, we plot only one token for three tokens in the stride-3 SimulST system.

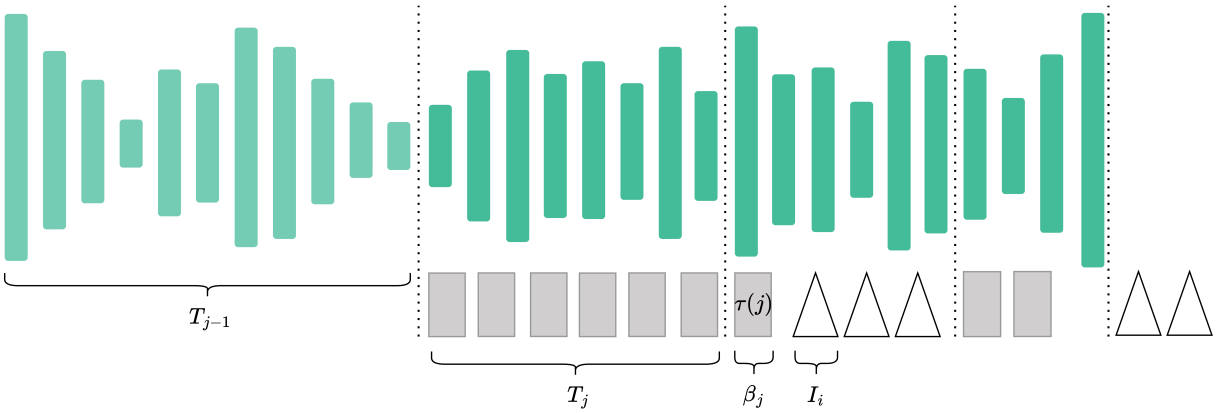

Figure 4: Previous generation time exceeding the current segment x_j introduces additional delay for generating tokens in the current frame. For the given example, if generating each token takes 1 second, after reading 1 second of speech, the system takes 2 seconds to generate 'ein Mississippi', resulting in β_j being 1 second. The delay for I_i would be the sum of the previous speech duration, β_j, and I_i. If there is no buffer, later tokens are not affected.

accumulates the computation cost at each step and deviates from the real-world computation-aware delay, which in turn makes the latency measurement in Equation 2 unreliable.

To illustrate this more concretely, let us consider an example. Suppose we have a source sequence consisting of three segments, each representing the phrase "one Mississippi," which takes approximately one second to say. Thus, $\mathbf{X} = [x_1, x_2, x_3]$ consists of three speech segments, each lasting one second ($T_1 = T_2 = T_3 = 1$ s). The system decides to generate output after processing x_1, and it takes 0.5 second to generate each token. For an English-to-German translation, the system generates the translation "ein" at the timestamp of 1.5 seconds and "Mississippi" at 2 seconds.

Intuitively, after processing all the source segments, the total delay for generating the subsequent tokens y_5 and y_6 should be $\sum_{k=1}^{3} T_k + 0.5 = 3.5$ seconds and 4 seconds, respectively. However, with the current method, the computation timestamps C_i for the third segment are calculated with accumulated computation costs at previous steps. The generation times C_i for the fifth and sixth tokens would be 2.5 seconds and 3 seconds, resulting in delays of 5.5 seconds (3+2.5) and 6 seconds (3+3) for the two tokens.

4 Method

We propose a revised way of calculating CA delay to fix this problem. We define the *inference time* I_i as the elapsed time since processing the previous source segment x_{j-1}. Formally, this is expressed as:

$$I_i = C_i - C_{\tau(j)}, \tag{3}$$

where C_i is the computation timestamp when generating token y_i, and $C_{\tau(j)}$ is the computation timestamp associated with the reference token $y_{\tau(j)}$. The index $\tau(j)$ represents the last token generated before processing the current source segment x_j, defined as:

$$\tau(j) = \max\left\{ i \mid d_i \leq \sum_{k=1}^{j-1} T_k \right\}. \tag{4}$$

Here, d_i is the theoretical computation unaware delay at token y_i, and $\sum_{k=1}^{j-1} T_k$ is the cumulative duration of the source segments up to x_{j-1}.

To represent the accumulated delay effect caused by discrepancies between computation time and source segment durations, we introduce the a buffer β_j corresponding to the source segment x_j:

$$\beta_j = \max\left\{0, \beta_{j-1} + I_{\tau(j)} - T_j\right\}, \tag{5}$$

where β_{j-1} is the buffer from the previous source segment x_{j-1} that affects the accumulation status while processing segment x_j, and T_j is the duration of the current source segment x_j. The buffer accumulates any excess inference time that is not covered by the source segment durations.

Finally, the computation-aware delay d_i for token y_i can be defined as:

$$d_i = \beta_j + I_i + \sum_{k=1}^{j} T_k, \tag{6}$$

which is calculated by combining the accumulation status buffer β, the inference time I, and the cumulative source segment duration.

5 Experiment

To assess the effectiveness of our proposed modification in aligning with the real-world performance of SimulST systems, we simulated a streaming speech translation task using a 25-second input. The average completion time was 27,020 ms[2]. The SimulEval recorded last token's CA* delay calculated by our proposed method was 26,311 ms, **resulting in a difference within 2%**. In contrast, the last token CA delay without our modification was 39,602 ms, **resulting in a difference of 46.6%**.

FRAME	2	6	10	12
CU	1256	1668	2144	2582
CA	1897	2317	2874	3391
CA*	1617	2008	2551	3063

Table 1: AL in ms, we use AlignAtt as SimulST system's policy and set the frame to 2, 6, 10, 12 to represent different latency setting.

