SimulS2ST-Omni: Data-Efficient Streaming Speech-to-Speech Translation via Explicit Trajectory Supervision

Rongshen He[1,2], Xinyu Liang[1,2], Dekun Chen[1], Jiaqi Li[1], Mingjie Chen[1,2], Zhizheng Wu[1,2,*]

[1]The Chinese University of Hong Kong, Shenzhen
[2]Amphion Technology Co., Ltd.
rongshenhe@link.cuhk.edu.cn, wuzhizheng@cuhk.edu.cn
[*]Corresponding author

Abstract

Long-form streaming speech-to-speech translation (S2ST) requires incremental, unbounded translation under strict latency constraints. Existing methods typically suffer from sentence-bounded supervision or demand massive paired-S2ST supervision. We introduce a training recipe enabling a speech language model for sentence-level and long-form streaming S2ST using only ∼2k hours of paired cross-lingual S2ST data, layered atop auxiliary supervision. Anchored by auxiliary multitask training, our approach remains robust even when the paired-S2ST budget itself is reduced by 90%. Our core contribution, joint text-code trajectory supervision, schedules target text and acoustic semantic codes as a unified commitment path, eliminating the need for separate, unstable speech-side emission controllers. Furthermore, our two-stream Thinker–Talker factorization significantly outperforms unified-decoder baselines by decoupling linguistic reasoning from dense acoustic prediction to mitigate modality interference. Finally, our system achieves highly competitive quality-latency trade-offs on RealSI and ACL60/60-dev, matching state-of-the-art, closed-source S2ST systems such as LiveInterpret 2.0 on ASR-BLEU.

Project page: https://hasaki321.github.io/SimulS2ST-Omni.demo/

1 Introduction

Real-time, cross-lingual communication demands systems capable of translating continuous source speech into target speech with high fidelity and low latency. However, current Speech-to-Speech Translation (S2ST) approaches struggle to meet these demands efficiently. Cascaded systems suffer from inherent latency bottlenecks and compounding errors (Nakamura et al., 2006), while end-to-end Speech Language Models (SLMs) remain largely restricted to offline, sentence-bounded processing (Cheng et al., 2025b; Dong et al., 2023; Deng et al., 2025). Although continuous streaming S2ST systems have recently emerged, they rely heavily on massive synthetic supervision — often exceeding 40k hours of paired audio, or require complex, expert-annotated reinforcement learning (Labiausse et al., 2025, 2026; Cheng et al., 2025a). Thus, achieving robust, long-form streaming S2ST under *limited* paired supervision remains a fundamental, unresolved challenge.

This data bottleneck is exacerbated by two deep structural obstacles. First, long-form streaming is inherently a *latent path problem*. A model must autonomously navigate a read/wait/write policy without explicit step-by-step supervision. While trajectory-based methods address this for text-output streaming (Ouyang et al., 2025b; Wang et al., 2025; Yu et al., 2025), they fail for speech output, which requires the irrevocable, time-sensitive commitment of acoustic semantic codes. Second, the model must learn source-speech understanding, target-text planning, and dense target-code prediction from a very small amount of end-to-end evidence. Recent studies show that unified understanding and generation impose conflicting representation requirements (Shi et al., 2025; Xu et al., 2025c). Decoupling these pathways or modules has therefore emerged as a necessary strategy to alleviate this conflict (Wu et al., 2024; Chen et al., 2024b).

This paper overcomes these limitations through three strategic design choices. First, to maximize the efficiency of paired-S2ST supervision specifically—the scarcest resource in this setting—we construct only ∼2,000 hours of high-quality

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 26030–26058
October 24-29, 2026 ©2026 Association for Computational Linguistics

Chinese-English paired S2ST data from public Automatic Speech Recognition (ASR) and Speech-to-Text Translation (S2TT) corpora via cross-lingual alignment and monotonicity filtering, layered atop large-scale auxiliary ASR/S2TT/MT/TTS supervision that is comparatively abundant. Second, we introduce a *joint* commitment path — extending trajectory supervision to schedule target text and acoustic semantic codes simultaneously. This eliminates the need for unstable, separate speech-side emission controllers. Third, we establish that backbone architecture is paramount. We demonstrate that a two-stream Thinker–Talker factorization significantly mitigates interference between linguistic planning and dense code prediction, outperforming unified-decoder baselines under paired-S2ST data constraints.

