SimulS2S-LLM: Unlocking Simultaneous Inference of Speech LLMs for Speech-to-Speech Translation

Keqi Deng[1], Wenxi Chen[2], Xie Chen[2], Philip C. Woodland[1]
[1]Department of Engineering, University of Cambridge, Trumpington St., Cambridge, UK.
[2]Shanghai Jiao Tong University, Shanghai, China
{kd502, pw117}@cam.ac.uk

Abstract

Simultaneous speech translation (SST) outputs translations in parallel with streaming speech input, balancing translation quality and latency. While large language models (LLMs) have been extended to handle the speech modality, streaming remains challenging as speech is prepended as a prompt for the entire generation process. To unlock LLM streaming capability, this paper proposes SimulS2S-LLM, which trains speech LLMs offline and employs a test-time policy to guide simultaneous inference. SimulS2S-LLM alleviates the mismatch between training and inference by extracting boundary-aware speech prompts that allows it to be better matched with text input data. SimulS2S-LLM achieves simultaneous speech-to-speech translation (Simul-S2ST) by predicting discrete output speech tokens and then synthesising output speech using a pre-trained vocoder. An incremental beam search is designed to expand the search space of speech token prediction without increasing latency. Experiments on the CVSS speech data show that SimulS2S-LLM offers a better translation quality-latency trade-off than existing methods that use the same training data, such as improving ASR-BLEU scores by 3 points at similar latency.

1 Introduction

Simultaneous speech translation converts input speech into translation output before the speech input utterance ends, enabling low-latency interaction (Zhang et al., 2024). The translation can be text or speech, classified as Simul-S2TT or Simul-S2ST. Conventional Simul-S2ST use a cascaded approach that includes automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) (Nakamura et al., 2006). However, cascaded methods suffer from error propagation and hinder joint optimisation (Deng and Woodland, 2024b). In Simul-S2ST the model must decide when to emit translation tokens from incomplete speech input, which is challenging due to the continuous nature and the uncertain duration of spoken data[1]. Recent work has begun exploring end-to-end (E2E) Simul-S2ST (Ma et al., 2024a; Zhang et al., 2024; Barrault et al., 2023). However, leveraging large language models (LLMs), known for their remarkable performance across a wide range of tasks, in Simul-S2ST remains a challenge.

Text-based LLMs have shown widespread success (Brown et al., 2020; Touvron et al., 2023; Dubey et al., 2024; Ouyang et al., 2022) and have been extended to handle speech (Chu et al., 2023; Tang et al., 2024; Deng et al., 2025), by prepending the speech as a prompt for LLM output generation and conditioning the LLM on the speech prompts. However, this decoder-only architecture struggles with streaming, since all of the speech prompt is prepended beforehand, and all subsequent generated output attends to the speech prompts (Chen et al., 2024). Therefore, online modifications must rely on previously obtained speech-text alignments to limit the speech accessible for each text token (Seide et al., 2024; Tsunoo et al., 2024).

To address these challenges and enable the use of speech LLMs for simultaneous speech translation, this paper proposes SimulS2S-LLM, which to the best of our knowledge is the first work to apply LLMs for Simul-S2ST. Moreover, SimulS2S-LLM aims to avoid restricting speech LLMs to specific streaming tasks, achieved via offline training. SimulS2S-LLM adopts a test-time Wait-k strategy (Ma et al., 2018) during inference to achieve simultaneous translation, allowing it to use only limited speech input as prompts to generate predicted translations. To alleviate the training-testing mismatch caused by offline training, SimulS2S-LLM leverages a continuous integrate and fire (CIF) mech-

[1]Simultaneous inference/translation means both speech input and output are streamed, in contrast to work that only streams speech generation based on complete input speech.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16718–16734
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

anism (Dong and Xu, 2020) to extract a token boundary-aware speech prompt from the streaming encoder input. For Simul-S2ST, SimulS2S-LLM predicts target-language discrete speech tokens based on the LLM hidden states and then synthesising output speech in the target language using a pre-trained vocoder. An incremental beam search is introduced to expand the search space while avoiding additional latency. The system is trained in an end-to-end fashion with a fixed text LLM. The proposed SimulS2S-LLM was evaluated on a Common Voice-based Speech-to-Speech (CVSS) translation corpus, showing improved quality-latency trade-offs compared to existing Simul-S2ST methods, despite being trained offline.

The main contributions of the paper are listed below:

- SimulS2S-LLM, to our knowledge, is the first work extending LLMs to Simul-S2ST.
- With boundary-aware speech prompts, a novel offline training method is proposed, unlocking the Simul-S2ST capabilities of speech LLMs without restricting them to certain streaming tasks, aligning with the expectations of LLMs.
- Based on LLM multi-layer hidden states, incremental beam search is designed to expand the prediction search space of speech tokens.
- Extensive experiments were conducted, including comparisons of different methods to extract speech prompts.

2 Related Work

2.1 Simultaneous Speech Translation

Existing simultaneous speech translation methods focus on speech-to-text translation (Simul-S2TT), which can be divided into fixed and flexible policies. Wait-k is a typical fixed read-write policy that was initially proposed for text machine translation (Ma et al., 2018) and then extended to speech translation (Ma et al., 2020c; Ren et al., 2020; Zeng et al., 2021; Dong et al., 2022). Furthermore, many studies have also explored flexible policy approach, including monotonic multi-head attention (MMA) (Ma et al., 2020b), the CIF-based method (Chang and Lee, 2022), neural transducers (Xue et al., 2022), and its variants (Deng and Woodland, 2024b; Liu et al., 2021; Tang et al., 2023). These methods train the model in a streaming manner, enabling it to decide when to emit translation tokens on the fly. Recently, some studies have explored using offline-trained attention-based encoder-decoder models for simultaneous inference (Liu et al., 2020; Papi et al., 2023a,b), such as determining whether to output translations based on attention scores (Papi et al., 2023a). However, this strategy may pose challenges when applied to decoder-only architectures due to the reliance solely on self-attention.

2.2 Direct Speech-to-Speech Translation

Recent advancements in direct speech-to-speech translation have been driven by the use of discrete speech tokens, extracted from self-supervised pre-trained models such as HuBERT (Hsu et al., 2021). The target-language discrete speech tokens are used as the training objective and vocoders are used to synthesise speech (Lee et al., 2022). Inaguma et al. (2023a) first transforms the source speech into hidden text states in the target language, based on which the target discrete speech tokens are generated. Dong et al. (2024) uses cross-lingual LMs to convert source semantic tokens into target semantic tokens, which are then used to predict target acoustic tokens for speech generation. Similarly, Le et al. (2024) jointly predicts the target-language text and residual vector quantisation codes.

Direct speech-to-speech translation is already very challenging, and performing it simultaneously (Simul-S2ST) requires the translation to be generated based on incomplete source speech and is therefore a still harder task. StreamSpeech (Zhang et al., 2024) uses connectionist temporal classification (CTC) (Graves et al., 2006) to align the source speech with the source text and target text, which are then used to guide simultaneous inference. It employs multi-task training, including ASR and Simul-S2TT tasks, to help in Simul-S2ST training. (Zhao et al., 2024) uses a neural transducer model (Graves, 2012) to predict target-language discrete speech tokens from source speech. However, there is still a lack of research that effectively leverages powerful LLMs for Simul-S2ST.

2.3 Speech Large Language Models

LLMs have achieved success (Achiam et al., 2023; Scao et al., 2022) and have been applied to text-based simultaneous translation (Koshkin et al., 2024a,b). Several studies have extended LLMs to handle speech input (Chu et al., 2023; Zhang et al.,

2023). Speech LLMs[2] can be divided into two categories (Cui et al., 2024). The first uses discrete speech tokens to extend the LLM vocabulary and build spoken generative LMs (Zhang et al., 2023; Borsos et al., 2023; Wang et al., 2023). The second category uses continuous speech representations as the prompt to condition LLMs (Chu et al., 2023; Deng et al., 2025; Fathullah et al., 2024; Wu et al., 2023; Yu et al., 2024; Chen et al., 2023; Huang et al., 2024). This paper falls into the second category, as previous work (Fang et al., 2024) has shown that this approach can effectively leverage off-the-shelf text-based LLMs for efficient training.