FRAME	2	6	10	12
CU	951	1459	1983	2448
CA	1675	2167	2763	3306
CA*	1346	1823	2411	2950

Table 2: LAAL in ms, we use AlignAtt as SimulST system's policy and set the frame to 2, 6, 10, 12 to represent different latency setting.

Effects on Pre-Segmented Speech Due to the incorrect calculation of computation-aware delay, the computation-aware latency deviates from the real latency, and this deviation increases as speech length increases, as shown in Figure 1. However, even on pre-segmented short speeches, the deviation caused by CA delay is still large.

We use the MuST-C v1.0 En-De tst-COMMON (Di Gangi et al., 2019) to examine such effects. We utilize a Conformer-based encoder-decoder offline ST model combined with the AlignAtt policy (Papi et al., 2023a), which relies on cross-attention to determine whether to emit translated words or wait for additional information.

We identify severe CA computation issues with input lengths of 25 seconds. As illustrated in Tables 1 and 2, even for pre-segmented speech averaging 5 seconds in length, the discrepancies in both Average Lagging (AL) and Length-Adaptive Average Lagging (LAAL) are greater than 300 ms.

6 Conclusion

In this work, we investigated the shortcomings of current latency evaluation metrics for SimulST systems, focusing on the discrepancies caused by computation-aware delay miscalculations. We demonstrated that these inaccuracies lead to unrealistic latency estimates, not only in long streaming speech translation but also in pre-segmented speech. Our proposed modification addresses these misconceptions and aligns the latency calculations more accurately with the SimulST system's real behavior, leading to a more reliable evaluation for both SimulST and StreamingST systems.

Limitations

This work introduces CA* to improve the accuracy of latency evaluation for SimulST systems by addressing computation-aware delay. However, other limitations remain. Notably, oracle delay approximations may inaccurate, especially for unsegmented streaming speech, which includes long pauses and varied segments. Future work should refine these approximations to reduce errors.

Furthermore, this study focused on AL and LAAL. Although CA* can be generalized to other metrics, additional evaluation is required to confirm its effectiveness across various latency measures.

We also used ChatGPT for grammar revisions.

References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, William Chen, Khalid Choukri, Alexandra Chronopoulou, Anna Currey, Thierry Declerck, Qianqian Dong, Kevin Duh, Yannick Estève, Marcello Federico, Souhir Gahbiche, Barry Haddow, Benjamin Hsu, Phu Mon Htut, Hirofumi Inaguma, Dávid Javorský, John Judge, Yasumasa Kano, Tom Ko, Rishu Kumar, Pengwei Li, Xutai Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Ha Nguyen, Jan Niehues, Xing Niu, Atul Kr. Ojha, John E. Ortega,

[2]We built the service based on Gradio and measured the finish time by detecting the EOS token. The experiment was run five times, employing a wait-k stride-n policy and a Conformer-based encoder-decoder offline ST model.

Proyag Pal, Juan Pino, Lonneke van der Plas, Peter Polák, Elijah Rippeth, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Yun Tang, Brian Thompson, Kevin Tran, Marco Turchi, Alex Waibel, Mingxuan Wang, Shinji Watanabe, and Rodolfo Zevallos. 2023. FINDINGS OF THE IWSLT 2023 EVALUATION CAMPAIGN. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 1–61, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Ibrahim Said Ahmad, Antonios Anastasopoulos, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, William Chen, Qianqian Dong, Marcello Federico, Barry Haddow, Dávid Javorský, Mateusz Krubiński, Tsz Kim Lam, Xutai Ma, Prashant Mathur, Evgeny Matusov, Chandresh Maurya, John McCrae, Kenton Murray, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Atul Kr. Ojha, John Ortega, Sara Papi, Peter Polák, Adam Pospíšil, Pavel Pecina, Elizabeth Salesky, Nivedita Sethiya, Balaram Sarkar, Jiatong Shi, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Alex Waibel, Shinji Watanabe, Patrick Wilken, Petr Zemánek, and Rodolfo Zevallos. 2024. FINDINGS OF THE IWSLT 2024 EVALUATION CAMPAIGN. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 1–11, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation. *Preprint*, arXiv:1906.00048.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *Preprint*, arXiv:1606.02012.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Javier Iranzo-Sánchez, Jorge Civera Saiz, and Alfons Juan. 2021. Stream-level latency evaluation for simultaneous machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 664–670, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2022. Average token delay: A latency metric for simultaneous translation. *arXiv preprint arXiv:2211.13173*.

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. 2019. 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*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587, Suzhou, China. Association for Computational Linguistics.

Siqi Ouyang, Xi Xu, Chinmay Dandekar, and Lei Li. 2024. Fasst: Fast llm-based simultaneous speech translation. *Preprint*, arXiv:2408.09430.

Sara Papi, Marco Gaido, Matteo Negri, and Luisa Bentivogli. 2024. StreamAtt: Direct streaming speech-to-text translation with attention-based audio history selection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3692–3707, Bangkok, Thailand. Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages

12–17, Online. Association for Computational Linguistics.

Sara Papi, Matteo Negri, and Marco Turchi. 2023a. Attention as a guide for simultaneous speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada. Association for Computational Linguistics.

Sara Papi, Marco Turchi, and Matteo Negri. 2023b. Alignatt: Using attention-based audio-translation alignments as a guide for simultaneous speech translation. In *Proceedings of INTERSPEECH 2023*, pages 3974–3978.

Peter Polák. 2023. Long-form simultaneous speech translation: Thesis proposal. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 64–74, Nusa Dua, Bali. Association for Computational Linguistics.

Xi Xu, Siqi Ouyang, Brian Yan, Patrick Fernandes, William Chen, Lei Li, Graham Neubig, and Shinji Watanabe. 2024. CMU's IWSLT 2024 simultaneous speech translation system. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 154–159, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.