A Practical Evaluation Method for Long-Form Simultaneous Speech-to-Speech Translation

Yulin Xue, Siqi Ouyang, Lei Li
Carnegie Mellon University
{yulinx,siqiouya}@andrew.cmu.edu,leili@cs.cmu.edu

Abstract

Simultaneous speech-to-speech translation (SimulS2ST) enables real-time cross-lingual communication, but existing evaluation has focused largely on short or pre-segmented speech rather than long-form, continuous input. Prior approaches are difficult to reproduce and make assumptions that do not hold for end-to-end systems. We present a practical evaluation method for long-form SimulS2ST. Given source speech, pre-segmented source transcripts, and reference translations, we run automatic speech recognition (ASR) and forced alignment on the generated target speech to recover token-level timestamps, then apply a sentence-embedding-based aligner to match the target text to its corresponding source sentences. This enables sentence-level computation of latency and quality metrics, including YAAL and xCOMET, which are then aggregated into final system-level scores. Experiments on representative SimulS2ST systems show that the method is effective in practice and reveal that current systems suffer from substantial latency accumulation on long speech.

1 Introduction

Simultaneous Speech-to-Speech Translation (SimulS2ST) translates streaming source speech into target-language speech in real time (Zheng et al., 2020), enabling low-latency cross-lingual communication in scenarios such as multilingual conversations and international conferences. However, most prior work evaluates SimulS2ST on pre-segmented or short speech, despite the fact that real-world input, such as conference speech, is often continuous and may last for hours (Sudoh et al., 2020; Ma et al., 2022; Liu et al., 2022; Communication et al., 2023; Zhang et al., 2024).

One early effort toward long-form SimulS2ST evaluation is Boundary-Aware Latency (pBAL), which segments target speech into sentences, applies forced alignment to recover target token timestamps, and computes latency based on these timestamps (Zheng et al., 2020). This general paradigm is closely related to recent efforts in long-form simultaneous speech-to-text translation (SimulS2TT) evaluation (Papi et al., 2024; Polák et al., 2026). However, pBAL has important practical limitations. First, it is not open-sourced, which makes it difficult to reproduce and adopt in subsequent research. Second, pBAL was designed for cascade systems comprising ASR, machine translation (MT), and text-to-speech (TTS), which introduces several limitations. In particular, it segments the target speech to align with the streaming ASR output of the source speech rather than with ground-truth source sentences, making the evaluation sensitive to source-side ASR errors. It also assumes access to target text for forced alignment, which is not available for some end-to-end (E2E) SimulS2ST systems (Labiausse et al., 2025).

In this paper, we propose a practical evaluation method for long-form SimulS2ST. We assume access to source speech, source transcripts pre-segmented into sentences, and their corresponding translation sentences. Given target speech produced by a SimulS2ST system, we first run ASR and forced alignment with state-of-the-art models to obtain target text with token-level timestamps. We then use the sentence-embedding-based method SEGALE (Wang et al., 2025) to segment the target text into sentences aligned with the source sentences. Finally, for each aligned sentence, we compute standard latency metrics such as YAAL (Polák et al., 2026) and quality metrics such as xCOMET (Guerreiro et al., 2024), and average the sentence-level scores to obtain the final latency and quality scores. In our experiments, we evaluate several representative SimulS2ST systems with this method and analyze the quality of both ASR and sentence segmentation. We observe that even state-of-the-art systems exhibit latency accumulation on long speech. We will release the

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 32–39

GitHub repository in the camera-ready version.

2 Related Works

Long-form simultaneous translation evaluation Latency evaluation for simultaneous translation has traditionally been studied in pre-segmented settings, where the input speech is split into utterances prior to evaluation. StreamLAAL (Papi et al., 2024) extends utterance-level evaluation to the long-form setting by first segmenting the hypothesis into utterances aligned with the reference translation sentences using mwerSegmenter (Matusov et al., 2005), then computing latency for each aligned hypothesis utterance and its corresponding reference sentence. LongYAAL (Polák et al., 2026) improves upon StreamLAAL by mitigating the structural bias in latency evaluation and introduces SoftSegmenter, which yields better segmentation and alignment than mwerSegmenter. These methods are designed for simultaneous speech-to-text translation, while our work extends them to the evaluation of simultaneous speech-to-speech translation.