With the advent of GPT-4o, speech-to-speech LLMs have attracted more attention and given rise to a series of models such as SpeechGPT (Zhang et al., 2023). Mini-Omni (Xie and Wu, 2024) introduced parallel generation of text and audio, allowing models to initiate reasoning directly in audio. Llama-Omni (Fang et al., 2024) generates semantic speech tokens based on text hidden states. Moshi (Défossez et al., 2024) leverages both acoustic and semantic speech tokens to simultaneously model the input and output streams, enabling full-duplex operation. LSLM (Ma et al., 2024b) introduces a simplified way to achieve full-duplex operation using only semantic tokens. There are also some multi-modal generative LLMs, such as AnyGPT (Zhan et al., 2024). Our work differs by focusing on simultaneous inference, predicting target speech from incomplete source speech. In contrast, prior work (Fang et al., 2024; Xie and Wu, 2024; Ma et al., 2024b) supports streaming speech generation but needs complete speech inputs or segments with sufficient information, whereas Simul-S2ST requires low latency and is thus more challenging.

3 SimulS2S-LLM

SimulS2S-LLM, as shown in Fig. 1, uses a streaming encoder to extract a boundary-aware speech prompt from the source speech and pre-pends it before the embeddings ($\boldsymbol{z}_0 \cdots \boldsymbol{z}_N$) of LLM text token input ($[sos], \cdots, y_N$), which follows a decoder-only architecture. The multi-layer hidden states of the LLM are weighted and summed, and the discrete output speech tokens ($s_1 \cdots s_L$) are predicted in a streaming manner. SimulS2S-LLM is trained in an offline[3] manner, so that SimulS2S-LLM retains the potential to be applied to other non-streaming tasks. After training, SimulS2S-LLM directly performs simultaneous inference for Simul-S2ST.

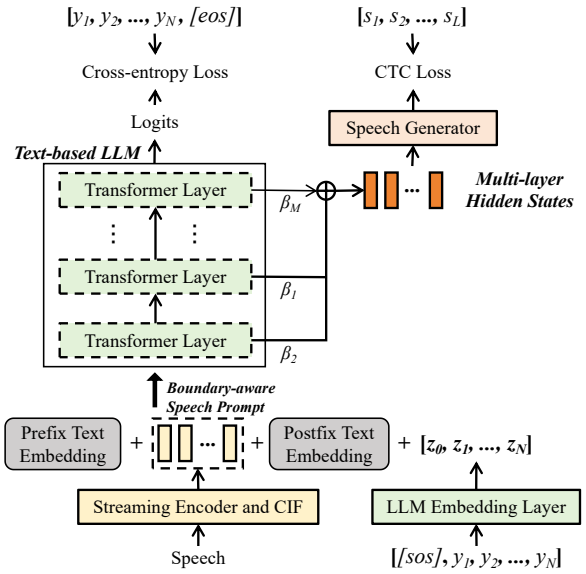


Figure 1: Illustration of SimulS2S-LLM offline training. $\oplus$ denotes addition. Prefix and postfix text contain template instructions. The hidden states of each LLM layer are weighted (β_i) and summed.

3.1 SimulS2S-LLM Architecture

SimulS2S-LLM contains four main modules: a streaming acoustic encoder, a CIF module, a text-based LLM, and a streaming speech generator. The encoder and the CIF are used to extract the boundary-aware speech prompts in a streaming manner, which is then fed into the text-based LLM along with other prompt templates that contain task instructions, i.e. the prefix and postfix text in Fig. 1, which are used to condition the text generation.

Training with teacher-forcing in LLMs can introduce a mismatch between training and testing, which may particularly affect the last-layer hidden state due to its focus on semantic information (Chang et al., 2023). To address this, this paper employs a weighted sum of multi-layer LLM hidden states. Specifically, denote $\boldsymbol{h}_i^m$ as the hidden state of the m-th layer at the i-th step, the weighted sum is obtained with trainable weights β_m:

$$\boldsymbol{h}_i = \beta_1 \cdot \boldsymbol{h}_i^1 + \cdots + \beta_m \cdot \boldsymbol{h}_i^m \quad (1)$$

The hidden states $\boldsymbol{h}_i$ are fed into a streaming speech generator, which uses causal Transformer

[2]Further analysis of speech LLMs refers to (Fathullah et al., 2024; Deng et al., 2025).

[3]Offline training refers to training where speech LLMs are not streaming-based, meaning that during training, the prediction of all tokens can attend to the entire speech input. This is independent of whether a streaming encoder is used.

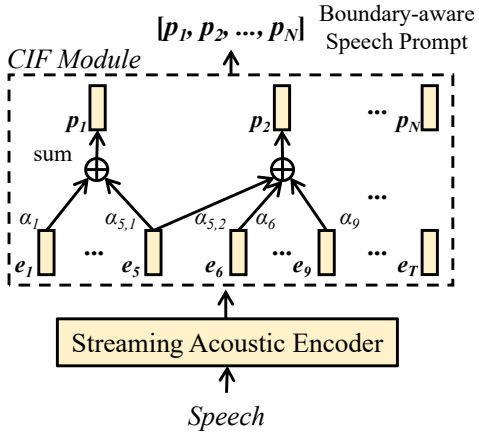


Figure 2: Illustration of the boundary-aware speech prompt extraction using the CIF. $\oplus$ denotes addition.

layers to enable streaming. SimulS2S-LLM employs semantic speech tokens as targets, with the predicted tokens passed to a vocoder to synthesise speech in the target language. Inspired by (Saharia et al., 2020; Fang et al., 2024), SimulS2S-LLM up-samples $\boldsymbol{h}_i$ and applies a simple CTC objective to align the target speech tokens.

3.2 Boundary-aware Speech Prompt

The challenges faced by SimulS2S-LLM mainly stem from the mismatch between offline training and simultaneous inference. This paper does not focus on scenarios with extremely low latency (e.g., AL < 1 s in Simul-S2TT), as prior work (Deng and Woodland, 2024b) indicates these are unsuitable for offline-trained models and compromise translation quality due to the need for re-ordering.

Preliminary experiments showed that simply using down-sampling methods, such as stacking encoder outputs (Fathullah et al., 2024; Yu et al., 2024; Ma et al., 2024c) to obtain speech prompts, leads to poor performance during simultaneous inference after offline training. However, inspired by the fact that the test-time wait-k strategy works well in text-based simultaneous machine translation (Gu et al., 2017; Ma et al., 2018), this paper proposes that the key to unlocking simultaneous inference for offline-trained speech LLMs is extracting boundary-aware speech prompts, which makes the system closer to the text-based scenario. Simple down-sampling during simultaneous decoding, where only partial speech prompts are available, ignores word boundary information and prevents the model from making correct predictions.

SimulS2S-LLM obtains the boundary-aware speech prompts using the CIF[4] mechanism (Dong and Xu, 2020), a non-autoregressive method that jointly learns alignments and high-level representations. To be more specific, a scalar weight α_t is learned for each encoder output frame $\boldsymbol{e}_t$, and the boundary-aware speech prompts $\boldsymbol{p}_i$ are obtained via weighted addition. Following (Deng and Woodland, 2024b), this paper simply uses the last dimension of $\boldsymbol{e}_t$ as the raw scalar attention value α_t to avoid additional parameters: $\alpha_t = \text{sigmoid}(e_{t,d})$, where d is the dimension size of $\boldsymbol{e}_t$. The weights α_t are accumulated from left to right (i.e., to support streaming) until the sum exceeds a threshold of 1.0. Once the threshold is reached, the current weight α_t is split into two parts $\alpha_{t,1}$ and $\alpha_{t,2}$: $\alpha_{t,1}$ ensures the accumulation of exactly 1.0, while $\alpha_{t,2}$ is used for the next integration. For instance, as shown in Fig. 2, if the threshold 1.0 is reached at $t = 5$, the boundary-aware speech prompts at the 1-st step can be obtained via: $\boldsymbol{p}_1 = \sum_{j=1}^{4} \alpha_j \cdot \boldsymbol{e}_{j,1:d-1} + \alpha_{5,1} \cdot \boldsymbol{e}_{5,1:d-1}$. The $\boldsymbol{p}_i$ will be mapped to the same dimension size as the LLM embedding size before being fed in. The accumulation is then reset to zero and conducted incrementally. To learn the CIF alignment, a quantity loss $\mathcal{L}_{\text{qua}} = |\sum_{j=1}^{T} \alpha_j - N|$ is calculated during training, guiding accumulated weights to align with the source text length (N).

