InfiniSST: Simultaneous Translation of Unbounded Speech with Large Language Model

Anonymous ACL submission

Abstract

Simultaneous translation of unbounded streaming speech remains a challenging problem due to the need for effectively processing the historical speech context and past translations so that quality and latency, including computation overhead, can be balanced. Most prior works assume pre-segmented speech, limiting their real-world applicability. In this paper, we propose InfiniSST, a novel approach that formulates SST as a multi-turn dialogue task, enabling seamless translation of unbounded speech. We construct translation trajectories and robust segments from MuST-C with multi-latency augmentation during training and develop a key-value (KV) cache management strategy to facilitate efficient inference. Experiments on MuST-C En-Es, En-De, and En-Zh demonstrate that InfiniSST reduces computation-aware latency by 0.5 to 1 second while maintaining the same translation quality compared to baselines. Ablation studies further validate the contributions of our data construction and cache management strategy.

1 Introduction

Simultaneous speech translation (SST) is the task of translating partial speech input from a source language into text in a target language, with a wide range of applications, including conference interpretation and live-streaming translation (Ma et al., 2020b; Ren et al., 2020). Most prior research on SST focuses on translating pre-segmented speech (SST-S), assuming that ground-truth segmentation is provided (Liu et al., 2021; Zeng et al., 2021; Dong et al., 2022; Papi et al., 2023, 2024b). However, translating unbounded, streaming speech (SST-U) remains underexplored.

Unbounded speech presents a major challenge that the model has to effectively process the historical speech context and past translations so that quality and latency, including computation overhead, can be balanced. Large language model (LLM) is a promising solution for long-context modeling with the recent advancements (Su et al., 2021; Xiao et al., 2024; Han et al., 2024). Moreover, LLM-based architectures have been shown to improve SST-S performance (Xu et al., 2024). However, conventional SST-S approaches suffer from high computational costs, as they require recomputing features for past speech and generated text every time a new speech chunk arrives. Some studies mitigate this issue by framing SST as a multi-turn dialogue task, either explicitly (Yu et al., 2025; Wang et al., 2024) or implicitly (Ouyang et al., 2024; Raffel et al., 2024), leveraging key-value (KV) caching to improve efficiency. While effective for segmented speech and text, these methods do not seamlessly extend to unbounded speech.

In this paper, we propose InfiniSST, a method for simultaneous translation of unbounded speech using a multi-turn dialogue format. We construct SST trajectories and derive robust speech segments for training from the MuST-C dataset, enhancing them with a multi-latency strategy to increase diversity. During inference, we employ a KV cache management strategy, inspired by Han et al. (2024), to enable seamless extrapolation to unbounded speech input. Experiments on MuST-C En-Es, En-De, and En-Zh (Di Gangi et al., 2019) show that InfiniSST reduces computation-aware latency by 0.5 to 1 second while maintaining the same BLEU score as baselines. A detailed ablation study further validates the effectiveness of our data construction and cache management strategies during inference.

2 Related Works

2.1 SST on Unbounded Speech

Cascade Approaches Cascade-based methods typically use an automatic speech recognition (ASR) model to segment and transcribe the input, followed by a machine translation model that translates the transcription (Fugen et al., 2006; Yoshimura et al., 2020; Huang et al., 2022; Donato et al., 2021; Iranzo-Sánchez et al., 2024). However, segmentation errors and the lack of punctuation degrade translation quality, which complicates maintaining low latency and high quality.

Direct SST on Unbounded Speech Several works explore end-to-end approaches for SST on unbounded speech (Schneider and Waibel, 2020; Papi et al., 2024a). These methods avoid external segmentation by dynamically preserving relevant audio context and previously generated text while discarding older information. Papi et al. (2024a) extends AlignAtt to unbounded speech by storing text and audio history in a fully streaming way, which helps reduce latency and maintain contextual awareness. Despite these advances, balancing translation quality, latency, and computational demands remains a challenge. Our approach addresses these issues by managing unbounded speech input without loss in translation accuracy and with improved computational efficiency.

2.2 Length Extrapolation of LLM

Recent advances in positional encoding (Su et al., 2021; Press et al., 2021; Sun et al., 2023) have enabled models to handle longer sequences with little or no additional training. ReRoPE (Su, 2023) introduces an NTK-aware Scaled RoPE that extends context length to infinite without fine-tuning. Han et al. (2024) and Xiao et al. (2024) propose on-the-fly length generalization based on a $\mathbf{\Lambda}$-shaped attention window, allowing nearly unlimited input length with no fine-tuning. InfiniSST is a successful application of RoPE and $\mathbf{\Lambda}$-shaped attention window in SST-U.

3 Method

3.1 Problem Formulation

Let $\boldsymbol{s}_{1:t} = (s_1, s_2, \ldots, s_t)$ be the partial input of an unbounded input speech sequence and $\boldsymbol{y}_{1:i} = (y_1, y_2, \ldots, y_i)$ represent the partial text translation. Here $\boldsymbol{s}_{1:t}$ is the waveform with a specific sampling rate. Define $\pi(\boldsymbol{s}_{1:t}, \boldsymbol{y}_{1:i}) \in [0, 1]$ as the policy to determine whether to take more speech input (=0) or to generate target translation tokens (=1). Whenever $\pi(\boldsymbol{s}_{1:t}, \boldsymbol{y}_{1:i}) = 1$, we define $g_{i+1} = t$ as the delay of $i+1$-th token. Let $g_0 = 0$. In addition, let θ be the model parameter, we define the probability of generating next token given a partial speech input as $P_\theta(y_{i+1} \mid \boldsymbol{s}_{1:t}, \boldsymbol{y}_{1:i})$. In our formulation, we use a simple policy by checking whether the current generated token y_i is a special ending token T_0 (e.g. stop writing translation and read speech input when encountering "⟨EOT⟩" token in Llama (Grattafiori et al., 2024)).

$$\pi(\boldsymbol{s}_{1:t}, \boldsymbol{y}_{1:i}) = \begin{cases} 0, & \text{if } y_i = T_0 \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