Long-form machine translation evaluation. Another related line of work studies automatic evaluation for long-form machine translation. mwerSegmenter (Matusov et al., 2005) aligns hypothesis and reference translation sentences by minimizing word error rate; however, it handles sentence boundaries poorly and often fails in cases of over- or under-translation. SEGALE (Wang et al., 2025) improves upon mwerSegmenter by using a sentence boundary detector such as spaCy[1] to recover sentence boundaries and by correctly penalizing over- and under-translation. Our work leverages SEGALE as a more robust segmenter for the long-form hypothesis.

3 Method

In this section, we first introduce the formulation (Section 3.1). We then describe the ASR and forced alignment procedures (Section 3.3), the target speech segmentation method (Section 3.4), and the computation of the final latency and quality scores (Section 3.5).

3.1 Formulation

We define a long-form input speech stream as $\boldsymbol{s} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$, where each $\boldsymbol{x}_i \in \mathbb{R}^{|\boldsymbol{x}_i|}$ denotes the speech waveform of the i-th sentence. Let $\boldsymbol{y}_i$ denote the reference text translation of sentence i. Given the input speech stream $\boldsymbol{s}$, a SimulS2ST system incrementally generates target speech $\hat{\boldsymbol{t}}$. We assume that the input and target speech streams are temporally aligned at the start, i.e., they share the same initial timestamp. The goal of the evaluation method is to compute latency and quality scores for the generated target speech $\hat{\boldsymbol{t}}$ given $\boldsymbol{s}$, $\boldsymbol{x}_{1:n}$, and $\boldsymbol{y}_{1:n}$.

3.2 Overview

At a high level, our evaluation pipeline consists of three stages. First, given the target speech generated by a SimulS2ST system, we run ASR to obtain the target-side text and apply forced alignment to recover token-level timestamps on the target speech. Second, following SEGALE, we segment the target text into sentences and align them with the source transcript sentences and their reference translations, producing sentence groups that may reflect one-to-one, one-to-many, many-to-one, many-to-many, or null alignments. Finally, for each aligned group, we compute latency using existing metrics such as YAAL and translation quality using sentence-level metrics such as xCOMET. The group-level scores are then averaged into final system-level latency and quality scores.

3.3 Transcribe with Timestamps

Given target speech $\hat{\boldsymbol{t}}$, we use state-of-the-art ASR and forced alignment models: Qwen3-ASR-1.7B and Qwen3-ForcedAligner-0.6B (Shi et al., 2026), to transcribe target speech $\hat{\boldsymbol{t}}$ into text $\hat{\boldsymbol{y}} = (\hat{y}_1, \cdots, \hat{y}_{|\hat{\boldsymbol{y}}|})$ and obtain token-level timestamps $\boldsymbol{d} = (d_1, \cdots, d_{|\hat{\boldsymbol{y}}|})$ where d_i denotes the end time of token $\hat{y}_i$.

For long-form speech, we process the input in a chunk-wise manner. We divide the target speech $\hat{\boldsymbol{t}}$ into C consecutive chunks, each with duration 180 seconds,

$$\hat{\boldsymbol{t}} = (\hat{\boldsymbol{t}}^{(1)}, \hat{\boldsymbol{t}}^{(2)}, \ldots, \hat{\boldsymbol{t}}^{(C)}). \tag{1}$$

For each chunk $\hat{\boldsymbol{t}}^{(c)}$, the ASR model produces a partial transcription

$$\hat{\boldsymbol{y}}^{(c)} = (\hat{y}_1^{(c)}, \ldots, \hat{y}_{|\hat{\boldsymbol{y}}^{(c)}|}^{(c)}). \tag{2}$$

Forced alignment is then applied to each chunk using the corresponding audio and recognized text to produce a chunk-level timestamp sequence

$$\boldsymbol{d}^{(c)} = (d_1^{(c)}, \ldots, d_{|\hat{\boldsymbol{y}}^{(c)}|}^{(c)}), \tag{3}$$

[1] https://spacy.io/

where $d_i^{(c)}$ denotes the end time of token $\hat{y}_i^{(c)}$ within the c-th chunk. Let o_c be the starting time offset of chunk $\boldsymbol{t}^{(c)}$ in the original speech stream. We map chunk-level timestamps back to the global timeline by

$$\tilde{d}_i^{(c)} = d_i^{(c)} + o_c. \tag{4}$$

Finally, the full transcription and timestamp sequence are obtained by concatenating all chunk-level results:

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}^{(1)} \oplus \cdots \oplus \hat{\boldsymbol{y}}^{(C)} \tag{5}$$

$$\boldsymbol{d} = \tilde{\boldsymbol{d}}^{(1)} \oplus \cdots \oplus \tilde{\boldsymbol{d}}^{(C)}. \tag{6}$$

3.4 Robust Segmentation with SEGALE

We segment the target text $\hat{\boldsymbol{y}}$ into sentence-level units and align them with the source speech sentences $\boldsymbol{x}_{1:n}$ using SEGALE. We first split $\hat{\boldsymbol{y}}$ into sentences $\hat{\boldsymbol{y}}_{1:m}$ with spaCy [2]. Given the source speech sentences $\boldsymbol{x}_{1:n}$, their reference translations $\boldsymbol{y}_{1:n}$, and the segmented target sentences $\hat{\boldsymbol{y}}_{1:m}$, SEGALE performs sentence alignment using Vecalign (Thompson and Koehn, 2020) with an adaptive skip-penalty search strategy.

To support many-to-many alignment, SEGALE constructs candidate contiguous spans on both the source and target sides, rather than restricting alignment to individual sentences. Let $\boldsymbol{x}_{i:j} = \boldsymbol{x}_i \oplus \cdots \oplus \boldsymbol{x}_j$ denote a source span and $\hat{\boldsymbol{y}}_{p:q} = \hat{\boldsymbol{y}}_p \oplus \cdots \oplus \hat{\boldsymbol{y}}_q$ denote a target span, where $\oplus$ denotes concatenation. For each source span $\boldsymbol{x}_{i:j}$ and target span $\hat{\boldsymbol{y}}_{p:q}$, SEGALE computes an embedding-based matching cost, with lower cost assigned to more semantically similar spans. Vecalign then finds a monotonic alignment between the source and target sentence sequences while allowing null alignments on either side, controlled by a skip penalty β_{skip}.

The skip penalty determines the trade-off between forcing matches and allowing deletions. A large β_{skip} makes skipping expensive, so the aligner prefers fewer null alignments and more forced matches; this usually keeps the null-alignment ratio (NA ratio) low but increases the average alignment cost because semantically weak pairs are more likely to be matched. In contrast, a small β_{skip} makes skipping cheap, so the aligner more readily leaves segments unmatched; this typically increases the NA ratio and decreases the average alignment cost, since high-cost pairs are skipped and only easier matches remain. Therefore, SEGALE adaptively searches over β_{skip}: it starts from a relatively large value and progressively decreases it in small steps. Once the average alignment cost falls below a threshold or the NA ratio exceeds a threshold, SEGALE treats this as the onset of over-deletion and returns the alignment from the previous step.

We denote the alignment output as

$$A = (A_1, \ldots, A_r), \tag{7}$$

where each alignment group is defined as

$$A_k = (X_k, Y_k, \hat{Y}_k). \tag{8}$$

Here, X_k is a consecutive subset of source sentences from $\boldsymbol{x}_{1:n}$, Y_k is a consecutive subset of reference translation sentences from $\boldsymbol{y}_{1:n}$, and $\hat{Y}_k$ is a consecutive subset of target sentences from $\hat{\boldsymbol{y}}_{1:m}$. SEGALE naturally handles both over-translation and under-translation. In the case of over-translation, some target sentences do not correspond to any source sentence, resulting in an empty X_k and Y_k. In the case of under-translation, some source sentences do not correspond to any target sentence, resulting in an empty $\hat{Y}_k$. Such phenomena occur frequently in simultaneous translation, making this robustness important for long-form SimulS2ST evaluation.

3.5 Latency Computation

Given the alignment produced by SEGALE, we compute latency at the alignment-group level. For each group $A_k = (X_k, Y_k, \hat{Y}_k)$, let T_k^s and T_k^e denote the start and end times of the source span X_k, respectively. Let $\boldsymbol{d}_k = (d_1, \ldots, d_{|\hat{Y}_k|})$ denote the token-level timestamps obtained by forced alignment for the target sentence group $\hat{Y}_k$.

We define the ideal delay of the i-th target token as

$$d_i^* = T_k^s + (i-1) \cdot \frac{T_k^e - T_k^s}{\max\{|Y_k|, |\hat{Y}_k|\}}, \tag{9}$$

where $|Y_k|$ and $|\hat{Y}_k|$ are the numbers of tokens in the reference and target sentence groups, respectively.