3.3 Offline Training of SimulS2S-LLM

SimulS2S-LLM uses a two-stage training strategy. The first stage of training corresponds to the speech-to-text translation task. An off-the-shelf text-based LLM is used and kept fixed. The encoder and CIF are optimised under the supervision of the cross-entropy loss function as shown in Fig. 1, where the target-language text tokens $(y_1, \cdots, [eos])$ is used as the training target. In addition, the quantity loss is also considered:

$$\mathcal{L}_{\text{Train}}^{\text{First}} = \mathcal{L}_{\text{CE}} + \gamma \mathcal{L}_{\text{qua}} \tag{2}$$

Note the entire speech prompt is pre-pended to the input text embedding sequence $(\boldsymbol{z}_0 \cdots \boldsymbol{z}_N)$, enabling offline training. In addition, the template instructions that determine the translation task, e.g. "Translate the French text into English", are used in both the first and second-stage training.

In the second stage of training, only the layer-wise weights β_i and speech generator are updated

[4]The visualisation of CIF alignment refers to the supplementary materials of Deng and Woodland (2024a)

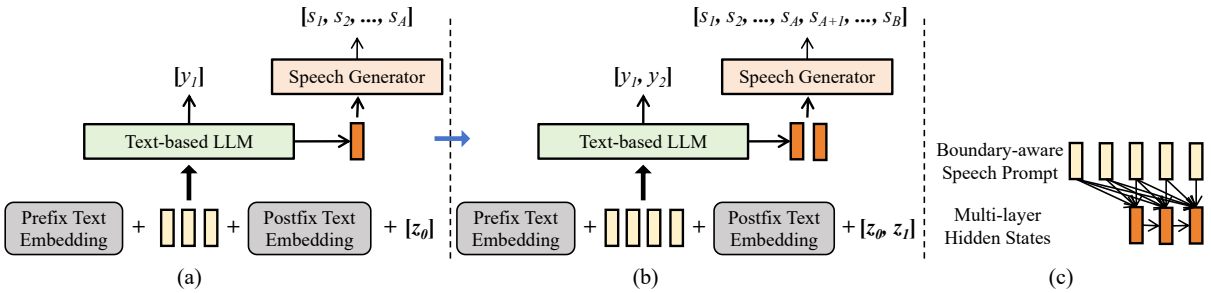


Figure 3: Illustration of the simultaneous inference of offline-trained SimulS2S-LLM (with wait-3 as an example) (a) the generation of the 1-st hidden state and corresponding speech tokens t_i; (b) 2-nd generation; (c) overall illustration of the hidden state generation order according to the speech prompt (always wait 3 more steps here).

under the supervision of the CTC loss, where the speech semantic tokens are used as the target.

3.4 Simultaneous Inference of SimulS2S-LLM

SimulS2S-LLM uses the wait-k strategy during testing to achieve simultaneous inference. The CIF module, along with the streaming encoder, extracts the speech prompts online, whose length is at the text token level. Therefore, the inference process is determined and driven by CIF. When $k = 3$ in wait-k, as shown in Fig. 3, the hidden states corresponding to the translation generated by the LLM are always two steps behind the speech prompt. For example, in Fig. 3, a speech prompt of length 3 corresponds to the generation of the first LLM token (Fig. 3(a)), while a speech prompt of length 4 corresponds to the generation of the second LLM token (Fig. 3(b)). Note when a new speech prompt is obtained from newly received speech, past keys and values including positional information need to be updated accordingly before LLM generation. Once the entire speech input is loaded, the LLM is no longer constrained by the speech prompt length and completes the prediction auto-regressively, making use of tail beam search (Ma et al., 2018).

Whenever a new hidden state is generated, it is fed into the speech generator, where it undergoes up-sampling (e.g., by a factor of sampling rate U) before being passed into a causal Transformer layer. The speech generator then outputs new CTC logits. To mitigate the independence assumption of CTC, a speech token-based n-gram LM is built to assist the CTC frame-synchronous decoding via shallow fusion. To expand the search space without introducing additional latency, an incremental beam search is designed. Specifically, within the range of new CTC logits (i.e., of length U, which is the up-sampling rate)), decoding is performed frame by frame using beam search. After decoding the final frame, only the highest-probability hypothesis is retained, while other hypotheses are pruned. For example, the predicted speech tokens ($s_1 \cdots s_A$) in Fig. 3(a) are the prefixes of the predicted speech tokens in Fig. 3(b). Note this pruning is no longer needed once the input speech has been fully loaded.

This paper uses a chunk-based mask operation to implement the streaming Transformer encoder. Therefore, at inference, the speech input is loaded chunk by chunk, and CIF continues to accumulate α_t based on the newly read speech chunk, dynamically generating new speech prompts. Before the entire speech input is read, the number of new LLM tokens generated for each new speech chunk is still determined by wait-k, i.e. the total length of LLM tokens remains k shorter than the latest speech prompt length. If multiple LLM tokens can be generated within a single speech chunk, beam search is used to expand the search space. After generating the last LLM token in each chunk, only the highest-probability hypothesis is retained to implement pruning, avoiding additional latency.

The detailed procedure for this simultaneous inference is shown in Algorithm 1, where the inputs

Algorithm 1 SimulS2S-LLM Inference

Input: $\mathbf{E}_{:(n+1)*c}$, $\boldsymbol{y}$, $L_{\max}$, K, *Final*
Output: $\boldsymbol{s}^{\text{gen}}$

$L_{\text{prev}} \leftarrow \text{len}(\boldsymbol{y})$: Get the previous token $\boldsymbol{y}$ length L_{prev}
$L_p, \mathbf{p} \leftarrow \text{CIF}(\mathbf{E}_{:(n+1)*c})$: Get speech prompt $\mathbf{p}$ and its length L_p with the input chunks of speech $\mathbf{E}_{0:(n+1)*c}$
if *Final* **then**
    $L_{\text{gen}} \leftarrow L_{\max}$: Set the new token number L_{gen} to the max length $L_{\max}$ if the input is the final complete one
else
    $L_{\text{gen}} \leftarrow (L_p - L_{\text{prev}} - K + 1)$: Guided by wait-k
if $L_{\text{gen}} <= 0$ **then**
    return
$\boldsymbol{y}^{\text{gen}}, \mathbf{h}^{\text{gen}} \leftarrow \text{LLM}(\mathbf{p}, L_{\text{gen}})$: New tokens $\boldsymbol{y}^{\text{gen}}$ and hidden states $\mathbf{h}^{\text{gen}}$ based on $\mathbf{p}$ and length constraint L_{gen}
$\boldsymbol{s}^{\text{gen}} \leftarrow \text{Speech-Generator}(\mathbf{h}^{\text{gen}})$: New speech tokens
return $\boldsymbol{s}^{\text{gen}}$

are the streaming speech encoder output $\mathbf{E}_{:(n+1)*c}$ (with a chunk size of c), previously predicted tokens $\boldsymbol{y}$, the maximum generation length L_{max}, the wait-k K steps, and whether the input speech is now a complete utterance, denoted *Final*. Then the newly predicted speech semantic tokens $\boldsymbol{s}^{\text{gen}}$ $(s_1 \cdots s_A \cdots)$ will be returned. Note that the functions LLM and Speech-Generator in Algorithm 1 have recorded the past keys and values in a cache, so only the speech content of the current chunk is needed to complete the generation.

4 Experimental Setup

4.1 Dataset

Experiments were conducted on CVSS-C data (Jia et al., 2022b), which is a large-scale speech-to-speech translation data created from the CoVoST 2 (Wang et al., 2021) speech-to-text translation dataset with synthesised target speech. SimulS2S-LLM was evaluated on Spanish-English (Es-En), French-English (Fr-En), and German-English (De-En) pairs. Additional details about the data are provided in Appendix A.

