

# SASST: Leveraging Syntax-Aware Chunking and LLMs for Simultaneous Speech Translation

Zeyu Yang<sup>1,2</sup>, Lai Wei<sup>1</sup>, Roman Koshkin<sup>3</sup>, Xi Chen<sup>1</sup>, Satoshi Nakamura<sup>1,2,4</sup>

<sup>1</sup>The Chinese University of Hong Kong, Shenzhen, China

<sup>2</sup>Shenzhen Loop Area Institute, Shenzhen, China

<sup>3</sup>Okinawa Institute of Science and Technology, Japan

<sup>4</sup>Nara Institute of Science and Technology, Japan

zeyuyang1@link.cuhk.edu.cn, s.nakamura@cuhk.edu.cn

## Abstract

This work proposes a grammar-based chunking strategy that segments input streams into semantically complete units by leveraging dependency relations such as noun phrase boundaries, verb-object structures, and punctuation cues. The method ensures chunk coherence and minimizes semantic fragmentation.

Building on this mechanism, we present SASST (Syntax-Aware Simultaneous Speech Translation), an end-to-end framework integrating a frozen Whisper encoder and a decoder-only large language model. The unified architecture dynamically produces translation tokens or wait decisions to jointly optimize translation timing and content, while target-side reordering mitigates word-order divergence.

Experiments on the CoVoST2 multilingual corpus (En→De, Zh, Ja) demonstrate substantial translation quality improvements and validate the effectiveness of syntactic structures in LLM-driven simultaneous speech translation systems.

**Code** — <https://github.com/zeyuyang-906/SSAST>

## Introduction

Simultaneous speech translation (SimulST) aims to generate target-language translations in real time while listening to ongoing source speech. Unlike offline translation, where the entire input is available before translation begins, SimulST must operate under streaming constraints and make decisions dynamically, balancing three often competing goals: translation quality, latency, and output coherence.

Traditional SimulST pipelines typically consist of multiple independent modules, such as automatic speech recognition (ASR), segmentation, and neural machine translation (NMT) (Ma et al. 2018; Zeng, Li, and Liu 2021). While modular designs provide flexibility, they also suffer from error propagation, latency accumulation, and a mismatch between training and inference. In particular, segmentation and triggering decisions are often based on heuristics or shallow models, lacking deep contextual reasoning and limiting adaptability to varying speech patterns.

Recent progress in large language models (LLMs) has revealed strong abilities in language generation, contextual reasoning, and task generalization (Brown et al. 2020;

Chowdhery et al. 2022; Achiam et al. 2023). This has motivated research into LLM-based SimulST, where powerful sequence modeling capabilities are leveraged for low-latency translation. However, existing approaches often retain external policy modules or handcrafted segmentation strategies (Zhang and Feng 2023), separating “when to translate” from “what to translate” and thereby limiting interpretability and joint optimization.

In this work, we propose a linguistically motivated, data-driven framework that **internalizes read/write decision-making into an instruction-tuned LLM**, unifying segmentation and translation within a single model. Instead of applying predefined segmentation rules during inference, we generate **chunk-aligned supervision** based on syntactic and semantic boundaries and use it in a two-stage training strategy to teach the model to predict explicit <WAIT> tokens alongside translation tokens. This enables the LLM to autonomously learn when and what to translate, guided by linguistic structure but without relying on external alignment tools or policy heads. Inspired by human interpreters, who naturally pause at syntactic or semantic boundaries, our approach yields translations that are more coherent and interpretable under streaming constraints.

To further improve output fluency for language pairs with divergent word orders, we incorporate a chunk-aware reordering mechanism that rearranges translated segments into the natural target-language order. Our framework is model-agnostic and can be instantiated with different decoder-only LLM backbones. In this work, we evaluate two representative backbones, LLaMA3-8B (Meta AI 2024) and Qwen3-8B (Yang et al. 2025), paired with a frozen Whisper encoder (Cao et al. 2012) for speech feature extraction, and operate under causal constraints with all segmentation, alignment, and translation decisions unified within the model itself.

**Contributions** Our main contributions are:

- We present a unified, end-to-end SimulST system that integrates translation generation and read/write decision-making into a single LLM.
- We propose a linguistically motivated chunk supervision method and a two-stage training strategy that enables the model to autonomously learn translation triggering through explicit <WAIT> token prediction, removing the

need for external decision modules or handcrafted rules.

- We design a chunk-aware reordering mechanism to improve translation coherence for language pairs with divergent word orders.

## Related Work

Simultaneous speech translation (SimulST) aims to deliver translations while speech input is still ongoing, requiring models to balance translation fidelity and latency.

### Cascaded Systems

Early SimulST systems predominantly adopted a **cascaded architecture** consisting of automatic speech recognition (ASR) followed by machine translation (MT) (Oda et al. 2014; Le, Lecouteux, and Besacier 2017). While effective, these pipelines suffered from error propagation and increased latency due to module coupling. Recent cascaded approaches have leveraged powerful pre-trained models to improve translation quality and latency control. For example, **BeaverTalk** (Raffel, Agostinelli, and Chen 2025a) combines Whisper ASR with an LLM-based translation module. Although these systems reduce latency and improve quality, their multi-module design still inherently couples recognition and translation processes, limiting joint optimization.