The latency of group A_k is then computed as

$$l_k = \frac{1}{|\hat{Y}_k|} \sum_{i=1}^{|\hat{Y}_k|} (d_i - d_i^*). \tag{10}$$

Following LongYAAL (Polák et al., 2026), we exclude the target tokens generated after the end of the full source stream $\boldsymbol{s}$.

[2] https://spacy.io/

Finally, we compute the long-form latency by averaging over all alignment groups:

$$\text{Latency} = \frac{1}{r}\sum_{k=1}^{r} l_k. \tag{11}$$

For over-translation or under-translation cases, where $X_k = Y_k = \emptyset$ or $\hat{Y}_k = \emptyset$, we exclude these groups from latency computation, since latency is not well-defined without both source and target content.

3.6 Quality Computation

Let Q denote a sentence-level quality metric, such as COMET (Rei et al., 2020) or MetricX (Juraska et al., 2024). For each alignment group $A_k = (X_k, Y_k, \hat{Y}_k)$, we compute the quality score of group A_k as

$$q_k = Q(X_k, Y_k, \hat{Y}_k). \tag{12}$$

For over-translation or under-translation cases, we directly assign the minimum possible score of the metric, denoted by $Q_{\min}$. For example, $Q_{\min} = 0$ for COMET and $Q_{\min} = -25$ for MetricX. The final long-form quality score is then computed by averaging over all alignment groups:

$$\text{Quality} = \frac{1}{r}\sum_{k=1}^{r} q_k. \tag{13}$$

4 Experiments

4.1 Setup

Datasets We evaluate existing SimulS2ST systems on two datasets: ACL 60/60 devv (Salesky et al., 2023) and Audio-NTREX-4L test (Labiausse et al., 2026). ACL 60/60 dev consists of five English ACL talks, each approximately 10 minutes long, translated into multiple languages, in which we consider three directions: English to German/Japanese/Chinese. Audio-NTREX-4L is a multilingual speech translation benchmark introduced in Hibiki-Zero (Labiausse et al., 2026). It is built from the NTREX text translation dataset by synthesizing source-language speech with high-quality TTS systems and multilingual speaker voices. The benchmark covers French, German, Portuguese, and Spanish as source languages and English as the target language. Each direction in our test split contains 450 speeches with an average duration of about 45 seconds, and we evaluate all four X-to-English directions.

SimulS2ST Systems We consider three representative multilingual SimulS2ST systems:

- **Seed LiveInterpret 2.0** (Cheng et al., 2025): a product-level end-to-end simultaneous interpretation system designed for high-fidelity, ultra-low-latency speech-to-speech generation. It supports voice cloning and is built on a duplex speech-to-speech architecture.

- **Hibiki-Zero** (Labiausse et al., 2026): an end-to-end simultaneous speech-to-speech translation system built on the Moshi duplex architecture (Défossez et al., 2024). It is first trained on sentence-level aligned speech translation data and then further optimized with GRPO (Shao et al., 2024) to reduce latency while preserving translation quality.

- **SeamlessStreaming** (Communication et al., 2023): a multilingual streaming speech translation model from the Seamless family. It uses Efficient Monotonic Multihead Attention (EMMA) (Ma et al., 2023) to generate low-latency translations without waiting for the full source utterance, enabling simultaneous speech-to-speech and speech-to-text translation across multiple source and target languages.

Seed LiveInterpret 2.0 is evaluated through the Volcano Engine API [3], while Hibiki-Zero and SeamlessStreaming are run locally on a single NVIDIA L40S GPU.

4.2 Evaluation

The evaluation results on ACL 60/60 dev set are shown in Table 1. Seed LiveInterpret 2.0 consistently achieves better translation quality than SeamlessStreaming, but at substantially higher latency, e.g., 9.4 seconds for En-Ja.

The results on Audio-NTREX-L test set are shown in Table 2. Overall, all systems achieve reasonably good translation quality. Among them, Seed LiveInterpret 2.0 obtains the best translation quality, but with more than 2 seconds higher latency than Hibiki-Zero and SeamlessStreaming.

We also observe that latency is less stable for En→X directions, whereas it is much more consistent for X→En directions.