4.2 Model Descriptions

Speech semantic tokens were extracted from the target speech using mHuBERT (Popuri et al., 2022). Based on the training set, target speech token-based 4-gram LMs were obtained using KenLM toolkit, which was incorporated into CTC decoding with 0.5 weight. The raw speech waveform was used as input. For Es-En and Fr-En, BLOOMZ-7B1 (Scao et al., 2022) was used as the text-based LLM and kept fixed all the time. For De-En, Llama3-8B (Dubey et al., 2024) was used as BLOOMZ underperforms in German. A pre-trained unit-based HiFi-GAN vocoder (Kong et al., 2020) was used to synthesise speech. To achieve streaming speech generation, partially predicted speech tokens are directly sent to the vocoder. The resulting audio signal is used as the prefix for the next prediction and will no longer be modified.

All models built in this paper used the same streaming Transformer encoder, fine-tuned from the "xlsr_53_56k" model provided by Fairseq (Ott et al., 2019), with a chunk-based masking operation. The chunk size was set to 32, corresponding to a theoretical average latency of 320 ms. More details can be found in Appendix B.

SimulS2S-LLM In addition to the encoder, as mentioned in Sec. 3.2, the CIF module only involves a fully-connected (FC) layer to map the speech prompt dimension to the LLM embedding dimension, i.e. 4096. The speech generator consists of 8 causal Transformer layers (1024 attention dimension, 2048 feed-forward dimension, and 8 heads), which use subsequent masks to avoid seeing future information. The up-sampling rate U was set to 25. The beam size of the incremental beam search was set to 10.

Boundary-unaware SimulS2S-LLM A fixed down-sampling method (Fathullah et al., 2024) was implemented to extract a boundary-unaware speech prompt for comparison with SimulS2S-LLM. This model, referred to as boundary-unaware SimulS2S-LLM, serves as the baseline model. Following (Fathullah et al., 2024), considering the frame stride of the encoder was 20 ms, every 16 consecutive acoustic encoder output frames were stacked to achieve down-sampling. Then, an additional FC layer was applied to map the stacked encoder outputs to the LLM embedding dimension (i.e., 4096) before feeding them into the LLM. This model also used the wait-k policy for inference, with every fixed 16 encoder outputs used as one step.

StreamSpeech StreamSpeech (Zhang et al., 2024) is a recent Simul-S2ST model that has achieved state-of-the-art (SOTA) results, with the same speech tokens and vocoder used as in SimulS2S-LLM. Note that it is not an LLM-based approach and is used to provide a benchmark result.

4.3 Metrics

Experiments were implemented based on the ESPnet-ST (Inaguma et al., 2020). SimulEval (Ma et al., 2020a) was used to evaluate the models.

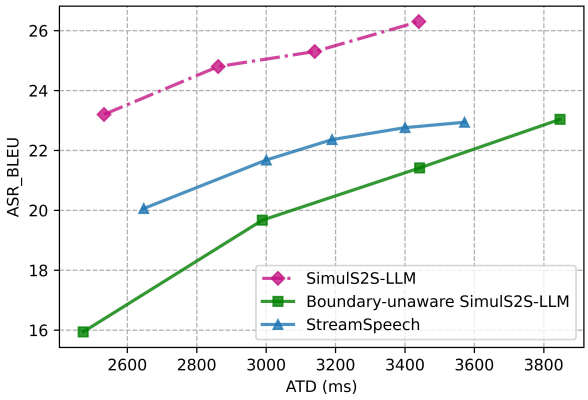
ASR-BLEU For Simul-S2ST, the ASR-BLEU toolkit[5] was used to evaluate the translation quality, which transcribes the synthesised speech into text before calculating SacreBLEU (Post, 2018) with the reference text.

ATD Following the Simuleval example[6], the average token delay (ATD) (Kano et al., 2022) was used to measure the speech generation latency. ATD refers to the average delay between output sub-segments and corresponding input sub-segments.

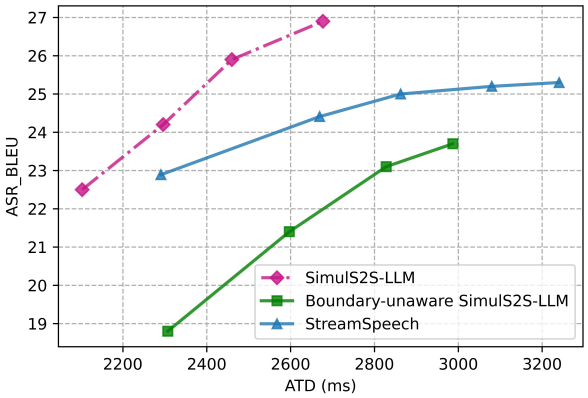
Text output Simul-S2TT was also evaluated, and the translation quality was measured using Sacre-

[5] https://github.com/facebookresearch/fairseq/tree/ust/examples/speech_to_speech/asr_bleu

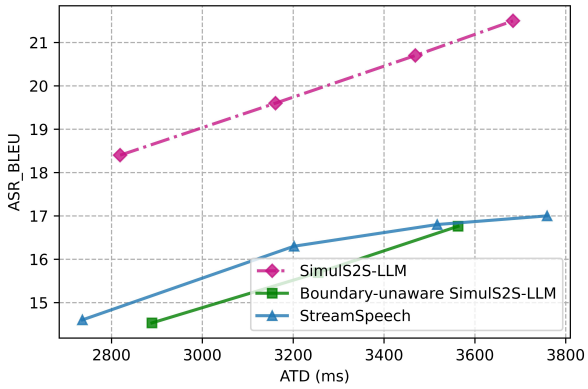
[6] https://github.com/facebookresearch/SimulEval/tree/main/examples/speech_to_speech



Simul-S2ST Es-En



Simul-S2ST Fr-En



Simul-S2ST De-En

Figure 4: Simul-S2ST quality-latency trade-off curves on CVSS-C Es-En, Fr-En, and De-En test sets. The x-axis represents the latency, measured by ATD, and the y-axis represents the speech translation quality, measured by ASR-BLEU. Note that the y-axis scales can be different on different sub-figures.

BLEU. The speech version of the word-level Average Lagging (AL) (Ma et al., 2018, 2020c) was used to measure latency.

5 Experimental Results

This section compares the proposed SimulS2S-LLM with existing methods, such as StreamSpeech, as well as boundary-unaware SimulS2S-LLM. Ablation studies were conducted to evaluate the effectiveness of boundary-aware speech prompts, utilising multi-layer hidden states and speech token generation. Note, in order to ensure high translation quality, SimulS2S-LLM does not target scenarios requiring extremely low latency.

S2ST Models	Es-En	Fr-En	De-En
Offline			
S2UT	18.53	22.23	–
Translatotron	8.72	16.96	–
Translatotron 2	22.93	26.07	16.91
DASpeech	21.37	25.03	16.14
UnitY	24.95	27.77	18.74
Offline StreamSpeech	27.25	28.45	20.93
Streaming			
StreamSpeech	22.94	25.30	17.0
SimulS2S-LLM	26.33	26.93	21.5

Table 1: ASR-BLEU (↑) results on the CVSS-C data for different models, including S2UT (Lee et al., 2022), Translatotron (Jia et al., 2019), Translatotron 2(Jia et al., 2022a), DASpeech (Fang et al., 2023), UnitY (Inaguma et al., 2023b). The SimulS2S-LLM results correspond to the last points in Fig. 4. Note the comparisons are not well-controlled. The published benchmark results are reproduced from Zhang et al. (2024) on CVSS-C.

5.1 Simul-S2ST Results

Figure 4 shows the Simul-S2ST results on CVSS-C Es-En, Fr-En, and De-En data, with the ASR-BLEU scores plotted against ATD. Although SimulS2S-LLM was trained offline, it still clearly outperforms the strong StreamSpeech models with simultaneous inference. For example, on the Es-En test set, SimulS2S-LLM outperformed StreamSpeech by approximately 4 ASR-BLEU points while maintaining the same latency. This demonstrates that SimulS2S-LLM can effectively leverage the strong LLM generation capabilities in a streaming manner. Previous work has shown that text-based LLMs can be extended to speech with strong performance across a wide range of tasks, such as translation and question answering (Tang et al., 2024; Chu et al., 2023). SimulS2S-LLM further unlocks simultaneous inference while potentially retaining these emergent abilities by following the same training paradigm. Additionally, the boundary-unaware version of SimulS2S-LLM failed to achieve such strong performance, in line with our expectation that learning boundary-aware speech prompts can unlock the simultaneous in-

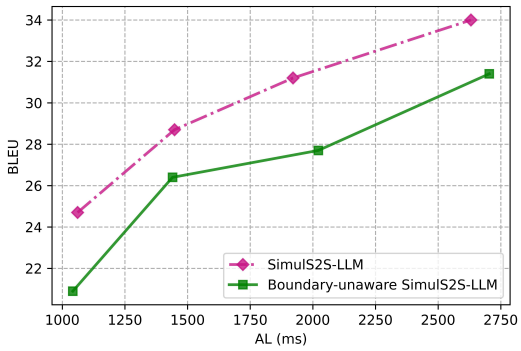


Figure 5: Simul-S2TT quality-latency trade-off curves on CVSS-C Es-En test set. The x-axis represents the latency, measured by AL, and the y-axis represents the speech translation quality, measured by BLEU.