### End-to-End SimulST

To overcome the limitations of cascaded architectures, **end-to-end (E2E) SimulST** models directly map input speech to translations within a single unified neural network, avoiding explicit intermediate transcriptions and module coupling (Berard et al. 2016; Weiss et al. 2017; Bansal et al. 2018; Ren et al. 2020). By integrating acoustic modeling, language modeling, and translation into a single optimization objective, these systems jointly balance latency and translation quality. Early approaches adopted encoder–decoder frameworks with streaming encoders and monotonic attention to handle partial speech input, enabling low-latency generation without waiting for utterance completion. Subsequent work further leveraged pretraining, adaptive alignment mechanisms, and multi-task objectives (e.g., simultaneous ASR + translation) to improve robustness and reduce lag. Despite their success, these models often require carefully designed read/write policies or specialized attention modules to handle the streaming nature of speech, and their decision-making process for when to emit translations remains either fixed or dependent on external heuristics. This limitation has motivated the recent shift towards incorporating large language models (LLMs) into SimulST, aiming to exploit their strong reasoning and generative capabilities while reducing reliance on handcrafted decision modules.

### LLM-based SimulST

Recently, **large language models (LLMs)** have been introduced into SimulST to exploit their strong reasoning and generation capabilities. **TransLLaMA** (Koshkin, Sudoh, and Nakamura 2024) is one of the earliest works to use LLMs for integrated read/write policy learning, showing that

translation triggering decisions can be learned jointly with content generation. **SimulS2S-LLM** (Deng et al. 2025) is the first to extend speech LLMs for simultaneous speech-to-speech translation (Simul-S2ST), leveraging boundary-aware speech prompts and a test-time wait-k policy to unlock streaming capability for offline-trained LLMs. **StreamUni** (Guo et al. 2025) further explores unifying segmentation, translation, and generation within a single model using multi-stage reasoning steps. These approaches demonstrate the potential of LLM-based architectures for streaming translation but often rely on explicit decision policies or intermediate reasoning stages to determine translation triggers, instead of fully integrating decision-making into the translation process itself.

### Read/Write Policies and Segmentation Strategies

A key challenge in SimulST is deciding when to emit translation tokens, commonly referred to as the **read/write policy**. Fixed strategies such as **wait-k** (Ma et al. 2018) and fixed-length chunking (Ma et al. 2021) offer predictable latency but lack adaptability. Adaptive approaches learn context-dependent policies through attention analysis (Papi, Negri, and Turchi 2022; Papi, Turchi, and Negri 2023), information-flow estimation (Zhang and Feng 2022), or segmentation-based decision making (Zhang et al. 2022; Dong et al. 2021). Systems like **EASiST** (Fu et al. 2025) introduce lightweight policy heads, while **SeqPOS** (Xu et al. 2025) frames translation as a sequential decision-making problem using reinforcement learning. Although effective, these approaches often depend on auxiliary classifiers or handcrafted cues, which increase system complexity and limit interpretability.

Segmentation-based strategies represent another research trend, where models learn to identify translation trigger points at semantically consistent boundaries. Examples include **RealTrans** (Zeng, Li, and Liu 2021) with trigger-based decoding, **MoSST** (Dong et al. 2021) emphasizing modular SimulST design, and **DiSeg** (Zhang and Feng 2023) using differentiable segmentation for improved trigger learning. These methods improve timing interpretability but still treat segmentation as an external or auxiliary process.

### Our Approach

In contrast, our work adopts a **linguistically motivated and data-driven perspective**: we identify syntactic and semantic chunk boundaries in bilingual corpora and use them as supervision in a **two-stage training procedure** to internalize decision-making into the model itself. Rather than relying on separate policy heads, segmentation modules, or heuristic boundary rules, our model jointly produces translations and explicit `<WAIT>` tokens, learning context-sensitive translation triggers directly within the generation process. This design yields a unified and interpretable SimulST model that integrates boundary reasoning and translation in a single LLM-based architecture, offering a compact alternative to approaches requiring external decision components.

## Method

### Syntax-Aware Chunking and Chunk-Level Alignment

A core component of our simultaneous speech translation system is a syntax-aware chunking policy that supervises both read/write decisions and translation timing. Unlike fixed windowing or pause-based segmentation methods, our approach leverages syntactic information to decide when an input segment is semantically complete and ready for translation. This enables the system to produce translation units that align with meaningful linguistic constituents such as clauses and noun phrases, improving semantic focus and fluency under streaming constraints.

To obtain chunk boundaries, we parse source sentences using the `en_core_web_trf` model from spaCy, which provides token-level part-of-speech tags and dependency relations. Chunk segmentation is guided by syntactic boundaries derived from noun phrases (NP), verb phrases (VP), and prepositional phrases (PP), as well as punctuation and dependency transitions (e.g., `nsubj`  $\rightarrow$  `VERB`). Rule-based constraints ensure that each chunk forms a semantically coherent unit and does not exceed a maximum span of seven tokens.

To train the model to learn streaming read/write decisions, we construct chunk-aligned bilingual data. Given each chunked source utterance, we first obtain fine-grained word-level timestamps using a Whisper-based speech recognizer. These timestamps are then aligned to target translations using SimAlign (Sabet, Dufter, and Schütze 2020), which yields soft bilingual word correspondences. For each chunk boundary, the aligned target words are grouped into a translation segment, and a special `<WAIT>` token is inserted for segments where the model must delay translation. This alignment produces training supervision that couples source segmentation with target output timing, enabling causal training of read/write policies without relying on manually annotated delays. An example of this chunk-based alignment and its effect on streaming output is illustrated in Figure 1.

### Target-Side Reordering

While syntax-aware chunking determines when to start translation, incremental models must also learn what to output when only partial source context is available. Directly using the original target sentence can mislead the model because many target words (e.g., verbs in German or function words in Japanese) appear late and depend on unseen source context. To address this, we perform a lightweight target-side reordering step to construct training targets that reflect the temporal structure of incremental decoding.