We make the following contributions:

1. We deliver a streaming S2ST recipe capable of long-form streaming S2ST using only ∼2k hours of filtered paired-S2ST data, or less, by leveraging auxiliary multitask supervision.

2. We introduce a unified commitment path that schedules text and acoustic codes simultaneously without relying on closed-loop rewards or external policies.

3. We conduct controlled experiments showing that the two-stream architecture outperforms a unified-decoder baseline for speech translation under low-resource S2ST.

2 Related Work

Research on Speech-to-Speech Translation (S2ST) has increasingly shifted from offline systems (Inaguma et al., 2023; Communication et al., 2023a) toward streaming models that minimize latency. Streaming S2ST requires a model to autonomously decide both *what* to generate and *when* to commit based on incomplete speech input. Earlier task-specific streaming architectures typically relied on sentence-bounded cues, such as CTC alignment triggers (Zhang et al., 2024), transducer-style structures (Ma et al., 2024), or customized monotonic attention and latent-alignment segmentation (Communication et al., 2023b; Polák and Bojar, 2024). Although effective for short-form, pre-segmented utterances, these boundary-reliant methods fail to provide a stable training interface for continuous, unsegmented long-form speech. Furthermore, their limited reasoning capacities fall far short of modern large language models (LLMs).

Consequently, recent work has adopted Speech Language Models (SLMs) to inherit these advanced reasoning capabilities, generally clustering into unified-decoder (Wang et al., 2023; Zeng et al., 2024; Cheng et al., 2025b) and decoupled two-stream (Dong et al., 2023; Xu et al., 2025a) architectures. While state-of-the-art streaming SLMs such as Hibiki (Labiausse et al., 2025, 2026) and LiveInterpret 2.0 (Cheng et al., 2024, 2025a) successfully achieve unbounded speech-output streaming, they rely heavily on massive volumes of paired source-target audio, expert-annotated alignments, or complex reinforcement learning.

To mitigate the severe scarcity of paired-S2ST data, multiple workarounds have been proposed. Some approaches reduce dependence on paired audio by leveraging multitask or self-supervised representations (Lee et al., 2022; Zheng et al., 2026; Fang et al., 2024; Nachmani et al., 2024). Others, such as SimulS2S-LLM (Deng et al., 2025) and SimulU (Djanibekov et al., 2026), bypass it entirely through test-time wait-k policies or training-free cross-attention decision rules. However, these mitigation strategies remain highly sensitive to hyperparameters and heavily dependent on offline, pre-segmented alignments, preventing them from scaling efficiently to continuous, long-form streams.

To exploit the sequential reasoning of LLMs for unbounded continuous input, a complementary line of research models streaming as a multi-turn dialogue or as a long-form, text-only commitment path (Wang et al., 2025; Guo et al., 2025). Systems like InfiniSST (Ouyang et al., 2025b) successfully manage continuous streams by embedding the streaming policy directly into the autoregressive LLM, thereby leveraging its native pretraining intelligence. We bridge the remaining gap between these text-only trajectories and actual speech generation by introducing a *joint text-code commitment path*. By providing token-level joint supervision of text and acoustic codes along time-aligned boundaries, our approach fully preserves the reasoning capability of the LLM. This enables robust, long-form speech-output streaming under a highly constrained paired-data budget, eliminating the need for a separate speech-side emission controller (Sudoh et al., 2020; Le et al., 2025).