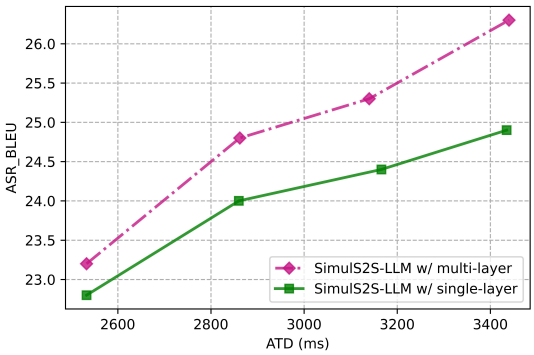


Figure 6: Simul-S2ST Es-En results for SimulS2S-LLM with single or multi-layer LLM hidden states.

ference abilities of offline-trained speech LLMs. With a boundary-aware speech prompt, SimulS2S-LLM shares more similarities with text-based simultaneous translation, where test-time wait-k is commonly used. Hence, the extensive comparisons with both StreamSpeech and the boundary-unaware SimulS2S-LLM provide strong evidence for the superiority of our method. The numerical results in Fig. 4 are given in Appendix C and extended computation-aware results are shown in Appendix D.

Table 1 compares streaming SimulS2S-LLM with published speech-to-speech translation (S2ST) results from the literature on CVSS-C, showing that SimulS2S-LLM achieves competitive performance as a streaming model. In the streaming scenario, the results for SimulS2S-LLM and StreamSpeech are represented by the last points in Fig. 4.

Appendix E compares with SOTA models like SeamlessStreaming (Barrault et al., 2023) which uses 9,300 hours of speech-to-speech data in contrast to between 69.5 and 174 hours of speech-to-speech data for individual language pairs used here.

5.2 Ablation on Speech Prompt Type for Simul-S2TT

This sub-section compares SimulS2S-LLM and boundary-unaware SimulS2S-LLM on the Simul-S2TT task. Since the only difference between them is the speech prompt used, this comparison can effectively evaluate the importance of boundary-aware speech prompts in unlocking simultaneous inference. As shown in Fig. 5, SimulS2S-LLM consistently outperformed the boundary-unaware one. For example, with similar latency, the boundary-aware SimulS2S-LLM was about 4 BLEU points higher. Hence, the experimental results on both the Simul-S2ST and Simul-S2TT tasks demonstrate the importance of using boundary-aware speech prompts for offline-trained speech LLMs. The numerical results in Fig. 5 are given in Appendix F

Moreover, by comparing the differences in translation quality between SimulS2S-LLM and boundary-unaware SimulS2S-LLM in Fig. 4 and Fig. 5, it can be observed that the gap is similar, with the ASR-BLEU and BLEU values both differing by around 4 points at similar latencies. Therefore, the main reason for the poorer performance of boundary-unaware SimulS2S-LLM is the prediction error of the LLM, which is in line with expectations as they use the same speech generator.

5.3 Ablation on Multi-layer Hidden States

An ablation study was conducted to evaluate the effectiveness of using multiple layers of LLM hidden states. Fig. 6 shows that leveraging LLM multi-layer hidden states is more beneficial for predicting speech tokens, causing about one ASR-BLEU point improvement. The final hidden layer focuses on semantic information (Chang et al., 2023), which is favourable for text token prediction, whereas multiple hidden states capture richer information. Moreover, due to the mismatch caused by teacher forcing in training, using multiple hidden states seems to be more robust.

5.4 Ablation on Speech Token Generation

This sub-section compares the use of n-gram and greedy search for predicting discrete speech tokens. As shown in Table 2, although discrete speech tokens are more challenging to predict than text units, n-gram LMs based on discrete speech tokens can still assist CTC in making more accurate predictions, thus improving the translated speech quality.

Models	ASR-BLEU	ATD (ms)
SimulS2S-LLM w/ n-gram	26.3	3440
SimulS2S-LLM w/ greedy	24.7	3439

Table 2: ASR-BLEU (↑) results on the CVSS-C Es-En data for different speech token generation methods.

S2ST Models	Unsupervised	QE	Ref
Offline UnitY	0.51	3.33	3.26
Offline StreamSpeech	0.52	3.37	3.35
SimulS2S-LLM	0.72	3.72	3.59

Table 3: BLASER 2.0 (↑) results on the CVSS-C Es-En data. The published results of offline UnitY and StreamSpeech are from Zhang et al. (2024).

5.5 Speech Evaluation with BLASER 2.0

This section further uses BLASER 2.0 (Dale and Costa-jussà, 2024) to evaluate the generated speech quality of SimulS2S-LLM. BLASER 2.0 includes three scores: Unsupervised (0–1), QE (1–5), and Ref (1–5). The Unsupervised version computes cosine similarity between sentence-level embeddings without supervision, while QE and Ref are supervised models trained to predict human ratings, with Ref additionally requiring reference target speech.

Table 3 results show that as a streaming model, SimulS2S-LLM gave higher BLASER 2.0 scores compared to the offline StreamSpeech and UnitY models on the CVSS-C benchmark data, demonstrating superior translation and speech quality.

6 Conclusions

This paper proposes SimulS2S-LLM, the first work to extend LLMs to Simul-S2ST while avoiding being constrained to specific streaming tasks via offline training. SimulS2S-LLM uses a test-time wait-k policy to guide the simultaneous inference. To alleviate offline training and simultaneous inference mismatch, SimulS2S-LLM extracts boundary-aware speech prompts based on CIF. To generate high-quality speech in streaming, multi-layer LLM hidden states are used by a causal Transformer-based speech generator to predict discrete speech tokens. To enhance this prediction process, an incremental beam search is designed to expand the search space of speech tokens without introducing additional latency, while a speech token-based n-gram LM is also incorporated. Experiments show that SimulS2S-LLM gives a better quality-latency trade-off than existing Simul-S2ST methods.

Limitations

This paper has the following limitations.

1. SimulS2S-LLM relies on off-the-shelf text-based LLMs, meaning that the performance is inherently constrained by the capabilities of the available text-based models. Due to limited computing resources, this paper focuses on using 7B/8B LLMs, as larger models are beyond our computational capacity. Additionally, SimulS2S-LLM is restricted to open-source LLMs and cannot use closed-source models like GPT-4.

2. This paper does not focus on scenarios with extremely low latency (e.g., AL $<$ 1 s in Simul-S2TT), as prior work (Deng and Woodland, 2024b) indicates these are unsuitable for offline-trained models and compromise translation quality due to the need for re-ordering. Moreover, using LLMs increases the computational load, which leads to higher latency when considering computation time. This is a common challenge faced by the community, and significant research development is needed to accelerate LLM inference speed. As such, this aspect is beyond the scope of this paper and is left as future work. In addition, since this is the first work to apply LLM to the Simul-S2ST task, we couldn't find an LLM-based method to compare with SimulS2S-LLM on the Simul-S2ST task.

3. As mentioned in Section 3.4, during the simultaneous inference of the proposed SimulS2S-LLM, when a new speech chunk is read in, the past keys and values need to be updated before LLM generation. However, according to the analysis in Appendix D, the time consumed by this process should not be significant as it can be performed in parallel. This paper has also not evaluated SimulS2S-LLM on long-form Simul-S2ST due to the lack of data.

4. Due to limitations in training data and computing resources, we were unable to train our SimulS2S-LLM as extensively as some foundation models like SeamlessStreaming (Barrault et al., 2023) which uses 9,300 hours of speech-speech data. However, we conducted comprehensive experiments across three language pairs and CVSS-C is the most widely used speech-to-speech translation data set,

even though the individual language pairs have only between 69.5 and 174 hours of speech-to-speech data. In addition, if more speech-to-speech translation data is used, the speech generation performance of SimulS2S-LLM can be expected to greatly improve.