[3] https://www.volcengine.com/docs/6561/1756902?lang=en

System	En→De	En→Ja	En→Zh
SeamlessStreaming	**4.333** / 67.56	**2.434** / 42.89	**1.725** / 40.66
Seed LiveInterpret 2.0	7.939 / **85.39**	9.413 / **45.48**	5.306 / **72.78**

Table 1: Evaluation results of SimulS2ST systems on ACL 60/60 dev set. A / B denotes Latency (second) / xCOMET-XL. The best latency and quality scores are shown in bold.

System	Fr→En	De→En	Pt→En	Es→En
SeamlessStreaming	3.520 / 77.50	3.833 / 78.95	3.566 / 76.11	**3.608** / 77.88
Seed LiveInterpret 2.0	5.892 / **86.67**	5.933 / **88.63**	5.530 / **86.94**	5.592 / **88.63**
Hibiki-Zero	**3.271** / 80.21	**3.313** / 79.50	**3.312** / 79.00	3.657 / 81.39

Table 2: Evaluation results of SimulS2ST systems on Audio-NTREX-L test set. A / B denotes Latency (second) / xCOMET-XL. The best latency and quality scores are shown in bold.

Model	De	Ja	Zh
Qwen3-ASR-1.7B	15.37	27.60	4.50
WhisperX	13.80	27.30	5.52

Table 3: WER/CER of Qwen3-ASR-1.7B and WhisperX on generated target speech for ACL 60/60 dev.

4.3 Analysis

Poor En→Ja Performance of Seed LiveInterpret 2.0 On En→Ja direction, Seed LiveInterpret 2.0 exhibits very high latency, reaching nearly 10 seconds, while achieving an xCOMET-XL score of only 45.48. Our initial analysis suggests that the ASR transcripts of the generated target speech contain substantial gibberish and fragmented Japanese. To determine whether this issue stems from Qwen3-ASR's limited Japanese recognition performance or from poor Japanese speech synthesis quality, we measure word/character error rates (WER/CER) using both Qwen3-ASR-1.7B and WhisperX (Bain et al., 2023). The ground-truth target text is taken from the Seed LiveInterpret 2.0 API, which returns both synthesized target speech and target text. The results are shown in Table 3. Both Qwen3-ASR-1.7B and WhisperX yield very high CER on En→Ja, suggesting that the problem is more likely caused by poor Japanese speech synthesis quality in Seed LiveInterpret 2.0.

Segmentation Quality We analyze the segmentation quality of SEGALE on the ACL 60/60 En→Zh dev set and compare it with the recently proposed SoftSegmenter (Polák et al., 2026). The results show that SEGALE achieves a segmentation accuracy of 90.9%, substantially outperforming SoftSegmenter (79.1%). We observe that SoftSegmenter often shifts boundary-adjacent fragments across neighboring sentences, e.g., attaching the beginning of a sentence to the previous segment or the ending to the next one. Softsegmenter is also particularly brittle when semantically related content is realized with different surface forms. For example, when the reference contains foreign-language expressions while the prediction translates or paraphrases them, local token-level matching can break down. This is likely because its re-segmentation relies on local token-level matching rather than explicit sentence-level boundary modeling. In contrast, such errors are much less frequent with SEGALE. In contrast, SEGALE is much less affected by such cases and produces more semantically coherent segmentation.

Latency Accumulation We further observe that, on long-form speech in the ACL 60/60 dev set (around 10 minutes), latency is substantially higher than on the Audio-NTREX-L test set (around 45 seconds). To better understand this phenomenon, we compute the ending offset of SEGALE-aligned sentences for both SeamlessStreaming and Seed LiveInterpret 2.0 on each speech in the ACL 60/60 dev set, as shown in Figure 1. We find that on long speech, both systems exhibit increasingly larger ending offsets as more input speech arrives, with the only exception being Seed LiveInterpret 2.0 on the En→Zh direction. Further analysis shows that latency accumulation is related to the sentence-level target-source duration difference, as shown in Figure 2. For En→Zh, where the target speech is generally shorter than the source speech and the duration difference is concentrated in a negative range