Given $\boldsymbol{s}$, we define the conditional probability of generating a translation sequence $\boldsymbol{y}_{1:i}$ with associated delays for each token $\boldsymbol{g}_{1:i}$ as:

$$P(\boldsymbol{y}, \boldsymbol{g}|\boldsymbol{s}) = \prod_{i=1}^{|\boldsymbol{y}|} \Bigg(P_\theta(y_i | \boldsymbol{s}_{1:g_i}, \boldsymbol{y}_{1:i-1}) \tag{2}$$
$$\pi(\boldsymbol{s}_{1:g_i}, \boldsymbol{y}_{1:i-1}) \prod_{j=g_{i-1}}^{g_i - 1} \left(1 - \pi(\boldsymbol{s}_{1:j}, \boldsymbol{y}_{i-1})\right) \Bigg)$$

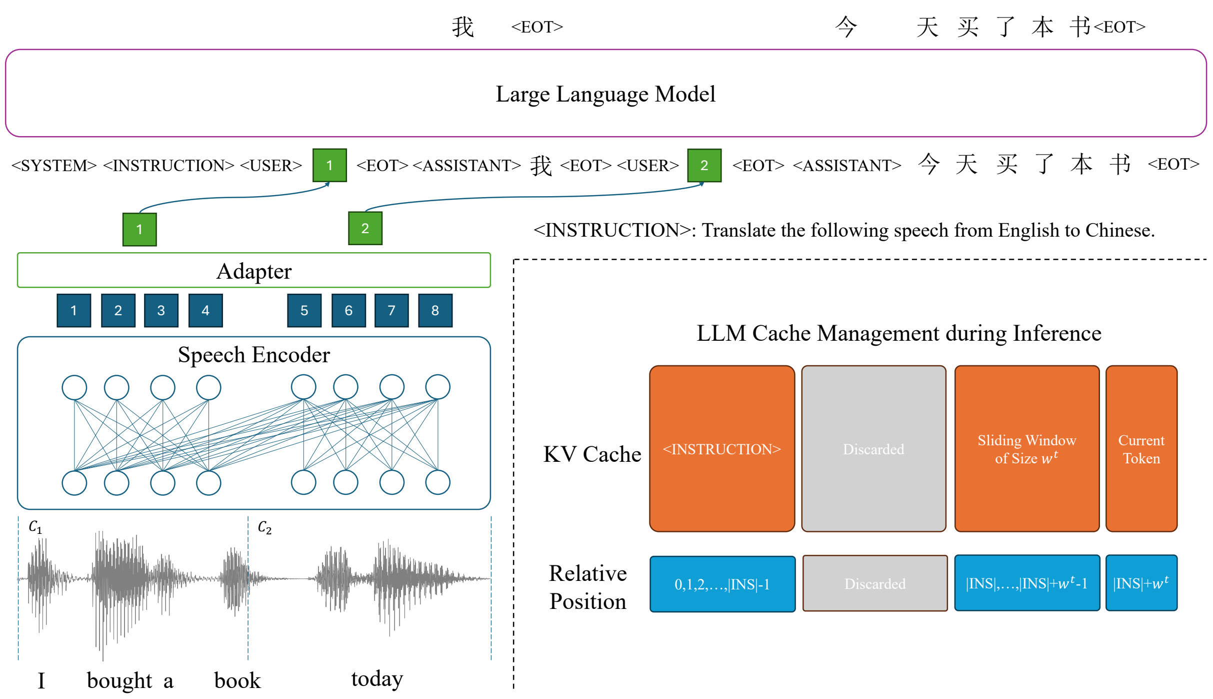
The translation quality and latency are subsequently evaluated based on $\boldsymbol{s}$, $\boldsymbol{y}$ and $\boldsymbol{g}$.

3.2 Model Architecture

We design InfiniSST, a simultaneous speech translation model that can take unbounded streaming speech input and generate target text efficiently. The InfiniSST consists of 1) a streaming speech encoder to incrementally compute representations of partial speech input without recomputation, 2) a speech-to-token embedding adapter to match speech representations to LLM's token embedding space, and 3) an multi-turn LLM decoder to interactively take speech input and generate translation as needed (Figure 1).

Streaming Speech Encoder We modify a pretrained wav2vec2 (Baevski et al., 2020)[1] speech encoder to encode the unbounded streaming speech input. However, there is a major limitation of the original wav2vec2. It uses bidirectional attention and bidirectional convolutional position embedding, which needs to recompute the representations for every new segment of streaming speech input. To handle unbounded speech input, we introduce three modifications to the speech encoder. Firstly, we replace the wav2vec2's convolutional positional embedding with a rotary positional embedding (RoPE) (Su et al., 2024) because it shows better extensibility for long sequences. Secondly, we replace bidirectional attention with chunk-wise

[1] Wav2Vec 2.0 Large (LV-60) + Self Training on LibriSpeech. https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt



我 <EOT> 今 天 买 了 本 书<EOT>
Large Language Model
<SYSTEM> <INSTRUCTION> <USER> 1 <EOT> <ASSISTANT> 我 <EOT> <USER> 2 <EOT> <ASSISTANT> 今 天 买 了 本 书 <EOT>
<INSTRUCTION>: Translate the following speech from English to Chinese.
Adapter
Speech Encoder
LLM Cache Management during Inference
KV Cache
<INSTRUCTION>
Discarded
Sliding Window of Size w^t
Current Token
Relative Position
0,1,2,…,|INS|-1
Discarded
|INS|,…,|INS|+w^t-1
|INS|+w^t
c_1
c_2
I bought a book today

Figure 1: Model architecture of InfiniSST. InfiniSST first encodes speech using a chunkwise-causal speech encoder, then compresses the speech features into embeddings via an adapter. The large language model (LLM) processes the input by first reading a system instruction, then alternating between consuming speech embeddings and generating translations. The translation process stops when the LLM generates an EOT token. During inference, we employ a sliding window of size w^t for the LLM, conditioning the translation on the most recent w^t KV caches along with the KV cache of the system instruction, enabling extrapolation to unbounded speech input.

causal attention (Deng et al., 2022). Each chunk contains 48 frames in wav2vec2, with a total duration of 960ms. The multihead attention within each chunk remains bidirectional while attention across chunks is causal. This is achieved by adding blockwise masking to the attention weights. Thirdly, we apply a sliding window mechanism with window size w^s to maintain a finite context length, restricting chunk i to attend only to hidden states of chunks $[i - w^s + 1, i]$. In practice, we use $w^s = 10$ so each speech embedding is computed from roughly 9.6 seconds of the preceding speech input.