5. This paper evaluates SimulS2S-LLM on three European language pairs, each in a single translation direction (i.e., Es-En, Fr-En, and De-En). While we believe the technique can be extended to other languages, including non-European ones, and additional translation directions, its performance in these cases remains unverified and is left for future work. In addition, simultaneous translation varies in difficulty for different language pairs due to the extent of re-ordering, so achieving Simul-S2ST for certain language pairs can be challenging.

6. This paper evaluates SimulS2S-LLM only on simultaneous speech translation tasks, including Simul-S2TT and Simul-S2ST. Although it claims that SimulS2S-LLM avoids constraining speech LLMs to specific streaming tasks through offline training, it does not directly evaluate its performance on other simultaneous inference tasks with speech as the input or on offline inference tasks. This is because preserving the zero-shot task capabilities of speech LLMs is not the main focus of this paper and has already been extensively studied in prior work.

Ethics Statement

Deep learning systems are data-hungry, and without sufficient data, it is difficult to achieve promising model performance. For under-resourced languages or domains, this issue will be even more severe. This can lead to a poor user experience for minority groups, resulting in their views being underrepresented or misunderstood. The SimulS2S-LLM technique proposed in this paper can alleviate this issue by translating low-resource data into high-resource data in a low-latency manner, enabling the model to better handle the task.

Acknowledgments

Keqi Deng is funded by the Cambridge Trust. This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/T022159/1.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. AudioLM: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proc. NeurIPS*, volume 33, Online.

Chih-Chiang Chang and Hung-yi Lee. 2022. Exploring continuous integrate-and-fire for adaptive simultaneous speech translation. In *Proc. Interspeech*, Incheon, Korea.

Xuankai Chang, Brian Yan, Yuya Fujita, Takashi Maekaku, and Shinji Watanabe. 2023. Exploration of efficient end-to-end ASR using discretized input from self-supervised learning. In *Proc. Interspeech*, Dublin, Ireland.

Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. 2023. X-LLM: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv preprint arXiv:2305.04160*.

Zhehuai Chen, He Huang, Oleksii Hrinchuk, Krishna C Puvvada, Nithin Rao Koluguri, Piotr Żelasko, Jagadeesh Balam, and Boris Ginsburg. 2024. BESTOW: Efficient and streamable speech language model with the best of two worlds in GPT and T5. *arXiv preprint arXiv:2406.19954*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Wenqian Cui, Dianzhi Yu, Xiaoqi Jiao, Ziqiao Meng, Guangyan Zhang, Qichao Wang, Yiwen Guo, and Irwin King. 2024. Recent advances in speech language models: A survey. *arXiv preprint arXiv:2410.03751*.

David Dale and Marta R. Costa-jussà. 2024. BLASER 2.0: a metric for evaluation and quality estimation of massively multilingual speech and text translation. In *Proc. EMNLP (Findings)*, Miami, Florida, USA.

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. *arXiv preprint arXiv:2410.00037*.

Keqi Deng, Guangzhi Sun, and Philip C Woodland. 2025. Wav2Prompt: End-to-end speech prompt learning and task-based fine-tuning for text-based LLMs. In *Proc. NAACL*, Albuquerque, New Mexico, USA.

Keqi Deng and Philip C. Woodland. 2024a. Label-synchronous neural transducer for adaptable online E2E speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3507–3516.

Keqi Deng and Philip C Woodland. 2024b. Label-synchronous neural transducer for E2E simultaneous speech translation. In *Proc. ACL*, Bangkok, Thailand.

Linhao Dong and Bo Xu. 2020. CIF: Continuous integrate-and-fire for end-to-end speech recognition. In *Proc. ICASSP*, Barcelona, Spain.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. Learning when to translate for streaming speech. In *Proc. ACL*, Dublin, Ireland.

Qianqian Dong, Zhiying Huang, Qi Tian, Chen Xu, Tom Ko, Yunlong Zhao, Siyuan Feng, Tang Li, Kexin Wang, Xuxin Cheng, Fengpeng Yue, Ye Bai, Xi Chen, Lu Lu, Zejun Ma, Yuping Wang, Mingxuan Wang, and Yuxuan Wang. 2024. Polyvoice: Language models for speech to speech translation. In *Proc. ICLR*, Vienna, Austria.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2024. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*.

Qingkai Fang, Yan Zhou, and Yang Feng. 2023. Daspeech: Directed acyclic transformer for fast and high-quality speech-to-speech translation. In *Proc. NeurIPS*.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024. Prompting large language models with speech recognition abilities. In *Proc. ICASSP*, Seoul, Korea.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *ArXiv*, abs/1211.3711.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, Pittsburgh, Pennsylvania.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proc. EACL*, Valencia, Spain.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Chien-yu Huang, Ke-Han Lu, Shih-Heng Wang, Chi-Yuan Hsiao, Chun-Yi Kuan, Haibin Wu, Siddhant Arora, Kai-Wei Chang, Jiatong Shi, Yifan Peng, et al. 2024. Dynamic-Superb: Towards a dynamic, collaborative, and comprehensive instruction-tuning benchmark for speech. In *Proc. ICASSP*, Seoul, Korea.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. ESPnet-ST: All-in-one speech translation toolkit. In *Proc. ACL (demo)*, Seattle, Washington, USA.

Hirofumi Inaguma, Sravya Popuri, Ilia Kulikov, Peng-Jen Chen, Changhan Wang, Yu-An Chung, Yun Tang, Ann Lee, Shinji Watanabe, and Juan Pino. 2023a. UnitY: Two-pass direct speech-to-speech translation with discrete units. In *Proc. ACL*, Toronto, Canada.

Hirofumi Inaguma, Sravya Popuri, Ilia Kulikov, Peng-Jen Chen, Changhan Wang, Yu-An Chung, Yun Tang, Ann Lee, Shinji Watanabe, and Juan Pino. 2023b. UnitY: Two-pass direct speech-to-speech translation with discrete units. In *Proc. ACL*, Toronto, Canada.

Ye Jia, Michelle Tadmor Ramanovich, Tal Remez, and Roi Pomerantz. 2022a. Translatotron 2: High-quality direct speech-to-speech translation with voice preservation. In *Proc. ICML*, Baltimore, USA.

Ye Jia, Michelle Tadmor Ramanovich, Quan Wang, and Heiga Zen. 2022b. CVSS corpus and massively multilingual speech-to-speech translation. In *Proc. LREC*, Marseille, France.

Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu. 2019. Direct speech-to-speech translation with a

sequence-to-sequence model. In *Proc. Interspeech*, Graz, Austria.

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

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Proc. NeurIPS*, Online.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024a. LLMs are zero-shot context-aware simultaneous translators. In *Proc. EMNLP*, Miami, USA.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024b. Transllama: Llm-based simultaneous translation system. In *Proc. EMNLP (Findings)*, Miami, USA.

Chenyang Le, Yao Qian, Dongmei Wang, Long Zhou, Shujie Liu, Xiaofei Wang, Midia Yousefi, Yanmin Qian, Jinyu Li, Sheng Zhao, et al. 2024. TransVIP: Speech to speech translation system with voice and isochrony preservation. *arXiv preprint arXiv:2405.17809*.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Sravya Popuri, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, and Wei-Ning Hsu. 2022. Direct speech-to-speech translation with discrete units. In *Proc. ACL*, Dublin, Ireland.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In *Proc. EMNLP*, Punta Cana, Dominican Republic.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection. In *Proc. Interspeech*, Shanghai, China.

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. 2018. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proc. ACL*, Florence, Italy.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proc. EMNLP (Demos)*, Online.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020b. Monotonic multihead attention. In *Proc. ICLR*, Online.

Xutai Ma, Juan Miguel Pino, and Philipp Koehn. 2020c. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proc. AACL/IJCNLP*, Suzhou, China.

Zhengrui Ma, Yang Feng, and Min Zhang. 2024a. Learning monotonic attention in transducer for streaming generation. *arXiv preprint arXiv:2411.17170*.