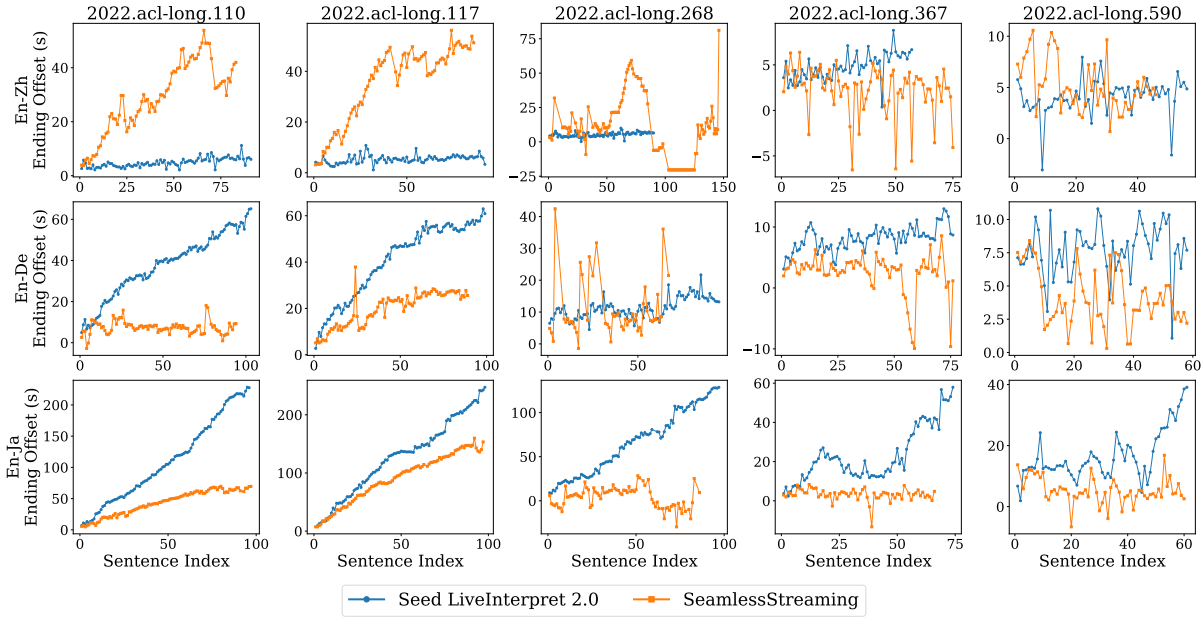

Figure 1: The ending offset of each aligned sentence for two systems on every speech in the ACL 60/60 dev set. The results show that latency generally accumulates as the source speech becomes longer.

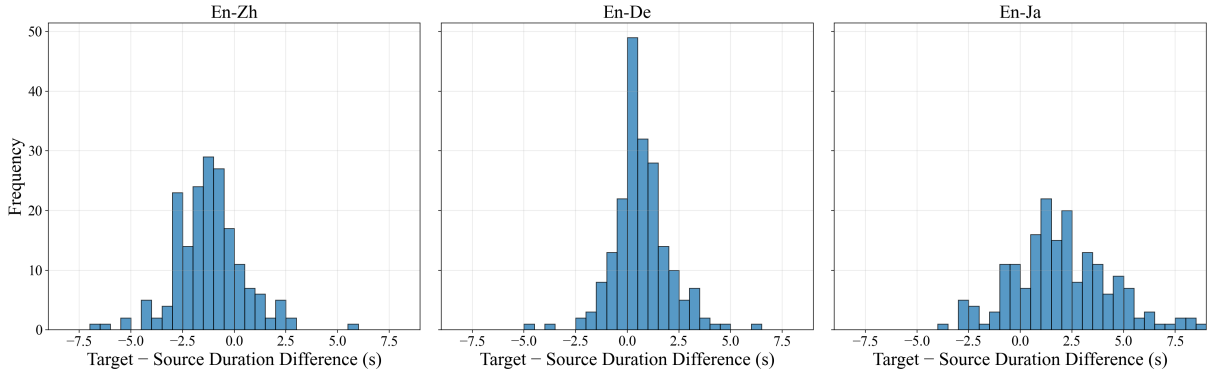

Figure 2: Distribution of target–source duration differences on the ACL 60/60 dev set. En→Zh is mostly negative, En→De is centered around zero, and En→Ja is mostly positive.

(roughly $[-2.5, -0.5]$ seconds), the ending offset stays small and stable. By contrast, for En→Ja, where the target speech is typically longer than the source and the duration difference falls in a positive range (roughly $[0, 3]$ seconds), the ending offset exceeds 200 seconds near the end of the speech. This suggests that even state-of-the-art SimulS2ST systems still suffer from latency accumulation on speech spanning minutes, highlighting the need for future research to address this issue.