Given chunk-level source-target alignments, we rearrange target tokens within each chunk according to their alignment indices and insert special `<WAIT>` tokens for positions where the model should delay output until more source context arrives. This transformation preserves lexical content and grammaticality of the final translation (the reordered target can be deterministically converted back to the original), but it exposes the model to realistic streaming

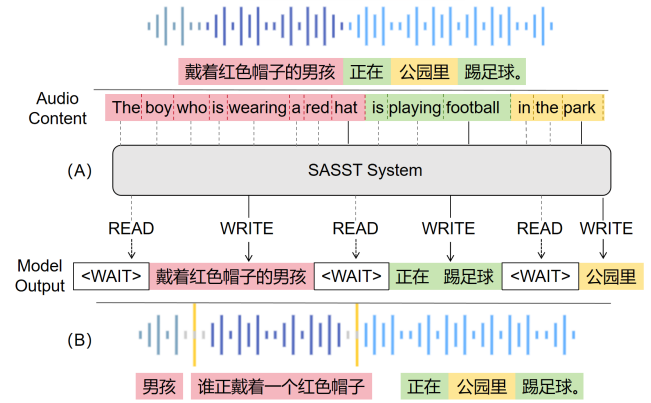


Figure 1: Comparison of translation behaviors with syntax-aware training versus pause-based chunking. (A) With syntax-aligned training data, our model learns to emit `<WAIT>` tokens when encountering incomplete semantic units, delaying generation until a complete syntactic chunk is observed, producing coherent partial translations. (B) In contrast, pause-based chunking leads to premature commitments and fragmented outputs, highlighting the advantage of syntax-aware chunking in preserving semantic integrity under streaming constraints.

scenarios where partial outputs and waiting decisions are required. Figure 2 illustrates an example: the original translation (left) places certain arguments and verbs late in the sentence, while our reordered target (right) distributes available words earlier and uses `<WAIT>` placeholders where future context is necessary. This supervision allows the model to produce fluent partial outputs while maintaining causal consistency during streaming.

### System Architecture

Our system adopts a unified end-to-end architecture that directly maps speech input to translated output under simultaneous translation constraints. Unlike previous designs that rely on multiple separate modules, such as independent chunk policy models, external alignment components, and standalone translation decoders (Oda et al. 2014; Ma et al. 2018; Zeng, Li, and Liu 2021; Bahar et al. 2020), our approach integrates multiple key functionalities into a single language model backbone. This design allows the model to learn to segment and translate simultaneously within a cohesive generative process, reducing inter-module complexity and improving overall efficiency.

The system consists of a frozen Whisper encoder and a Qwen3-based language model. Recent work has explored integrating decoder-only LLMs with speech encoders for streaming tasks (Chen et al. 2024). Building on this direction, our model embeds chunk-aware reasoning into the generation loop, enabling fine-grained control over read/write decisions and unifying segmentation with translation.

In our design, chunking and generation are unified into a single autoregressive language modeling task. The model is trained on streaming sequences, allowing it to learn natural pause points, maintain coherence over time, and operate un-

Original			Semantic Aligned		
en		zh	en		zh
1	He	他	He		他
2	took	把	took		<WAIT>
3	one	人类	one		<WAIT>
4	of	最严重的	of		<WAIT>
5	the	祸害	the		<WAIT>
6	worst	之一	worst		<WAIT>
7	scourges	从	scourges		<WAIT>
8	of	我们身边	of		<WAIT>
9	mankind	带走了	mankind		把
10	away	。	away		人类
11	from		from		最严重的
12	us		us		祸害
13	.		.		之一
14			---		从
15			---		我们身边
16			---		带走了
17			---		。

Figure 2: Illustration of the target-side reordering strategy. Target tokens are reordered to match chunk-level source alignment, and special <WAIT> tokens mark positions requiring future context, providing explicit supervision for streaming generation timing.

der causal decoding constraints. Compared to systems with distinct decision and generation stages, this structure simplifies deployment, reduces error propagation, and enables more effective utilization of large language models for both segmentation and translation. Figure 3 provides an overview of our architecture, illustrating the interaction between the Whisper encoder, chunk policy mechanism, and autoregressive translation process.

We fine-tune the model on streaming speech data derived from the CoVoST2 corpus (Wang, Wu, and Pino 2020), a large-scale multilingual speech translation dataset based on the Common Voice project, featuring diverse speakers, accents, and spontaneous speech patterns. Syntactic chunk boundaries are first extracted from the reference transcriptions using spaCy (AI 2020) and projected back to the source audio via their time-aligned word boundaries, yielding a set of audio segments with syntactically informed translation points. During training, <WAIT> tokens are inserted between non-aligned regions to supervise timing behavior, and the model is trained end-to-end to jointly learn segmentation and translation under causal constraints.

## Model Training

We adopt a two-stage training procedure to equip the model with both high-quality offline translation capability and read/write decision capability for streaming.