Ziyang Ma, Yakun Song, Chenpeng Du, Jian Cong, Zhuo Chen, Yuping Wang, Yuxuan Wang, and Xie Chen. 2024b. Language model can listen while speaking. *arXiv preprint arXiv:2408.02622*.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al. 2024c. An embarrassingly simple approach for llm with strong asr capacity. *arXiv preprint arXiv:2402.08846*.

S. Nakamura, K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J.-S. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto. 2006. The ATR multilingual speech-to-speech translation system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):365–376.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: a fast, extensible toolkit for sequence modeling. In *Proc. NAACL-HLT (Demonstrations)*, Minneapolis, Minnesota.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Proc. NeurIPS*.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proc. 3rd AutoSimTrans*, Online.

Sara Papi, Matteo Negri, and Marco Turchi. 2023a. Attention as a guide for simultaneous speech translation. In *Proc. ACL*, Toronto, Canada.

Sara Papi, Marco Turchi, and Matteo Negri. 2023b. AlignAtt: Using attention-based audio-translation alignments as a guide for simultaneous speech translation. In *Proc. Interspeech*, Dublin, Ireland.

Sravya Popuri, Peng-Jen Chen, Changhan Wang, Juan Pino, Yossi Adi, Jiatao Gu, Wei-Ning Hsu, and Ann Lee. 2022. Enhanced direct speech-to-speech translation using self-supervised pre-training and data augmentation. In *Proc. Interspeech*, Incheon, Korea.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *WMT*, pages 186–191. Association for Computational Linguistics.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proc. ACL*, Online.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proc. EMNLP*, online.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Frank Seide, Morrie Doulaty, Yangyang Shi, Yashesh Gaur, Junteng Jia, and Chunyang Wu. 2024. Speech ReaLLM – real-time streaming speech recognition with multimodal LLMs by teaching the flow of time. *arXiv preprint arXiv:2406.09569*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. 2024. SALMONN: Towards generic hearing abilities for large language models. In *Proc. ICLR*, Vienna, Austria.

Yun Tang, Anna Sun, Hirofumi Inaguma, Xinyue Chen, Ning Dong, Xutai Ma, Paden Tomasello, and Juan Pino. 2023. Hybrid transducer and attention based encoder-decoder modeling for speech-to-text tasks. In *Proc. ACL*, Toronto, Canada.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. LLaMa 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Emiru Tsunoo, Hayato Futami, Yosuke Kashiwagi, Siddhant Arora, and Shinji Watanabe. 2024. Decoder-only architecture for streaming end-to-end speech recognition. *arXiv preprint arXiv:2406.16107*.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. CoVoST 2 and massively multilingual speech translation. In *Proc. Interspeech*, Brno, Czech Republic.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. On decoder-only architecture for speech-to-text and large language model integration. In *Proc. ASRU*, Taipei.

Zhifei Xie and Changqiao Wu. 2024. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*.

Jian Xue, Peidong Wang, Jinyu Li, Matt Post, and Yashesh Gaur. 2022. Large-scale streaming end-to-end speech translation with neural transducers. In *Proc. Interspeech*, Incheon, Korea.

Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. Connecting speech encoder and large language model for ASR. In *Proc. ICASSP*, Seoul, Korea.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Proc. ACL/IJCNLP (Findings)*, Online.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. 2024. AnyGPT: Unified multimodal LLM with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Proc. EMNLP (Findings)*, Singapore.

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 *Proc. ACL*, Bangkok, Thailand.

Jinzheng Zhao, Niko Moritz, Egor Lakomkin, Ruiming Xie, Zhiping Xiu, Katerina Zmolikova, Zeeshan Ahmed, Yashesh Gaur, Duc Le, and Christian Fuegen. 2024. Textless streaming speech-to-speech translation using semantic speech tokens. *arXiv preprint arXiv:2410.03298*.

Simul-S2ST Models	ASR-BLEU	ATD	StartOffset	EndOffset
SimulS2S-LLM (k=5)	23.2	2533	3109	1722
SimulS2S-LLM (k=6)	24.8	2862	3484	1788
SimulS2S-LLM (k=7)	25.3	3140	3828	2055
SimulS2S-LLM (k=8)	26.3	3440	4191	2209
Boundary-unaware SimulS2S-LLM (k=10)	17.7	2649	3810	2885
Boundary-unaware SimulS2S-LLM (k=11)	19.7	2989	3811	2849
Boundary-unaware SimulS2S-LLM (k=13)	21.4	3442	4374	2904
Boundary-unaware SimulS2S-LLM (k=15)	23.0	3847	4879	3014

Table 4: Numerical results of SimulS2S-LLM on CVSS-C Es-En corresponding to Fig. 4.

Simul-S2ST Models	ASR-BLEU	ATD	StartOffset	EndOffset
SimulS2S-LLM (k=5)	22.5	2103	2659	1659
SimulS2S-LLM (k=6)	24.2	2296	2914	1763
SimulS2S-LLM (k=7)	25.9	2460	3136	1857
SimulS2S-LLM (k=8)	26.9	2677	3378	1914
Boundary-unaware SimulS2S-LLM (k=9)	18.8	2307	3025	2309
Boundary-unaware SimulS2S-LLM (k=11)	21.4	2597	3447	2363
Boundary-unaware SimulS2S-LLM (k=13)	23.1	2829	3770	2421
Boundary-unaware SimulS2S-LLM (k=15)	23.7	2987	4014	2518

Table 5: Numerical results of SimulS2S-LLM on CVSS-C Fr-En corresponding to Fig. 4.

	CVSS-C Es-En	
Train set	train	
-Duration	69.5 hours	
-Sentences	79K	
Test sets	test	dev
-Duration	12.4 hours	12.4 hours
-Sentences	13K	13K
	CVSS-C Fr-En	
Train set	train	
-Duration	174.0 hours	
-Sentences	207K	
Test sets	test	dev
-Duration	13.3 hours	13.0 hours
-Sentences	15K	15K
	CVSS-C De-En	
Train set	train	
-Duration	112.4 hours	
-Sentences	128K	
Test sets	test	dev
-Duration	12.1 hours	12.5 hours
-Sentences	14K	14K

Table 6: Statistics of datasets used in this paper

A Data Statistics

The training and test data statistics are summarised in Table 6. Data pre-processing followed ESPnet-ST recipes, including speed perturbation with factors of 0.9 and 1.1 during the first-stage training. Model training was conducted on two NVIDIA A100 GPUs, each with 80GB of memory. For CVSS-C Es-En, each epoch of first-stage training required approximately 3 hours, while second-stage training took about 20 minutes per epoch. For CVSS-C Fr-En, the first-stage training required around 9 hours per epoch, with the second stage taking approximately 1 hour per epoch. For CVSS-C De-En, the first-stage training required around 4 hours per epoch, with the second stage taking approximately 30 minutes per epoch.

B Hyper-parameters

The hyper-parameters of the models we built are as follows, with other hyper-parameters following standard ESPnet-ST recipes. γ in Eq. 2 was set to 0.05. The beam size for LLM-based inference was set to 5, while the speech token-based incremental beam search used a beam size of 10. For SimulS2S-LLM on the Simul-S2ST task, the wait-k policy was configured with $k \in \{5, 6, 7, 8\}$ on Es-En and Fr-En. For Simul-S2ST De-En, the wait-k policy was configured with $k \in \{11, 13, 15, 17\}$, because Llama3-8B is English-centric, making German text token sequences longer than English. Hence, the learned CIF-based speech prompts become rela-

Simul-S2ST Models	ASR-BLEU	ATD	StartOffset	EndOffset
SimulS2S-LLM (k=11)	18.4	2819	3855	2182
SimulS2S-LLM (k=13)	19.6	3161	4258	2435
SimulS2S-LLM (k=15)	20.7	3469	4621	2636
SimulS2S-LLM (k=17)	21.6	3684	4931	2838
Boundary-unaware SimulS2S-LLM (k=9)	12.4	2467	3176	2563
Boundary-unaware SimulS2S-LLM (k=11)	14.5	2889	3757	2660
Boundary-unaware SimulS2S-LLM (k=13)	15.7	3255	4270	2896
Boundary-unaware SimulS2S-LLM (k=15)	16.8	3563	4697	3043

Table 7: Numerical results of SimulS2S-LLM on CVSS-C De-En corresponding to Fig. 4.