5 Conclusion

We present a practical evaluation method for long-form SimulS2ST. Combining ASR, forced alignment, and SEGALE-based sentence alignment, it enables sentence-level evaluation of latency and translation quality on continuous speech across representative SimulS2ST systems. Experiments on ACL 60/60 and Audio-NTREX-L validate the method and show that SEGALE provides robust segmentation for long-form evaluation. More importantly, our analysis reveals a key limitation of current systems: latency accumulates substantially on long speech. We hope this work lays a strong foundation for future research on reliable, low-latency SimulS2ST.

References

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. In *Interspeech 2023*, pages 4489–4493.

Shanbo Cheng, Yu Bao, Zhichao Huang, Yu Lu, Ningxin Peng, Lu Xu, Runsheng Yu, Rong Cao, Yujiao Du, Ting Han, Yuxiang Hu, Zeyang Li, Sitong Liu, Shengtao Ma, Shiguang Pan, Jiongchen Xiao, Nuo Xu, Meng Yang, Rong Ye, and 9 others. 2025. Seed liveinterpret 2.0: End-to-end simultaneous speech-to-speech translation with your voice. *Preprint*, arXiv:2507.17527.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, and 46 others. 2023. Seamless: Multilingual expressive and streaming speech translation. *Preprint*, arXiv:2312.05187.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *Preprint*, arXiv:2410.00037.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xCOMET: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Tom Labiausse, Romain Fabre, Yannick Estève, Alexandre Défossez, and Neil Zeghidour. 2026. Simultaneous speech-to-speech translation without aligned data. *Preprint*, arXiv:2602.11072.

Tom Labiausse, Laurent Mazaré, Edouard Grave, Alexandre Défossez, and Neil Zeghidour. 2025. High-fidelity simultaneous speech-to-speech translation. In *Forty-second International Conference on Machine Learning*.

Danni Liu, Changhan Wang, Hongyu Gong, Xutai Ma, Yun Tang, and Juan Pino. 2022. From Start to Finish: Latency Reduction Strategies for Incremental Speech Synthesis in Simultaneous Speech-to-Speech Translation. In *Interspeech 2022*, pages 1771–1775.

Xutai Ma, Hongyu Gong, Danni Liu, Ann Lee, Yun Tang, Peng-Jen Chen, Wei-Ning Hsu, Phillip Koehn, and Juan Pino. 2022. Direct simultaneous speech-to-speech translation with variational monotonic multihead attention. *Preprint*, arXiv:2110.08250.

Xutai Ma, Anna Sun, Siqi Ouyang, Hirofumi Inaguma, and Paden Tomasello. 2023. Efficient monotonic multihead attention. *Preprint*, arXiv:2312.04515.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

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.

Peter Polák, Sara Papi, Luisa Bentivogli, and Ondřej Bojar. 2026. Better late than never: Meta-evaluation of latency metrics for simultaneous speech-to-text translation. *Preprint*, arXiv:2509.17349.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 62–78, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-asr technical report. *Preprint*, arXiv:2601.21337.

Katsuhito Sudoh, Takatomo Kano, Sashi Novitasari, Tomoya Yanagita, Sakriani Sakti, and Satoshi Nakamura. 2020. Simultaneous speech-to-speech translation system with neural incremental asr, mt, and tts. *Preprint*, arXiv:2011.04845.

Brian Thompson and Philipp Koehn. 2020. Exploiting sentence order in document alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007, Online. Association for Computational Linguistics.

Kuang-Da Wang, Shuoyang Ding, Chao-Han Huck Yang, Ping-Chun Hsieh, Wen-Chih Peng, Vitaly Lavrukhin, and Boris Ginsburg. 2025. Extending

automatic machine translation evaluation to book-length documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32323–32339, Suzhou, China. Association for Computational Linguistics.

Shaolei Zhang, Qingkai Fang, Shoutao Guo, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. StreamSpeech: Simultaneous speech-to-speech translation with multi-task learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8964–8986, Bangkok, Thailand. Association for Computational Linguistics.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, Jiahong Yuan, Kenneth Church, and Liang Huang. 2020. Fluent and low-latency simultaneous speech-to-speech translation with self-adaptive training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3928–3937, Online. Association for Computational Linguistics.