**Stage 1: Offline Translation.** In the first stage, we train the model on full-sentence speech translation pairs  $\mathcal{D}_{\text{offline}} = \{(S, Y)\}$ , where  $S = (s_1, \dots, s_T)$  is the input audio waveform and  $Y = (y_1, \dots, y_N)$  is the full target sentence. The speech encoder  $F_e(\cdot)$ , initialized from Whisper, maps input audio to acoustic features  $H = F_e(S)$ . A lightweight projection layer  $F_p(\cdot)$  converts  $H$  into the embedding space of the LLM decoder  $F_d(\cdot)$  (either LLaMA3-8B or Qwen3-

Algorithm 1: Syntax-Aware Chunk-Based Streaming Training

---

**Require:** Chunk-aligned dataset  $\mathcal{D}_{\text{stream}}$ , initial parameters  $\theta_0$

- 1: **for** each  $(S, Y') \in \mathcal{D}_{\text{stream}}$  **do**
- 2:   Encode speech:  $H = F_p(F_e(S))$
- 3:   **for** each token position  $i$  **do**
- 4:     Predict next token:  $\hat{y}_i = F_d(H_{\leq i}, y'_{<i}; \theta)$
- 5:     Compute loss:  $\mathcal{L}_i = \text{CE}(\hat{y}_i, y'_i)$
- 6:   **end for**
- 7:   Update parameters:  $\theta \leftarrow \theta - \eta \nabla_{\theta} \sum_i \mathcal{L}_i$
- 8: **end for**

---

8B). We freeze the high-level decoder layers and optionally the speech encoder, fine-tuning only the projection and low-level parameters using LoRA adapters (Hu et al. 2022). The training objective is the standard cross-entropy loss:

$$\mathcal{L}_{\text{offline}} = -\frac{1}{N} \sum_{i=1}^N \log P(y_i | H, y_{<i}; \theta). \quad (1)$$

**Stage 2: Chunk-Aligned Streaming.** To enable read/write decision learning, we further fine-tune the offline model on chunk-aligned data  $\mathcal{D}_{\text{stream}} = \{(S, Y')\}$ , where the target sequence  $Y'$  is augmented with an explicit <WAIT> token indicating when the model should pause writing output and continue reading input:

$$Y' = [t_1, \text{<WAIT>}, t_2, \dots, t_K]. \quad (2)$$

The training objective remains the cross-entropy loss but over the extended vocabulary including the <WAIT> token:

$$\mathcal{L}_{\text{stream}} = -\frac{1}{|Y'|} \sum_{i=1}^{|Y'|} \log P(y'_i | H_{\leq i}, y'_{<i}; \theta). \quad (3)$$

Unlike systems that rely on external segmenters such as SHAS to control read/write behavior, our model learns read/write decisions end-to-end through the explicit use of <WAIT> tokens, eliminating the need for a separate segmentation module.

## Streaming Inference and Prompt Encoding

**1) Simultaneous Inference.** Our system performs real-time speech translation in a streaming fashion via token-level incremental decoding. At each step, audio segmented by a sliding window is encoded by a frozen Whisper encoder, and the resulting embeddings are appended to the source context for the decoder-only language model. The model decides whether to output a translation token, a special <WAIT> token to defer output, or an <EOS> token to terminate the segment. During inference, <WAIT> tokens are discarded from the final translation but kept for latency evaluation using SimulEval (Ma et al. 2020).

**2) Incremental Prompt Encoding.** Our model adopts a multimodal prompt design inspired by recent LLM-based speech understanding systems. Each prompt consists of two parts: (1) a fixed instruction text that defines the translation

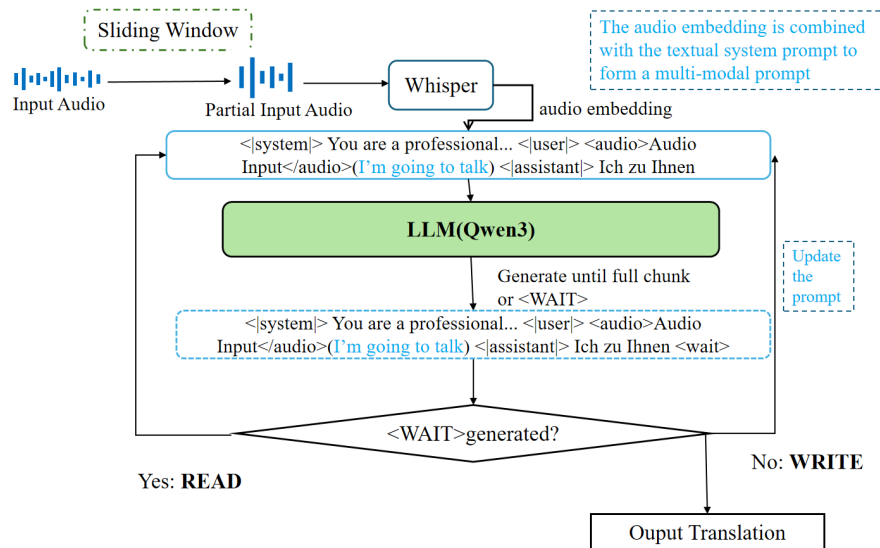


Figure 3: Overview of the SASST architecture for end-to-end simultaneous speech translation. Input audio is segmented by a sliding window and encoded by a frozen Whisper encoder. The resulting audio embeddings and textual instruction form a multi-modal prompt for a decoder-only LLM, which generates either translation tokens or a special `<WAIT>` token to control read/write decisions, enabling low-latency streaming translation.

task and streaming behavior, and (2) a sequence of audio-derived token embeddings extracted from the Whisper encoder. Unlike prior methods that rely on text transcripts or symbolic prompts, our system operates directly on speech inputs without intermediate ASR, enabling seamless end-to-end streaming translation.

The same multimodal prompt format is used during training and inference, which reduces domain shift and improves model consistency under streaming constraints. As decoding progresses, the prompt is updated incrementally by extending the source-side audio embedding stream and the target-side token history.

This design enables the model to simultaneously reason over speech context, track translation progress, and make timing decisions within a unified decoding process.