Speech-to-Token Embedding Adapter The speech encoder yields a sequence slightly longer than the transcript and with embeddings that differ from the LLM's token embeddings. Following Zhang et al. (2023), we downsample the encoder output with two 1-D convolutions (kernel size 2, stride 2, no padding) and linearly project the result into the LLM embedding space, so 48 input frames produce 12 embedding vectors.

Multi-turn LLM Decoder Our decoder needs to produce target text and a special token to indicate the switching from generation to taking speech input. To this end, we use Llama-3.1-8B-Instruct (Grattafiori et al., 2024)[2] and employ a multi-turn dialogue format to formulate the input. We first feed a system instruction

```
Translate the following speech
    from <LangX> to <LangY>.
```

We then add a special `USER` token to indicate that the following 12 embeddings and a trailing `EOT` (`End-Of-Turn`) token are for speech input. We then prompt the LLM with a special `ASSISTANT` token to force LLM to generate tokens. We add a policy module to check generated tokens. When the policy module encounters the special `EOT` token, it will feed a special `USER` token and take 12 new streaming speech embeddings with a trailing `EOT` token as new input to the LLM. We will describe later our inference method to incrementally compute embeddings and generate target tokens for infinite speech input.

[2]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

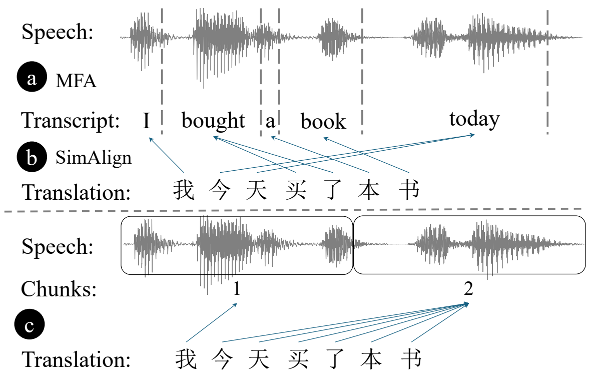

Speech:
a MFA
Transcript: I bought a book today
b SimAlign
Translation: 我 今 天 买 了 本 书
Speech:
Chunks: 1 2
c
Translation: 我 今 天 买 了 本 书

Figure 2: Trajectory construction process. We first align speech to its transcript using forced alignment, then align the transcript to its translation using SimAlign. The resulting alignments are monotonized and grouped by speech chunks.

3.3 Training Data Construction

SST Trajectory Common speech translation datasets like MuST-C are segmented from complete talks (Di Gangi et al., 2019). To train an SST model in a multi-turn dialogue format, we transform segmented ST triplets (speech $\boldsymbol{s}$, transcript $\boldsymbol{x}$, translation $\boldsymbol{y}$) from MuST-C dataset into SST trajectories. An SST trajectory represents an alternating action sequence of speech reading and translation writing.

As shown in Figure 2, we first align speech utterances with their corresponding transcripts using the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017)[3]. Let m_k^{sx} denote the right boundary of the speech segment corresponding to the transcript token x_k. Also, we utilize SimAlign (Jalili Sabet et al., 2020) with the LaBSE model (Feng et al., 2022) to align words between the transcript and translation. We then monotonize these alignments following Wang et al. (2024). Let $x_{m_i^{xy}}$ be the transcript token that corresponds to translation token y_i. By combining m^{sx} and m^{xy}, we establish a mapping from translation token y_i to its speech boundary $m_i^{sy} = m^{sx}_{m_i^{xy}}$, meaning that y_i is generated after reading $\boldsymbol{s}_{1:m_i^{sy}}$.

Finally, we cut the speech utterance into fixed-length chunks, each lasting 960 ms. We then concatenate translation tokens whose corresponding speech boundaries fall within the same chunk, forming a sequence of trajectory $(\boldsymbol{s}_{C_1}, \boldsymbol{y}_{C_1}), (\boldsymbol{s}_{C_2}, \boldsymbol{y}_{C_2}), \ldots$.

[3] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

Robust Segments for Training Segmented speech utterances primarily consist of human speech; however, non-linguistic sounds (e.g., laughter, applause) are also present. To enhance the robustness of the SST dataset, we cut the entire talk evenly into robust segments that each span 30 speech chunks. If a robust segment starts in the middle of a segmented speech utterance, we shift the robust segment to start with this utterance. The trajectories for a robust segment can then be built by concatenating the trajectories of segmented utterances within this robust segment according to their timestamps and filling the rest translation entries of the trajectory as empty strings.

Multi-Latency Augmentation To further enhance trajectory diversity during training, we propose a simple yet effective multi-latency augmentation strategy. Specifically, given a trajectory $(\boldsymbol{s}_{C_1}, \boldsymbol{y}_{C_1}), (\boldsymbol{s}_{C_2}, \boldsymbol{y}_{C_2}), \ldots$, we randomly select a latency multiplier $m \in [1, M]$ and merge every m consecutive chunks of speech with their corresponding translations. The i-th step in the augmented trajectory is then represented as

$$(\boldsymbol{s}_{C_{im},\ldots,C_{(i+1)m-1}}, \boldsymbol{y}_{C_{im},\ldots,C_{(i+1)m-1}}).$$

We also multiply the chunk size of speech encoder with m, i.e., number of frame in a chunk becomes $48m$.

3.4 Training

We train InfiniSST with standard cross-entropy loss on translation tokens, including `EOT`, of the augmented trajectory from robust segments. In the first stage, we freeze the LLM and train only the speech encoder and adapter. In the second stage, we freeze the speech encoder and adapter, training only the LLM.

3.5 Inference on Unbounded Speech

During inference, we cut the unbounded input speech into 960 ms chunks. The latency multiplier m during inference regulates latency by ensuring that translation begins only after every m new chunks have arrived.

At the i-th step, suppose the newly received speech chunks are $C_{im}, \ldots, C_{(i+1)m-1}$. Both the speech encoder and the LLM maintain a key-value (KV) cache to prevent redundant computations. Notably, the stored key and value features are extracted *before* applying RoPE, ensuring that no

Language	LAAL
En-Zh	1171 ms
En-De	924 ms
En-Es	850 ms

Table 1: The latency of constructed trajectories for MuST-C En-Zh/De/Es sentence-level utterances evaluated with LAAL.

positional information is embedded within the KV cache.

The speech encoder processes the m new chunks into $48m$ speech features, utilizing the KV cache from chunks $C_{im-w^s+1}, \ldots, C_{im-1}$, where w^s is the sliding window size defined in Section 3.2. The adapter then downsamples the $48m$ features into $12m$ embeddings, which are passed to the LLM.