Simul-S2ST Models	ASR-BLEU	ATD_CA	StartOffset_CA	EndOffset_CA
SimulS2S-LLM (k=5)	23.2	3114	3627	1722
SimulS2S-LLM (k=6)	24.8	3462	4055	1788
SimulS2S-LLM (k=7)	25.3	3816	4467	2055
SimulS2S-LLM (k=8)	26.3	4239	4945	2209
Boundary-unaware SimulS2S-LLM (k=10)	17.7	3416	4486	2885
Boundary-unaware SimulS2S-LLM (k=11)	19.7	3694	4292	2849
Boundary-unaware SimulS2S-LLM (k=13)	21.4	4195	4939	2904
Boundary-unaware SimulS2S-LLM (k=15)	23.0	4709	5573	3014

Table 8: Compution-aware results of SimulS2S-LLM on CVSS-C Es-En corresponding to Table 4.

Simul-S2ST Models	ASR-BLEU	ATD_CA	StartOffset_CA	EndOffset_CA
SimulS2S-LLM (k=5)	22.5	2751	3201	1659
SimulS2S-LLM (k=6)	24.2	2965	3502	1763
SimulS2S-LLM (k=7)	25.9	3178	3784	1857
SimulS2S-LLM (k=8)	26.9	3438	4077	1914
Boundary-unaware SimulS2S-LLM (k=9)	18.8	2993	3525	2309
Boundary-unaware SimulS2S-LLM (k=11)	21.4	3347	4031	2363
Boundary-unaware SimulS2S-LLM (k=13)	23.1	3652	4465	2421
Boundary-unaware SimulS2S-LLM (k=15)	23.7	3893	4804	2518

Table 9: Compution-aware results of SimulS2S-LLM on CVSS-C Fr-En corresponding to Table 5.

tively longer and require larger k values. For SimulS2TT, k was set to $k \in \{3, 4, 5, 7\}$ on Es-En. L_{max} for SimulS2S-LLM simultaneous inference is set to 0.15 times the length of the encoder output

C Numerical Values for Figure 4

The numerical values for Fig. 4 are provided in Tables 4, 5, and 7. In addition to the ATD values displayed in Fig. 4, the table includes the StartOffset and EndOffset metrics. StartOffset represents the delay before generating the first frame of the target speech, while EndOffset indicates the offset of the final frame of the target speech relative to the completion of the source speech. No matter which latency metric is used, the conclusion remains consistent.

D Compution-aware Latency Results

This section gives the latency results after considering the computation time, as shown in Table 8, Table 9, and Table 10. Note that, as mentioned in the Limitations section, the use of LLM will increase the computational load, which is a common challenge faced by the entire community. The results were tested using an A100 GPU, which will likely be lower if a more powerful GPU, such as the H100, is used.

Considering the actual computation time certainly increases latency. However, even with LLM used, the computation-aware latency and translation quality trade-off remain promising. This computation-aware latency result will evolve with hardware advancements, and reducing LLM com-

Simul-S2ST Models	ASR-BLEU	ATD_CA	StartOffset_CA	EndOffset_CA
SimulS2S-LLM (k=11)	18.4	3637	4589	2182
SimulS2S-LLM (k=13)	19.6	4073	5095	2435
SimulS2S-LLM (k=15)	20.7	4500	5575	2636
SimulS2S-LLM (k=17)	21.6	4832	5987	2838
Boundary-unaware SimulS2S-LLM (k=9)	12.4	3127	3635	2563
Boundary-unaware SimulS2S-LLM (k=11)	14.5	3528	4249	2660
Boundary-unaware SimulS2S-LLM (k=13)	15.7	4007	4879	2896
Boundary-unaware SimulS2S-LLM (k=15)	16.8	4432	5424	3043

Table 10: Compution-aware results of SimulS2S-LLM on CVSS-C De-En corresponding to Table 5.

putational load has been actively studied by the community.

In addition, comparing the latency results with and without considering the computation time, the smaller k does not make the gap larger than the larger k value. Smaller k-values will require more frequent past key-value updates before LLM generation after new speech input, as mentioned in Section 3.4, so it can be seen that this update does not greatly increase the computation time as it is calculated in parallel.

E Comparison with Foundation Models

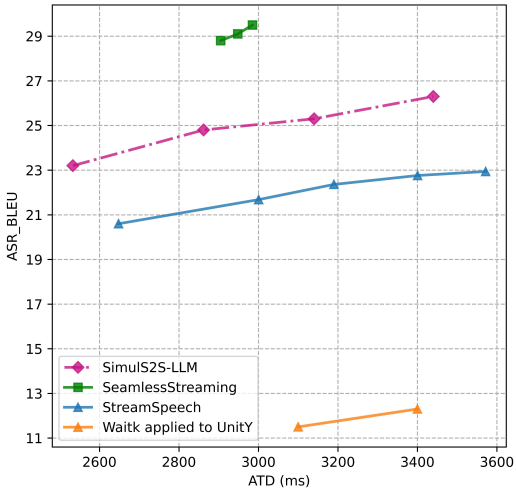


Figure 7: Simul-S2ST results of different models on CVSS-C Es-En. Note the comparison is not well-controlled as the SeamlessStreaming (Barrault et al., 2023) has been extensively trained as a translation foundation model using about 9,300 hours of speech-to-speech training data. Other models were trained on the same dataset, i.e., CVSS-C, but only SimulS2S-LLM can utilise the LLM.

This section further compares the proposed SimulS2S-LLM with the foundation model SeamlessStreaming (Barrault et al., 2023), although they are not directly comparable since SeamlessStreaming has been extensively trained on approximately 9,300 hours of speech-to-speech data. Additionally, the results of applying wait-k to the UnitY (Inaguma et al., 2023b) model, reproduced by Zhang et al. (2024), are also included for comparison.

As shown in Fig. 7 and Fig. 8, StreamSpeech greatly outperforms the UnitY model that uses the Wait-k strategy on both the Simul-S2ST and Simul-S2TT tasks, which is consistent with the findings of Zhang et al. (2024), showing that StreamSpeech is the existing state-of-the-art solution.

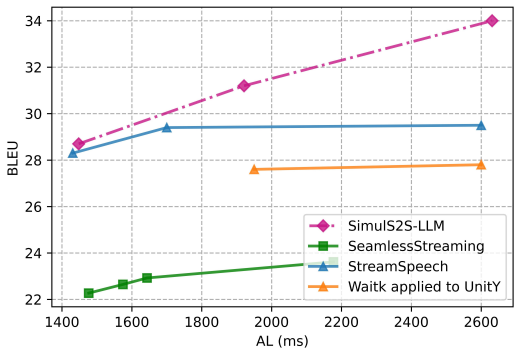


Figure 8: Simul-S2ST results of different models on CVSS-C Es-En. Note the comparison is not well-controlled as Fig. 7.

On the Simul-S2ST task, as shown in Fig. 7, SeamlessStreaming unsurprisingly achieves the best results, but performs poorly on Simul-S2TT as shown in Fig. 8, consistent with the findings of (Barrault et al., 2023). Therefore, in general, the proposed SimulS2S-LLM achieves promising results, and the gap with SeamlessStreaming on Simul-S2ST is also acceptable considering that SeamlessStreaming uses a much larger training data size. If there is more speech-to-speech training data, the performance of SimulS2S-LLM can be expected to be further greatly improved.

Simul-S2TT Models	BLEU	LAAL(ms)	AL(ms)
SimulS2S-LLM (k=3)	24.7	1384	1061
SimulS2S-LLM (k=4)	28.7	1764	1448
SimulS2S-LLM (k=5)	31.2	2170	1921
SimulS2S-LLM (k=7)	34.0	2805	2631
Boundary-unaware SimulS2S-LLM (k=5)	20.9	1343	1042
Boundary-unaware SimulS2S-LLM (k=7)	26.4	1761	1440
Boundary-unaware SimulS2S-LLM (k=9)	27.7	2317	2022
Boundary-unaware SimulS2S-LLM (k=11)	31.4	2902	2704

Table 11: Numerical Simul-S2TT results of SimulS2S-LLM on CVSS-C Es-En corresponding to Fig. 5.

F Numerical Values for Figure 5

The numerical values for Fig. 5 are provided in Table 11. In addition to the AL values displayed in Fig. 5, this table includes the alternative Length-Adaptive Average Lagging (LAAL) (Papi et al., 2022) latency metric results. The conclusion remains consistent across different metrics.