**3) Sliding Window Strategy.** To support real-time translation while maintaining causal access, we apply a sliding window strategy before audio encoding. Each input segment is derived from an overlapping 8-second audio window, formed by appending the latest  $\delta$  seconds of audio to the preceding  $8 - \delta$  seconds of buffered context. This setup preserves both local continuity and long-range acoustic dependencies, while avoiding access to future input. The stride parameter  $\delta$  (e.g., 0.5–2.0 seconds) directly controls the latency–quality tradeoff. The Whisper encoder processes each window to extract semantic audio embeddings, which are passed to the decoder for joint reasoning.

## Experimental Setup

### Data

We conduct experiments on the CoVoST2 dataset (Wang, Wu, and Pino 2020), which provides speech translation

pairs across multiple language directions. For this work, we focus on three directions: English→German (En→De), English→Chinese (En→Zh), and English→Japanese (En→Ja). The CoVoST2 dataset contains approximately 2,900 hours of speech covering 21 languages; for our selected pairs, we use the official training, validation, and test splits.

Following prior work (Dong et al. 2021; Zeng, Li, and Liu 2021), we evaluate on the official CoVoST2 test set for each language pair. In addition, to enable comparison with systems that report results on the ACL 60/60 benchmark, we also use this dataset for validation and testing.

### Evaluation Metrics

We evaluated the system performance using metrics that capture both translation quality and latency. For translation quality, we used BLEU calculated with SacreBLEU (Post 2018) and COMET (Rei et al. 2020). For latency, we used the Stream Length-Adaptive Average Lagging (Stream-LAAL) (Papi et al. 2024).

### Offline Translation Model

We train the offline translation model on full-sentence speech–text pairs using the AdamW optimizer ( $\beta_1 = 0.9, \beta_2 = 0.999$ ) with a learning rate of  $2.0 \times 10^{-4}$ , a warm-up ratio of 0.03, and gradient clipping set to 1.0. Training is performed for one epoch on  $4 \times V100$  GPUs with an effective batch size of 32 sentences (16 per GPU with gradient accumulation of 2). High-level decoder layers and optionally the speech encoder are frozen, while the projection layer and low-level parameters are fine-tuned with LoRA adapters. The best checkpoint is selected based on the BLEU score on the development set.

## Simultaneous Speech Translation Training

We initialize parameters from the offline translation model and fine-tune on chunk-aligned bilingual data with explicit `<WAIT>` tokens. The chunk boundaries and corresponding target-side reordering are derived from our syntax-aware chunking and alignment pipeline described in Section . This pipeline produces training examples where each chunk is paired with its aligned translation segment and `<WAIT>` placeholders for positions requiring delayed output. Such supervision allows the model to learn when to buffer additional input and when to emit translation in an incremental setting. During inference, we vary input chunk sizes  $\{0.5s, 0.75s, 1.0s, 1.5s, 2.0s, 2.5s, 3.0s\}$  to explore different quality–latency trade-offs without changing model architecture.

## Baseline Systems

We compare SASST with four representative simultaneous speech translation (SimulST) systems, all of which leverage large language models (LLMs) and represent the current state of streaming SimulST:

- **BeaverTalk** (Raffel, Agostinelli, and Chen 2025b): A cascaded pipeline with VAD-based segmentation, Whisper Large V2 ASR, and a LoRA-tuned Gemma 3 12B LLM, supporting both low- and high-latency settings.
- **NAIST-2025** (Tan et al. 2025): An end-to-end ST model using a Whisper encoder, DeCo projector, and Qwen LLM, with streaming enabled by the Local Agreement (LA) policy and online SHAS segmentation.
- **InfiniSST** (Ouyang, Xu, and Li 2025): A system for unbounded speech, featuring a chunkwise-causal encoder, speech–text adapter, and multi-turn LLM decoder with KV cache to reduce computation-aware latency.
- **SeamlessM4T-IWSLT**<sup>1</sup>: The official IWSLT 2025 baseline derived from Meta’s SeamlessM4T (Barrault et al. 2023), using a fixed-length segmenter (length 8) to provide stable results across latency regimes.

These baselines already cover the most recent streaming LLM-based approaches and thus reflect the current practical efficiency–quality trade-offs in the field. All models are evaluated on the same `ac160/60` splits and latency settings for fair comparison.

## Main Results

### Main Results

Table 1 lists the BLEU scores of our SASST model at representative latency points, measured by StreamLAAL. To better illustrate the latency–quality trade-off, Figures 4a–4c (summarized in Figure 4) compare SASST with representative state-of-the-art SimulST systems from the IWSLT 2025 shared task, including SeamlessM4T-IWSLT, BeaverTalk (both low- and high-latency configurations), NAIST-2025, and InfiniSST, on English–German, English–Chinese, and English–Japanese translation tasks.

<sup>1</sup><https://github.com/pe-trik/iwslt25-baselines>

Latency (ms)	1800	2500	3200	4000
<b>SASST (En→De) BLEU</b>	24.6	26.2	27.7	28.0
<b>SASST (En→De) COMET</b>	0.729	0.744	0.758	0.762
<b>SASST (En→Zh) BLEU</b>	34.1	38.5	40.2	41.5
<b>SASST (En→Zh) COMET</b>	0.706	0.767	0.779	0.797
<b>SASST (En→Ja) BLEU</b>	18.1	22.5	23.9	24.5
<b>SASST (En→Ja) COMET</b>	0.683	0.727	0.743	0.772

Table 1: BLEU and COMET scores of SASST at latency levels near 1.8 s, 2.5 s, 3.2 s, and 4.0 s (StreamLAAL), evaluated on CoVoST2 En→De, En→Zh, and En→Ja. Exact average latencies may vary slightly due to dynamic chunking.