As shown in Figure 1, the LLM employs a sliding window of size w^t. By default, $w^t = 1000$. Inspired by Han et al. (2024) and Xiao et al. (2024), we concatenate the KV cache of instruction with those of the most recent w^t tokens and apply RoPE on top of them. Then the LLM generate translations conditioned on this combined KV cache.

4 Experiment Setups

4.1 Data

We conduct experiments on the En-Es (v1), En-De (v1), and En-Zh (v2) directions of the MuST-C dataset (Di Gangi et al., 2019). Due to the poor alignment quality in the En-Zh training set, we filter out misaligned ST triplets using CometKiWi (Rei et al., 2022) and retranslate them using TowerInstruct (Alves et al., 2024). Then, we construct trajectories and robust segments as described in Section 3.3. Examples of trajectory are shown in Figure 10. The latency of trajectories are shown in Table 1. Further statistics can be found in the Appendix A.

4.2 Training

For all three language directions, we set maximum latency multiplier $M = 12$. We adopt a two-stage supervised fine-tuning approach. In the first stage, we freeze the LLM and train only the speech encoder and adapter for 6 epochs with an effective batch size of 57.6K tokens. We use Adam optimizer (Kingma and Ba, 2017) with learning rate 2×10^{-4} and 1000 warmup steps. In the second stage, we fine-tune the entire LLM for 1 epoch with an effective batch size of 76.8K tokens, a learning rate of 7×10^{-6} and 100 warmup steps. We apply gradient clipping with a norm of 1.0. We employ DeepSpeed Zero Stage-2 optimization[4], and enable optimizer and parameter offloading during the second training stage. All models referred to in this paper are trained on a single node of 8 L40S GPU.

4.3 Inference

We use beam search with beam width 4 for all methods. We set `no_repeat_ngram_size=5` and `repetition_penalty=1.2` to suppress repetition[5]. The sliding window size of LLM is set to 1000 tokens and that of speech encoder is set to 10 speech chunks. We vary the latency multiplier m from 1 to 5 for InfiniSST to obtain a quality-latency trade-off.

4.4 Evaluation

We evaluate SST on complete TED Talks from the MuST-C tst-COMMON set, which consists of 27 TED Talks with durations ranging from 3 to 23 minutes. To assess translation quality, we use SacreBLEU (Post, 2018) and COMET (Guerreiro et al., 2024). Following the WMT24 practice (Freitag et al., 2024), we compute the COMET score by averaging the scores from XCOMET-XL and XCOMET-XXL. For latency evaluation, we use Length-Adaptive Average Lagging (LAAL) (Papi et al., 2022) for segmented speech baselines and StreamLAAL (Papi et al., 2024a) for unbounded speech, both implemented within the SimulEval framework (Ma et al., 2020a). Computation cost is measured using both computation-aware StreamLAAL (StreamLAAL_CA) and the Real-Time Factor (RTF), defined as the ratio of wall-clock computation time to speech duration.

4.5 Baselines

We compare our method against the following baselines:

AlignAtt (Papi et al., 2023) is a state-of-the-art SST policy applied to offline ST models, translating based on attention scores between translation outputs and speech utterances. It is designed for SST on segmented speech, and we include its results as a reference. We train an offline ST model using segmented ST triplets from MuST-C and robust segments that we constructed. The offline

[4] https://github.com/deepspeedai/DeepSpeed

[5] https://huggingface.co/docs/transformers/en/main_classes/text_generation#transformers.GenerationConfig

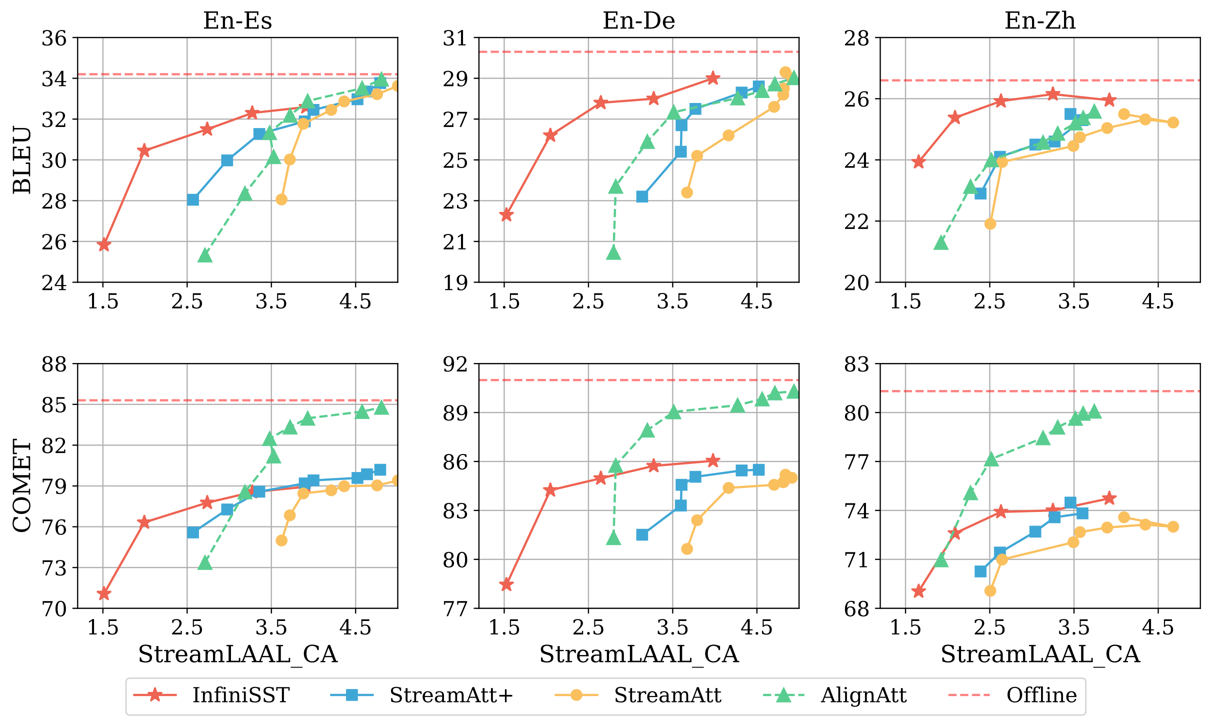


Figure 3: Quality-latency trade-off of InfiniSST compared to the baselines on complete TED talks from the MuST-C tst-COMMON dataset in the En-Es, En-De, and En-Zh directions. Translation quality is measured using BLEU and COMET scores, while latency is evaluated using the *computation-aware* StreamLAAL metric. For reference, we also include offline translation quality and results from AlignAtt tested on segmented speech. InfiniSST achieves significantly lower computation-aware latency compared to StreamAtt at the same quality.

