Main Changes

In our revised submission, we address all suggestions given by the Area Chair and the Reviewers. In general, we improve the writing by including more details and further results.

- Updated Figure 1 to include the True Latency and the proposed YAAL metric.
- Simplified notation in 2.1.
- Cited Huber et al. (2023) in 2.2 (lines 204-216) when mentioning long-form evaluation.
- Simplified Figure 2 by keeping only the two most important segmentation examples and added a better description.
- Moved description of the proposed SoftSegmenter from the Appendix to Section 3.2 (lines 341-368).
- Added a paragraph "Detecting Anomalous Policy" and Figure 4 to Section 5.1 (lines 474-498) that motivates the problems with latency overestimation mentioned in Section 3.1.
- Improved wording and typos.

Meta Review

Summary Of Suggested Revisions:

The authors should add many details and clarifications as promised.

The authors didn't reply to the review about the comparison of YAAL and human evaluation, which may help the readers understanding better.

As for the "motivation" mentioned by Reviewer s1kb, I think that's caused by the lack of detailed explanation of other metrics, which will help the readers to better understand the difference between the proposed YAAL and previous work. So I think it's better to provide this to reviewer for reference, as well as adding them into the next version.

We update all sections with better wording and clarifications. We also agree that human evaluation would be very interesting, but unfortunately, we do not have the budget to perform a comprehensive human evaluation study, which we leave for future work. We include a detailed explanation of other metrics in Section 3.1, while Section 2.1 contains their definitions as given in the original papers that introduced these metrics.

Reviewer r6zH

Summary Of Weaknesses: Authors opted to use pairwise comparison with minimal explanation provided in lines 416-419. It would be great to see how would YAAL perform as an absolute measure to rank different systems. Especially, it would be great to add YAAL to Figure 1. This is important because in lines 486-488, experiments show that AP can perform similar to YAAL.

We added absolute comparison (similar to Figure 3) to Appendix Figure 5. We also added True Latency and YAAL into Figure 1.

Reviewer FibH

Summary Of Weaknesses:
Short-Form Evaluation
To my understanding the proposed YAAL is simply AL without the last token. I can see that the tail word problem is a real issue, and needs to be addressed, but I'm having trouble understanding how this solution helps making latency measurements more realistic. From how I see it, footnote 2 even describes the existence of ”systems that delay a large portion of translation until the end of the segment” (page 4).

Consider for example a translation from English into German or Japanese. In those target languages the verb oftentimes comes in the end of the sentence, while in English the verb appears early in the input sentence, so the model might delay outputting the verb until the sentence end. It seems to me that considering the tail words is important in this case, because latency may be consistently overestimated in this setting.

Chinese on the other hand (as far as I'm aware) has a more similar word order to English (in terms of verb postition), and indeed Figure 3 is showing fewer off-diagonal outliers for English-Chinese translation, compared to German and Japanese.

The topic is futher discussed in line 452-467, stating that various metrics compute a ”severe overestimation of the systems' actual latency” (line 462) by considering the tail words. In their upcoming ACL 2025 paper [1] discuss behavior of SST systems which they term ”degeneration to an offline system” where the model waits until the end of the segment to start outputting any tokens. Their ”over-wait” metric measures this behavior but what they report is similar to what this work reports (line 452-460 and line 508), i.e. in the high-latency regime 72% of tokens are tail end tokens. It seems to me, however, that the proposed YAAL hides this degenerated behavior, or potentially even rewards it with lower latency scores. Looking at the example from footnote 5 further confuses me: why are 4.95 seconds latency for system that waits ¿8 seconds to output anything after the first token called ”severe overestimation”?

As discussed in Section 5.1, paragraph “Detecting Anomalous Policy,“ the overestimation hinders clear detection of the anomalous policy. The second-best metric, LAAL, which includes the tail words and thus overestimates the latency, is clearly less suitable for the detection of this degenerated behavior.

> The evaluation of YAAL shows strong correlation with/accuracy by comparison to the introduced ”true latency” (Equation 9), but it seems to me that this true latency is strongly biased towards YAAL, since the authors choose to also exclude tail words from the true latency calculation. Would these observations still hold true if the true latency considers all output tokens, but the SST system is fed silence after the input audio is fully consumed?

Excluding the tail words from the evaluation is indeed correct. After filtering out all systems that follow the anomalous policy (see the bottom part of Table 1), all metrics of the “AL” family (AL, LAAL, DAL) that have similar definition to YAAL, have a very similar accuracy to the proposed YAAL, showing that the “bias” is only present when including the degenerated systems. However, as discussed above, this bias that is present only in the degenerated systems actually helps to detect these systems, and does not affect normal systems. Finally, the evaluation of systems with appended silence is infeasible and incorrect. All available data used in this paper are based on system logs from IWSLT evaluation campaigns, so we do not have access to the systems, and thus we cannot replicate the comparison by appending the silence. However, more crucially, this evaluation would be incorrect, as a system could cheat if it detected artificial silence.

> Long-Form Evaluation Long-form evaluation without reliance on resegmentation (i.e. by mWERSegmenter-like tools) seems to be already solved in [2], whereas this work argues that proper long-form evaluation requires high-quality resegmentation.

We added a discussion on this paper in Section 2.2 (lines 204-216). However, it is not true that the cited paper proposes an evaluation without relying on resegmentation, as it requires the system itself to provide a segmentation.

Reviewer s1kb

> **Summary Of Weaknesses:** W1: Background and motivation are difficult to follow:
>
> Section 2: Each metric is accompanied by a mathematical formula, without a good explanation in text. The difference between each metric, their pros and cons, and the assumptions required for each metric are not stated

> explicitly. The explanation in L244-L286 is not provided with explicit links to the mathematical formulas defined in Section 2.
>
> Figure 2: It is unclear to me why the model's behaviors are different in these different settings. Why does the model under "Simultaneous Segmentation" setting output a big chunk afterward, but not smaller chunks like "Without Segmentation"? Why is there a delay in "Simultaneous Segmentation" compared to "Oracle Segmentation"?
>
> Line 263: Why would an offline system have infinite latency? Is this statement only applicable to Long-form?
>
> Can we simply append silence to the end of each audio segment for a more realistic simulation?

We improved the wording of Section 2 and Figure 2 and fixed the offline system's latency to undefined rather than infinite. See above for the discussion about appending silence.

> W2: Missing important related work
>
> Paper: "End-to-End Evaluation for Low-Latency Simultaneous Speech Translation". This work introduced a framework for evaluating latency in real-case usage, where the problems of tail-word-latency no longer exist. Additionally, the intuition behind the latency introduced in this work is similar to the True Latency.

We added a discussion on the paper in Section 2.2 (lines 204-216).

> W3: Regarding SoftSegmenter
>
> SoftSegmenter should be explained in the main text, if the authors want to include it as one of the contributions (as currently is for the abstract and introduction)
>
> The name is quite misleading: SoftSegmenter still considers the exact match on the surface level rather than semantic, so maybe the name "soft" is not suitable.

We moved the description into the main text, see Section 3.2 (lines 341-368). SoftSegmenter uses a soft matching between tokens based on character-level similarity, hence "Soft".