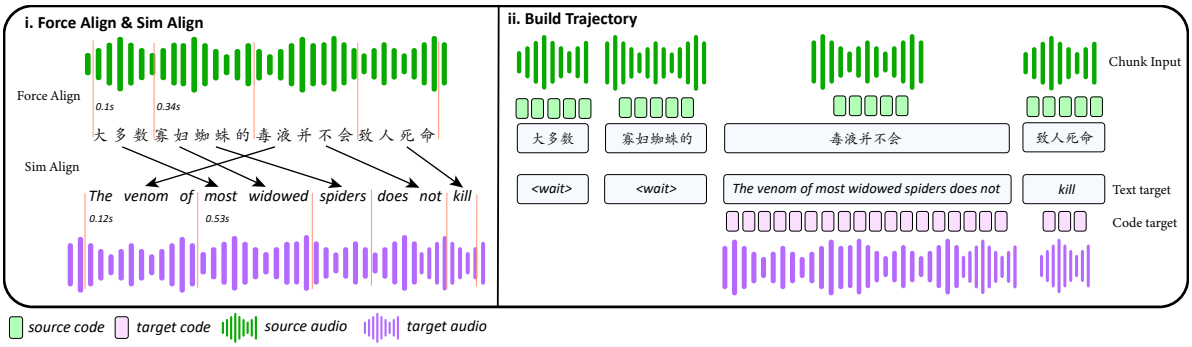

Figure 1: **Streaming trajectory construction. Step 1:** Word-level and cross-lingual alignments establish the earliest valid source prefix for each target word. **Step 2:** Target text and segmented target speech codes inherit monotonized boundaries and are grouped into discrete `read`/`wait`/`write` steps for joint emission.

3 Method

3.1 Problem Setup and Formulation

A paired-audio S2ST dataset consists of N training samples, denoted as $\{(\mathbf{X}_n, \mathbf{Y}_n^{\text{text}}, \mathbf{Y}_n^{\text{wav}})\}_{n=1}^{N}$, where $\mathbf{X}_n$ is the source speech, $\mathbf{Y}_n^{\text{text}}$ is the target text, and $\mathbf{Y}_n^{\text{wav}}$ is the target speech. For notational brevity, we omit the sample index n in the following sections.

Instead of modeling continuous waveforms directly, we represent the target speech using discrete semantic codes. Let $\mathbf{Y}^{\text{code}} = \mathcal{Q}(\mathbf{Y}^{\text{wav}})$ be the semantic token sequence extracted by a fixed speech tokenizer $\mathcal{Q}$. During inference, the waveform $\mathbf{Y}^{\text{wav}}$ is reconstructed from $\mathbf{Y}^{\text{code}}$ using a pre-trained, frozen flow-matching and vocoder backend $\mathcal{D}$ (Le et al., 2023; Chen et al., 2026; Siuzdak, 2023).

To train the model for streaming generation, we reformulate the joint prediction of target text and codes as a chunk-wise commitment process. We define a streaming **trajectory** $\boldsymbol{\tau} = \{(\mathbf{Y}_c^{\text{text}}, \mathbf{Y}_c^{\text{code}}, g_c)\}_{c=1}^{C}$ that partitions the joint target sequences into C chunks. Each chunk c is aligned to a source prefix bounded by the frame index g_c, representing the portion of the source speech the model must process before emitting the c-th chunk. These indices form a monotonically non-decreasing sequence $g_1 \leq g_2 \leq \cdots \leq g_C$, where the final index g_C corresponds to the total length of the source speech.

At each emission step c, generation is conditioned on the chunk context, which encompasses the consumed source speech along with the previously generated target text and codes:

$$\mathcal{C}_c = \left(\mathbf{X}_{1:g_c}, \mathbf{Y}_{<c}^{\text{text}}, \mathbf{Y}_{<c}^{\text{code}}\right). \tag{1}$$

The model is then trained to maximize the chunk-factorized log-likelihood of the joint target sequences given the trajectory:

$$\begin{aligned} &\log p\left(\mathbf{Y}^{\text{text}}, \mathbf{Y}^{\text{code}} \mid \mathbf{X}, \boldsymbol{\tau}\right) \\ &= \sum_{c=1}^{C} \log p\left(\mathbf{Y}_c^{\text{text}}, \mathbf{Y}_c^{\text{code}} \mid \mathcal{C}_c\right). \end{aligned} \tag{2}$$

3.2 Trajectory Construction

Given a paired-S2ST sample $(\mathbf{X}, \mathbf{Y}^{\text{text}}, \mathbf{Y}^{\text{wav}})$, our goal is to derive a chunk-level trajectory $\boldsymbol{\tau}$ that dictates which source prefix must be read before a corresponding target text and code chunk can be emitted. We construct $\boldsymbol{\tau}$ in the following two steps, as illustrated in Figure 1.