ST model uses the LST architecture (Zhang et al., 2023), where the LLM inputs are organized as (instruction, speech history, translation history) rather than interleaving speech and translation as in InfiniSST. We use the attention scores from layer 14 of the LLM and vary the number of frames from 1 to 8.

StreamAtt (Papi et al., 2024a) extends AlignAtt to unbounded speech by maintaining both text and audio history through attention-based selection. We adopt the Fixed-Word approach from StreamAtt, preserving 40 words in the text history. To prevent excessively long preserved speech, we apply truncation when the duration exceeds 28.8 seconds.

StreamAtt+ We observe that the vanilla truncation strategy sometimes removes too much audio, leading to critical misalignment between the preserved speech and its translation. To mitigate this issue, we modify StreamAtt by ensuring that audio segments shorter than 10 seconds are never truncated.

5 Main Results

Lower Computation Cost We run all inference experiments on a single NVIDIA L40S GPU and an AMD EPYC 9354 32-Core CPU. Results evaluated with StreamLAAL_CA are shown in Figure 3. InfiniSST achieves 0.5 to 1 second lower computation aware latency compared to StreamAtt and StreamAtt+ at the same quality level. We also compare the Real-Time Factor (RTF) of InfiniSST and StreamAtt+ in Figure 8. The RTF of InfiniSST is significantly lower than StreamAtt+, indicating that the computation overhead of InfiniSST is less than half of the StreamAtt+.

Competitive Translation Quality at the Same Theoretical Latency Results evaluated with non-computation-aware StreamLAAL are shown in Figure 4. When StreamLAAL is no more than 1.5 second, InfiniSST achieves slightly higher BLEU scores ($0.5 \sim 1.0$) and similar COMET scores than StreamAtt+ on all three language directions. When StreamLAAL is more than 1.5 second, InfiniSST still achieves higher BLEU score on En-Zh direction and competitive with StreamAtt+ on the En-De and En-Es directions. We note that Alig-

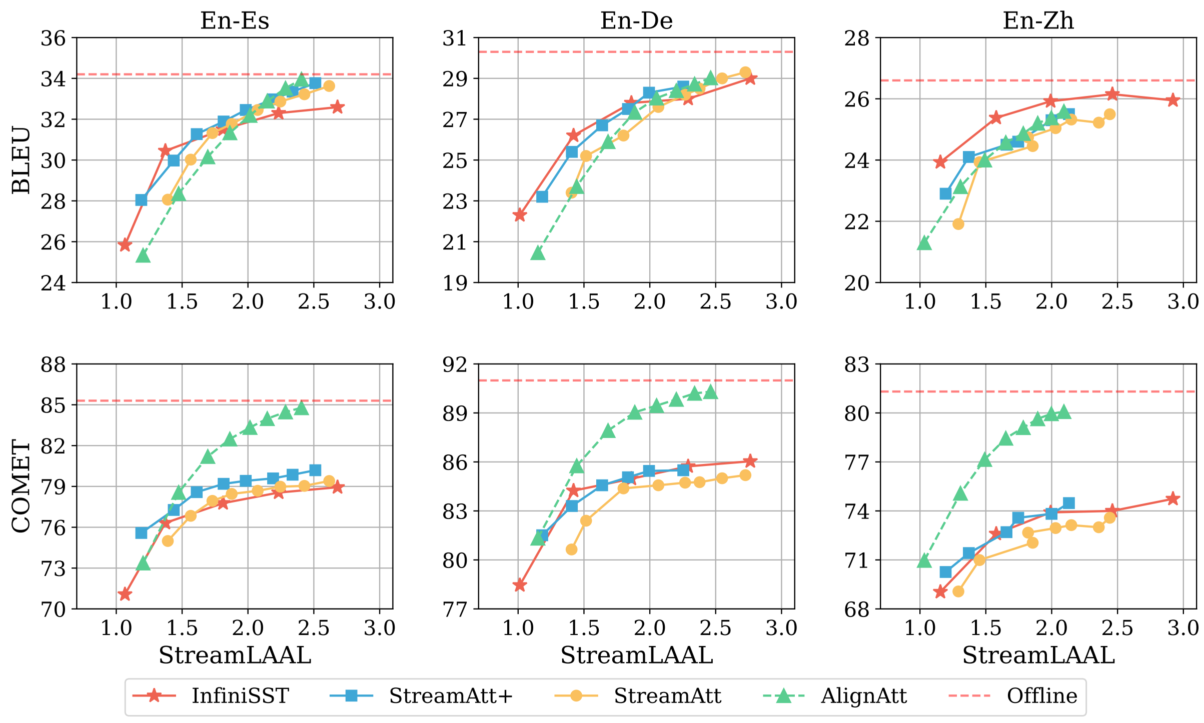


Figure 4: Quality-latency trade-off of InfiniSST compared to the baselines on complete TED talks from the MuST-C tst-COMMON dataset in the En-Es, En-De, and En-Zh directions. Translation quality is measured using BLEU and COMET scores, while latency is evaluated using the *non-computation-aware* StreamLAAL metric. For reference, we also include offline translation quality and results from AlignAtt tested on segmented speech. InfiniSST achieves slightly better translation quality than StreamAtt at latency $\leq$ 1.5 seconds and remains competitive at higher latency levels.

nAtt tested on segmented speech exhibit significant higher COMET scores but not BLEU scores than both InfiniSST and StreamAtt on all three language directions. A possible reason is that StreamLAAL uses mWERSegmenter (Matusov et al., 2005) to find alignment between translation of the complete talk and segmented references, and COMET is more sensitive to such misalignment than BLEU.

6 Ablation Studies

The default model we use in the ablation study is trained with robust segments and a maximum latency multiplier of $M = 4$ on the En-Zh direction.

6.1 Data

Robust Segments We evaluate the effectiveness of robust segments by comparing InfiniSST trained on trajectories of robust segments with InfiniSST trained on trajectories of original MuST-C segmented speech. Both models are evaluated on tst-COMMON En-Zh with latency multipliers $m \in [1, 4]$, and the results are presented in Table 2.