Compared to the official IWSLT 2025 baseline, which adopts a fixed-length segmentation strategy and achieves stable but syntax-agnostic performance, SASST demonstrates a superior quality–latency balance across all three language directions within the 2–3.5 second StreamLAAL latency range. For languages such as Chinese and Japanese, which exhibit substantial word-order differences, SASST effectively avoids fragmented outputs caused by insufficient context, maintaining coherent and semantically complete translations. This advantage stems from SASST’s two-stage training on syntax-aligned chunking data, which enables the model to identify semantically complete units and trigger translations at appropriate moments. By internalizing this decision-making capability into the model itself, SASST eliminates the need for external segmentation or triggering modules, thereby reducing error propagation and additional latency, and ultimately achieving more stable and efficient simultaneous speech translation.

## Ablation Study

To deepen the understanding of our approach, we conduct extensive analyses under a fixed input chunk size of 2.0 s to ensure fair comparison. We introduce each analytical experiment in detail below.

### Impact of Syntax-Aware Chunking

To isolate the effect of syntax-aware chunking, we re-trained SASST using a fixed-length segmentation policy and compared it with our syntax-aware segmentation on the En→Zh language pair, which exhibits significant word-order differences. As shown in Table 2, removing syntax-awareness causes a substantial drop of more than 15 BLEU points (38.5 → 23.2) under comparable latency. This demonstrates that triggering translations at linguistically meaningful boundaries rather than at arbitrary fixed windows is critical for translation quality and fluency.

We further analyzed the boundary alignment of the learned chunking policy. We measured the proportion of translation triggers that fall within one token of a syntactic boundary obtained from an offline dependency parser. Our syntax-aware model aligns with syntactic boundaries 82% of the time, compared to only 23% for fixed-length segmentation, indicating that SASST successfully learns to trigger translations near syntactic boundaries, resulting in more coherent and semantically complete outputs.

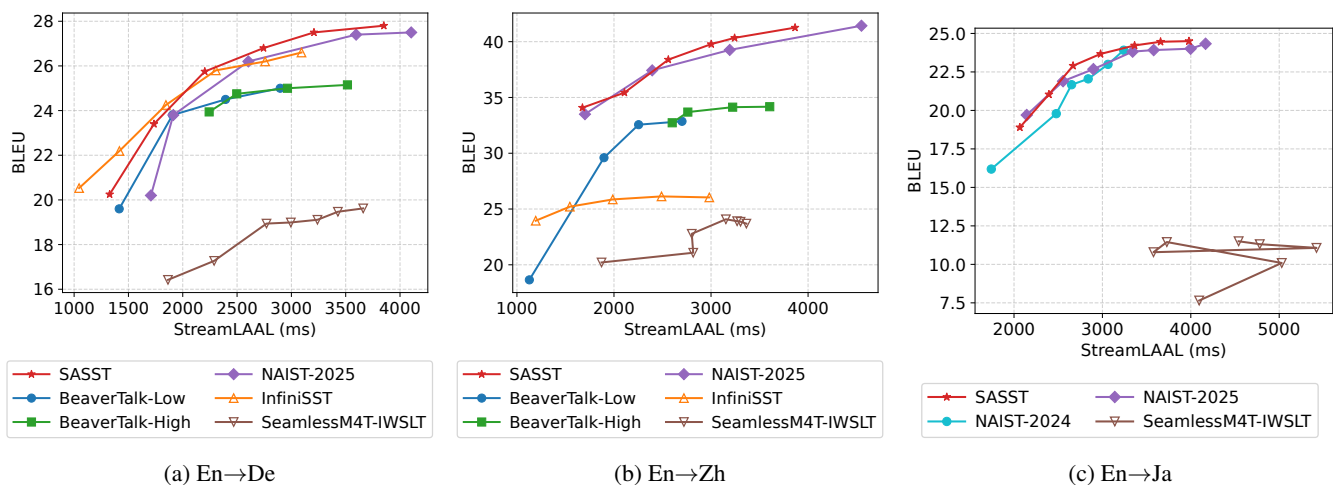


Figure 4: Performance of SASST and IWSLT 2025 baseline systems on acl60/60 En→De, En→Zh, and En→Ja datasets. We report BLEU versus StreamLAAL latency to evaluate the quality–latency trade-off for different language pairs with varying syntactic divergence.

Segmentation	BLEU	Boundary Alignment
Syntax-aware (Ours)	<b>38.5</b>	<b>82%</b>
Fixed-length	23.2	23%

Table 2: Impact of segmentation strategy on En→Zh translation. Boundary alignment measures the proportion of translation triggers aligned with syntactic boundaries.

Language Pair	LLM	BLEU	StreamLAAL (ms)
En→Ja	LLaMA3-8B	25.674	4267
	Qwen3-8B	<b>27.279</b>	4381
En→Zh	LLaMA3-8B	37.048	3303
	Qwen3-8B	<b>40.216</b>	3247
En→De	LLaMA3-8B	26.684	3623
	Qwen3-8B	<b>27.892</b>	3857

Table 3: Impact of LLM backbone on SASST performance.

## Influence of LLMs

We further examine how different foundation models impact SASST performance. We compare Qwen3-8B (default) with LLaMA3-8B across three language pairs. Table 3 shows that Qwen3-8B consistently outperforms LLaMA3-8B by 1.2–3.2 BLEU, while latency remains comparable (3.2–4.4 s StreamLAAL). These findings indicate that SASST benefits from the stronger instruction-following and multilingual capabilities of Qwen3, yet its relative advantage from syntax-aware chunking and unified decoding policy is preserved across different LLM backbones.