The model trained on trajectories of non-robust segments exhibits abnormal latency scores and lower translation quality compared to the model trained on trajectories of robust segments. Manual examination of translation instances reveals that the segmented speech model frequently falls into repetition of non-linguistic tokens such as （笑声） (meaning laughter) whenever non-linguistic sounds appear in the audio.

Robust Segments	Non-Robust Segments	Non-Robust Segments[*]
69.2 / 1.1	50.5 / -220	51.0 / -207
71.9 / 1.5	53.4 / -116	58.1 / -58
72.3 / 1.9	68.4 / 2	65.7 / -22
73.0 / 2.4	66.8 / -12	67.2 / -6

Table 2: Impact of robust segments evaluated on MuST-C En-Zh tst-COMMON with latency multipliers $m = 1, 2, 3, 4$. A / B stands for COMET / LAAL (in second). The model trained on non-robust segments fails to translate unbounded speech. [*]We suppress the non-linguistic sound tokens but still the model fails to generalize.

We attempted to suppress these tokens, and the results are reported in the last column of Table 2.

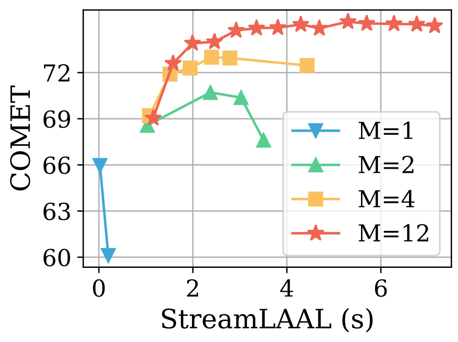


Figure 5: InfiniSST trained with maximum latency multipliers $M = 1, 2, 4, 12$ and evaluated with $m \leq M+2$. Larger maximum latency multipliers during training lead to improved quality-latency trade-offs.

Speech Cache Window w^s	LLM Cache Window w^t	Quality / Latency
10	1000	69.2 / 1.1
5	1000	68.7 / 1.1
20		68.3 / 1.0
40		66.1 / 0.9
10	500	69.0 / 1.0
	2000	69.4 / 1.2
	4000	69.4 / 1.2

Table 3: Impact of cache size during inference. Quality is evaluated with COMET and latency is evaluated with StreamLAAL (unit is second). Model is trained with speech encoder sliding window $w^s = 10$ and no sliding window for LLM. Latency multiplier is set to $m = 1$.

Instead of producing repetitive tokens, the model stops generating translations upon encountering non-linguistic sounds. These findings highlight the importance of training with robust segments.

Multi-Latency We evaluate the effectiveness of multi-latency augmentation by training models with maximum latency multipliers $M = 1, 2, 4,$ and 12, and performing inference with $m \leq M+2$. Results are shown in Figure 5.

Larger M consistently leads to better quality-latency trade-offs. Within the training range ($m \leq M$), translation quality improves with higher m; beyond it ($m > M$), quality degrades since it is out of the model's training distribution. However, models trained with larger M degrade less when extrapolated to $m > M$.

These results highlight the importance of training with a sufficiently large M while keeping $m \leq M$ at inference for optimal performance.

6.2 Speech Encoder

Inference Cache Window We first evaluate how the speech encoder's cache window during inference affects model performance. The model is trained with $w^s = 10$ and tested with $w^s = 5, 10, 20,$ and 40. The results, presented in Table 3, indicate that using a different cache window size during inference than the one used during training degrades translation quality.

Training Cache Window Furthermore, we train models with different cache window sizes $w^s = 10, 20, 30$ while ensuring that the cache window size matches between training and inference. Since each robust segment has a size of 30, training with $w^s = 30$ disables the sliding window mechanism.

The results, shown in Figure 7, reveal a surprising observation: the model trained with $w^s = 30$ successfully scales to unbounded speech during inference despite not using a sliding window during training. It also achieves a slightly better quality-latency trade-off compared to the model trained with $w^s = 10$. These findings suggest using the largest possible speech cache window that GPU memory allows.

6.3 LLM

Cache Instruction As described in Section 3.5, we explicitly preserve the KV cache of the translation instruction at the beginning (i.e., the system prompt). If this cache is not retained, the LLM stops translating once the window starts sliding.

Cache Window w^t We evaluate the impact of the LLM's cache window size during inference on model performance. Notably, the sliding window mechanism is not applied to the LLM during training. We vary the LLM cache window size as $w^t = 500, 1000, 2000, 4000$, and the results are presented in Table 3. Increasing the KV cache size slightly improves translation quality ($69 \rightarrow 69.4$) at the cost of marginally higher latency ($1.0 \rightarrow 1.2$). Compared to the speech encoder, the LLM demonstrates greater robustness to different KV cache window sizes.

Base LLM Context Length Throughout our experiments, we use Llama-3.1-8B-Instruct as the base LLM, which supports a context length of up to 128K tokens. To assess whether InfiniSST generalizes to an LLM with a shorter context limit, we replace it with Llama-3-8B-Instruct, which has

Model	Talks $\leq$ 10min	Talks $>$ 10min
Llama-3-8K	70.9 / 1.0	67.1 / 1.1
Llama-3.1-128K	71.6 / 1.0	68.0 / 1.1

Table 4: Impact of LLM context length. A / B stands for COMET / LAAL (in second). Llama-3 with 8K context length is still able to generalize to talks longer than 10 minutes.

an 8K context length[6]. The results, presented in Table 4, indicate that while Llama-3 exhibits lower translation quality compared to Llama-3.1, it is still capable of generalizing to unbounded speech with InfiniSST.

7 Conclusion

We propose InfiniSST that enables simultaneous translation of unbounded speech with state-of-the-art quality latency trade-off on three language directions of MuST-C dataset. Our ablations demonstrate the effectiveness of our carefully constructed data, including robust segments and multi-latency augmentation, and cache management strategy during inference.

Limitations

On the higher theoretical latency level, InfiniSST still falls behind AlignAtt and StreamAtt in some cases. This can be attributed to the limited bidirectional attention of the chunkwise-causal speech encoder. Also, we evaluated on En-X directions but not on other directions like X-En and X-X. We have not experimented with other pretrained speech encoders and non-Llama LLMs due to computation budget. Besides, the StreamLAAL metric is not perfectly reliable due to alignment errors of mWERSegmenter. Finally, we have not conducted human evaluation on user experience of different SST models, which might reveal undetected flaws in current models.

References

Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. In *First Conference on Language Modeling*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Keqi Deng, Shinji Watanabe, Jiatong Shi, and Siddhant Arora. 2022. Blockwise streaming transformer for spoken language understanding and simultaneous speech translation. In *Interspeech 2022*, pages 1746–1750.

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

Domenic Donato, Lei Yu, and Chris Dyer. 2021. Diverse pretrained context encodings improve document translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1299–1311, Online. Association for Computational Linguistics.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. Learning when to translate for streaming speech. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694, Dublin, Ireland. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.

C. Fugen, M. Kolss, D. Bernreuther, M. Paulik, S. Stuker, S. Vogel, and A. Waibel. 2006. Open domain speech recognition & translation:lectures and speeches. In *2006 IEEE International Conference on*

[6]A 10-minute speech already generates $10 \cdot 60 \cdot 12.5 = 7.5K$ speech embeddings, exceeding the 8K context limit of Llama-3-8B-Instruct if combined with the translation tokens.

Acoustics Speech and Signal Processing Proceedings, volume 1, pages I–I.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, Mexico City, Mexico. Association for Computational Linguistics.

W. Ronny Huang, Shuo yiin Chang, David Rybach, Rohit Prabhavalkar, Tara N. Sainath, Cyril Allauzen, Cal Peyser, and Zhiyun Lu. 2022. E2e segmenter: Joint segmenting and decoding for long-form asr. *Preprint*, arXiv:2204.10749.

Javier Iranzo-Sánchez, Jorge Iranzo-Sánchez, Adrià Giménez, Jorge Civera, and Alfons Juan. 2024. Segmentation-free streaming machine translation. *Transactions of the Association for Computational Linguistics*, 12:1104–1121.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 39–55, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

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.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech 2017*, pages 498–502.

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

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

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online. Association for Computational Linguistics.

Sara Papi, Peter Polak, Ondřej Bojar, and Dominik Macháček. 2024b. How "real" is your real-time simultaneous speech-to-text translation system? *Preprint*, arXiv:2412.18495.

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

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *ArXiv*, abs/2108.12409.

Matthew Raffel, Victor Agostinelli, and Lizhong Chen. 2024. Simultaneous masking, not prompting optimization: A paradigm shift in fine-tuning LLMs for simultaneous translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18302–18314, Miami, Florida, USA. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796, Online. Association for Computational Linguistics.

Felix Schneider and Alexander Waibel. 2020. Towards stream translation: Adaptive computation time for simultaneous machine translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 228–236, Online. Association for Computational Linguistics.

Jianlin Su. 2023. Rectified rotary position embeddings. `https://github.com/bojone/rerope`.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C).

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. 2023. A length-extrapolatable transformer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14590–14604, Toronto, Canada. Association for Computational Linguistics.

Minghan Wang, Thuy-Trang Vu, Yuxia Wang, Ehsan Shareghi, and Gholamreza Haffari. 2024. Conversational simulmt: Efficient simultaneous translation with large language models. *Preprint*, arXiv:2402.10552.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

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

Takenori Yoshimura, Tomoki Hayashi, Kazuya Takeda, and Shinji Watanabe. 2020. End-to-end automatic speech recognition integrated with ctc-based voice activity detection. *Preprint*, arXiv:2002.00551.

Donglei Yu, Yang Zhao, Jie Zhu, Yangyifan Xu, Yu Zhou, and Chengqing Zong. 2025. SimulPL: Aligning human preferences in simultaneous machine translation. In *The Thirteenth International Conference on Learning Representations*.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. Real-TranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474, Online. Association for Computational Linguistics.

Hao Zhang, Nianwen Si, Yaqi Chen, Wenlin Zhang, Xukui Yang, Dan Qu, and Xiaolin Jiao. 2023. Tuning large language model for end-to-end speech translation. *Preprint*, arXiv:2310.02050.

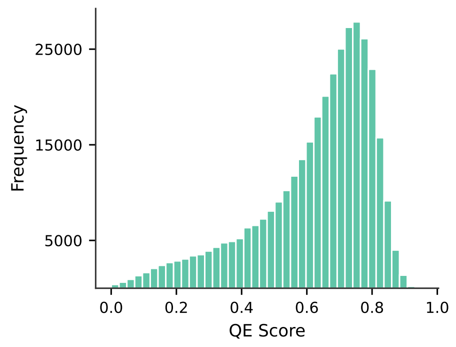


Figure 6: COMET-KIWI quality estimation score distribution on MuST-C en-zh data

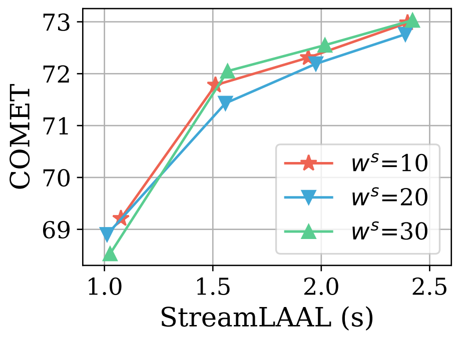


Figure 7: Impact of training-time sliding window size w^s of speech encoder.

A Additional Data Details

A.1 QE filtering and forward translation

We first use Whisper to perform automatic speech recognition (ASR) on all training segments. We then apply CometKiwi[7] to estimate the quality of ASR outputs by computing quality estimation (QE) scores between the ASR results and the reference text. As shown in Figure 6, we retain only instances where the QE score is greater than 0.5, which accounts for 78.64% of the data, resulting in a total of 280K instances.

Upon further inspection, we observed that many filtered-out cases exhibited acceptable word error rates (WER) between the ASR outputs and the source text. To recover these cases, we performed forward translation using the 7B version of TowerInstruct [8]on the source text using TowerInstruct with the following decoding settings: temperature = 0.0 and frequency penalty = 0.1. The translations were generated using vLLM.