These results indicate that aligning translation triggers with syntactic boundaries produces more coherent and semantically complete translations without increasing latency. Moreover, SASST maintains consistent gains across different LLM backbones, demonstrating robustness and scalability.

## Limitations

While our experiments demonstrate consistent improvements over strong streaming baselines, several limitations remain. First, our evaluation focuses on three high-resource language pairs on the CoVoST2 benchmark. Although these pairs cover typologically diverse structures, future work should investigate low-resource and code-switched scenarios to assess cross-domain generality. Second, our syntax-aware chunking relies on dependency parsers to determine linguistically meaningful translation triggers. Although modern parsers achieve over 90% labeled attachment score (LAS) on English, Chinese, Japanese, and German (Dozat and Manning 2016; Qi et al. 2020), their accuracy may degrade under noisy speech or out-of-domain inputs, potentially affecting chunk boundary quality. Finally, our chunk-alignment pipeline, while detailed in Section , introduces additional preprocessing steps that may influence downstream performance if implemented differently. Future work could explore parser-free approaches or joint optimization of segmentation and translation to further enhance robustness and portability.

## Conclusion

In this paper, we propose SASST, a novel LLM-driven simultaneous speech translation system that allows LLMs to determine translation timing and generate outputs concurrently with streaming speech. Experiments show that SASST achieves competitive translation quality while maintaining low latency, demonstrating strong potential for real-world streaming applications.

## Ethical Statement

This work utilizes publicly available large language models (e.g., Whisper, Qwen) for research purposes. Due to their probabilistic nature, these models may produce inaccurate or biased outputs, especially when handling sensitive content.

We use them only for research on speech translation, without deploying them in safety-critical scenarios or for making decisions that directly affect individuals. All datasets used in this work are publicly available and anonymized, and no additional human subjects were recruited for this study. We also used ChatGPT to assist with language refinement.

## Acknowledgments

This work was supported in part by the RFIS project W2531054 under NSFC.

## References

- Achiam, O. J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; ing Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; laine Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; abella Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Kaiser, L.; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, H.; Kiros, J. R.; Knight, M.; Kokotajlo, D.; Kondraciuk, L.; Kondrich, A.; Konstantinidis, A.; Kopic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C.; Lim, R.; Lin, M.; Lin, S.; teusz Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; drey Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D. P.; Mu, T.; Murati, M.; Murk, O.; M’ely, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Long, O.; O’Keefe, C.; Pachocki, J. W.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Pokorny, M.; Pocrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J. W.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M. D.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N. A.; Thompson, M.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C. L.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; ing Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2023. GPT-4 Technical Report.
- AI, E. 2020. spaCy: Industrial-strength Natural Language Processing in Python. In *Software available at <https://spacy.io>*.
- Bahar, P.; Wilken, P.; Alkhouli, T.; Guta, A.; Golik, P.; Matusov, E.; and Herold, C. 2020. Start-Before-End and End-to-End: Neural Speech Translation by AppTek and RWTH Aachen University. In *International Workshop on Spoken Language Translation*.
- Bansal, S.; Kamper, H.; Livescu, K.; Lopez, A.; and Goldwater, S. 2018. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *North American Chapter of the Association for Computational Linguistics*.
- Barrault, L.; Chung, Y.-A.; Meglioli, M. C.; Dale, D.; Dong, N.; Duquenne, P.-A.; Elsahar, H.; Gong, H.; Heffernan, K.; Hoffman, J.; et al. 2023. SeamlessM4T: Massively Multilingual & Multimodal Machine Translation. *arXiv preprint arXiv:2308.11596*.
- Berard, A.; Pietquin, O.; Servan, C.; and Besacier, L. 2016. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. In *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT)*.
- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; teusz Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *ArXiv*, abs/2005.14165.
- Cao, N.; Lin, Y.; Sun, X.; Lazer, D. M. J.; Liu, S.; and Qu, H. 2012. Whisper: Tracing the Spatiotemporal Process of Information Diffusion in Real Time. *IEEE Transactions on Visualization and Computer Graphics*, 18: 2649–2658.
- Chen, X.; Zhang, S.; Bai, Q.; Chen, K.; and Nakamura, S. 2024. LLaST: Improved End-to-End Speech Translation System Leveraged by Large Language Models. *arXiv preprint arXiv:2407.15415*.
- Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N. M.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B.

- Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; García, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omer-nick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Díaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K. S.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2022. PaLM: Scaling Language Modeling with Pathways. *ArXiv*, abs/2204.02311.
- Deng, K.; Chen, W.; Chen, X.; and Woodland, P. 2025. SimulS2S-LLM: Unlocking Simultaneous Inference of Speech LLMs for Speech-to-Speech Translation. In *Annual Meeting of the Association for Computational Linguistics*.
- Dong, Q.; Zhu, Y.; Wang, M.; and Li, L. 2021. Learning When to Translate for Streaming Speech. In *Annual Meeting of the Association for Computational Linguistics*.
- Dozat, T.; and Manning, C. D. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.
- Fu, B.; Yu, D.; Liao, M.; Li, C.; Chen, Y.; Fan, K.; and Shi, X. 2025. Efficient and Adaptive Simultaneous Speech Translation with Fully Unidirectional Architecture. *arXiv preprint arXiv:2501.12345*.
- Guo, S.; Li, X.; Zhang, S.; Liu, M.; Chen, W.; and Feng, Y. 2025. StreamUni: Achieving Streaming Speech Translation with a Unified Large Speech-Language Model. *arXiv preprint arXiv:2507.07803*.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.
- Koshkin, R.; Sudoh, K.; and Nakamura, S. 2024. TransLLaMa: LLM-based Simultaneous Translation System. *ArXiv*, abs/2402.04636.
- Le, N.-T.; Lecouteux, B.; and Besacier, L. 2017. Disentangling ASR and MT Errors in Speech Translation. In *Machine Translation Summit*.
- Ma, M.; Huang, L.; Xiong, H.; Zheng, R.; Liu, K.; Zheng, B.; Zhang, C.; He, Z.; Liu, H.; Li, X.; Wu, H.; and Wang, H. 2018. STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework. In *Annual Meeting of the Association for Computational Linguistics*.
- Ma, X.; Dousti, M. J.; Wang, C.; Gu, J.; and Pino, J. 2020. SimulEval: An Evaluation Toolkit for Simultaneous Translation. In *Proceedings of the EMNLP*.
- Ma, X.; Wang, Y.; Dousti, M. J.; Koehn, P.; and Pino, J. 2021. Streaming simultaneous speech translation with augmented memory transformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7523–7527. IEEE.
- Meta AI. 2024. The LLaMA 3 Herd of Models. Technical report, Meta AI.
- Oda, Y.; Neubig, G.; Sakti, S.; Toda, T.; and Nakamura, S. 2014. Optimizing Segmentation Strategies for Simultaneous Speech Translation. In *Annual Meeting of the Association for Computational Linguistics*.
- Ouyang, S.; Xu, X.; and Li, L. 2025. Infnisst: Simultaneous translation of unbounded speech with large language model. *arXiv preprint arXiv:2503.02969*.
- Papi, S.; Gaido, M.; Negri, M.; and Bentivogli, L. 2024. StreamAtt: Direct streaming speech-to-text translation with attention-based audio history selection. *arXiv preprint arXiv:2406.06097*.
- Papi, S.; Negri, M.; and Turchi, M. 2022. Attention as a Guide for Simultaneous Speech Translation. *arXiv preprint arXiv:2212.07850*.
- Papi, S.; Turchi, M.; and Negri, M. 2023. AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation. *ArXiv*, abs/2305.11408.
- Post, M. 2018. A Call for Clarity in Reporting BLEU Scores. In Bojar, O.; Chatterjee, R.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Yepes, A. J.; Koehn, P.; Monz, C.; Negri, M.; N  v  ol, A.; Neves, M.; Post, M.; Specia, L.; Turchi, M.; and Verspoor, K., eds., *Proceedings of the Third Conference on Machine Translation: Research Papers*, 186–191. Brussels, Belgium: Association for Computational Linguistics.
- Qi, P.; Zhang, Y.; Zhang, Y.; Bolton, J.; and Manning, C. D. 2020. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.
- Raffel, M.; Agostinelli, V.; and Chen, L. 2025a. BeaverTalk: Oregon State University’s IWSLT 2025 Simultaneous Speech Translation System. *ArXiv*, abs/2505.24016.
- Raffel, M.; Agostinelli, V.; and Chen, L. 2025b. BeaverTalk: Oregon State University’s IWSLT 2025 Simultaneous Speech Translation System. *arXiv preprint arXiv:2505.24016*.
- Rei, R.; Stewart, C.; Farinha, A. C.; and Lavie, A. 2020. COMET: A neural framework for MT evaluation. *arXiv preprint arXiv:2009.09025*.
- Ren, Y.; Liu, J.; Tan, X.; Zhang, C.; Qin, T.; Zhao, Z.; and Liu, T.-Y. 2020. SimulSpeech: End-to-End Simultaneous Speech to Text Translation. In *Annual Meeting of the Association for Computational Linguistics*.
- Sabet, M. J.; Dufter, P.; and Sch  tze, H. 2020. SimAlign: High Quality Word Alignments without Parallel Training Data using Static and Contextualized Embeddings. In *Findings*.
- Tan, H.; Widiaputri, R. F.; Saragih, J. M.; Ko, Y.; Sudoh, K.; Nakamura, S.; and Sakti, S. 2025. NAIST Simultaneous Speech Translation System for IWSLT 2025. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, 369–378.
- Wang, C.; Wu, A.; and Pino, J. 2020. Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*.
- Weiss, R. J.; Chorowski, J.; Jaitly, N.; Wu, Y.; and Chen, Z. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Interspeech*.

Xu, T.; Huang, Z.; Sun, J.; Cheng, S.; and Lam, W. 2025. SeqPO-SiMT: Sequential Policy Optimization for Simultaneous Machine Translation. *ArXiv*, abs/2505.20622.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.-C.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.-Q.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.-C.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report.

Zeng, X.; Li, L.; and Liu, Q. 2021. RealTranS: End-to-End Simultaneous Speech Translation with Convolutional Weighted-Shrinking Transformer. *arXiv preprint arXiv:2106.04833*.

Zhang, R.; He, Z.; Wu, H.; and Wang, H. 2022. Learning Adaptive Segmentation Policy for End-to-End Simultaneous Translation. In *Annual Meeting of the Association for Computational Linguistics*.

Zhang, S.; and Feng, Y. 2022. Information-Transport-based Policy for Simultaneous Translation. *arXiv preprint arXiv:2210.12357*.

Zhang, S.; and Feng, Y. 2023. End-to-End Simultaneous Speech Translation with Differentiable Segmentation. In *Annual Meeting of the Association for Computational Linguistics*.