[7]https://huggingface.co/Unbabel/wmt23-cometkiwi-da-xxl

[8]https://huggingface.co/Unbabel/TowerInstruct-7B-v0.2

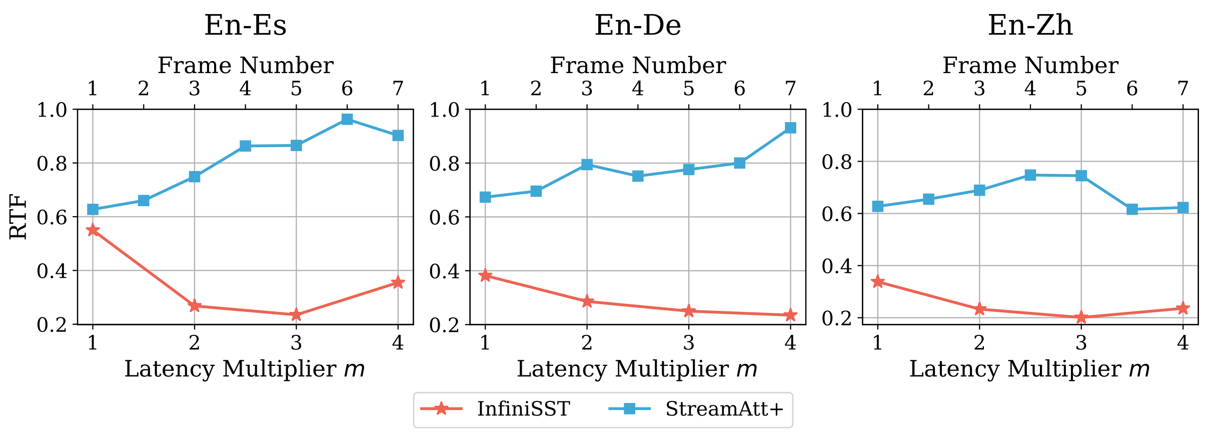


Figure 8: The Real-Time-Factor of InfiniSST and baseline StreamAtt+.

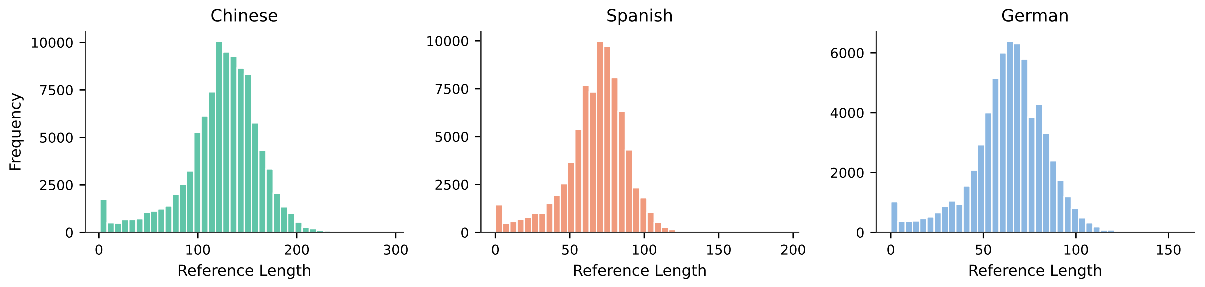


Figure 9: Reference length distribution of SST trajectories on MuST-C En-Zh, En-Es, and En-De.

A.2 Dataset Statistics

The MuST-C dataset used in our experiments consists of 105,647 instances for En-Zh, 88,725 for En-Es, and 70,037 for En-De.

Figure 9 shows the reference length distribution across these language pairs.

For En-Zh, the reference text length averages 124.32 characters, with a maximum of 444. En-Es has significantly longer references, averaging 400.06 characters and reaching a maximum of 1,116. En-De also exhibits long references, with an average of 419.47 characters and a maximum of 957. Spanish reference lengths in word count average 67.1 words, with a median of 70.0 and a 90th percentile of 90.0. German references are slightly shorter, averaging 63.6 words, with a median of 65.0 and a 90th percentile of 87.0.

En-Zh segments average 26.85 seconds, En-Es 25.25 seconds, and En-De 26.11 seconds, all with a maximum of 28.80 seconds, reflecting speech-text alignment across languages.

A.3 Examples of Trajectory

We show additional examples of trajectory in Figure 10.

B Additional Experiment Results

Impact of speech encoder window size during training w^s is shown in Figure 7. The RTF of InfiniSSTand baseline StreamAtt+ is shown in Figure 8.

We need to go
Wir müssen
We need to go for more
Wir müssen uns für mehr
We need to go for more research funding.
Wir müssen uns für mehr Forschungsgelder einsetzen.

If you gave me
Wenn Sie mir
If you gave me only one wish for the
Wenn Sie mir nur einen Wunsch für die
If you gave me only one wish for the next 50 years
Wenn Sie mir nur einen Wunsch für die nächsten 50 Jahre
If you gave me only one wish for the next 50 years — I could pick
Wenn Sie mir nur einen Wunsch für die nächsten 50 Jahre
If you gave me only one wish for the next 50 years — I could pick who's president,
Wenn Sie mir nur einen Wunsch für die nächsten 50 Jahre gestatten, ich könnte den Präsidenten auswählen.

These breakthroughs,
Diese Durchbrüche
These breakthroughs, we need to
Diese Durchbrüche müssen wir
These breakthroughs, we need to move those at full
Diese Durchbrüche müssen wir mit Vollgas verfolgen
These breakthroughs, we need to move those at full speed, and we can
Diese Durchbrüche müssen wir mit Vollgas verfolgen und
These breakthroughs, we need to move those at full speed, and we can measure that in terms of
Diese Durchbrüche müssen wir mit Vollgas verfolgen und das können wir messen: in
These breakthroughs, we need to move those at full speed, and we can measure that in terms of companies,
Diese Durchbrüche müssen wir mit Vollgas verfolgen und das können wir messen: in Firmenzahlen,
These breakthroughs, we need to move those at full speed, and we can measure that in terms of companies, pilot projects,
Diese Durchbrüche müssen wir mit Vollgas verfolgen und das können wir messen: in Firmenzahlen, in Pilotprojekten
These breakthroughs, we need to move those at full speed, and we can measure that in terms of companies, pilot projects, regulatory
Diese Durchbrüche müssen wir mit Vollgas verfolgen und das können wir messen: in Firmenzahlen, in Pilotprojekten
These breakthroughs, we need to move those at full speed, and we can measure that in terms of companies, pilot projects, regulatory things that have been changed.
Diese Durchbrüche müssen wir mit Vollgas verfolgen und das können wir messen: in Firmenzahlen, in Pilotprojekten und Regulierungsänderungen.

Figure 10: Three examples of En-De trajectory.