Step 1: Text and Audio Alignment. Given an S2TT sample $(\mathbf{X}, \mathbf{Y}^{\text{text}})$, we first force-align the source and target speech to obtain word-level boundaries, and apply SimAlign (Jalili Sabet et al., 2020) to extract cross-lingual word alignments $\hat{\mathbf{A}}$. If a target word y_i aligns to a source word $x_{a(i)}$, we identify the source-side end frame of $x_{a(i)}$, denoted as $t_{a(i)}$. This frame index represents the earliest source prefix containing sufficient acoustic information to commit y_i. Following trajectory-based simultaneous translation practices, we monotonize this boundary to ensure non-decreasing write positions even under local syntactic reordering:

$$\tilde{t}_i = \max(\tilde{t}_{i-1}, t_{a(i)}), \tag{3}$$

where $\tilde{t}_0 = 0$.

Step 2: Code Segmentation and Trajectory Grouping. Next, we tokenize the target audio into semantic-code sequences and segment these codes according to the target-word boundaries. Each target word and its corresponding code span

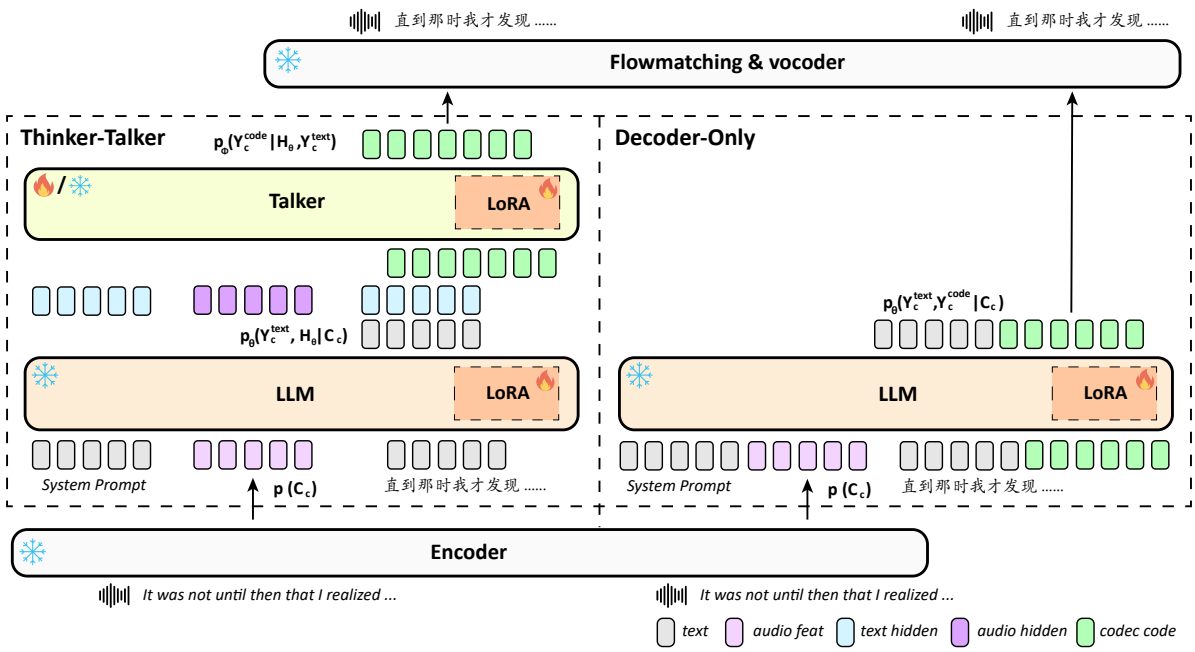

Figure 2: **Matched backbone comparison. Thinker–Talker** (left) and **Dec-only** (right) share an identical speech encoder, base LLM backbone (with independent LoRA adapters), semantic-code tokenizer $\mathcal{Q}$, and frozen flow-matching/vocoder backend.

inherit the same monotonized source boundary $\tilde{t}_i$. To form fixed-size streaming steps, we group adjacent target words and their codes whose boundaries fall within the same pre-defined source chunk intervals of *1 second*, yielding the discrete trajectory $\boldsymbol{\tau}$ defined previously. Within this sequence, g_c represents the required source frame boundary before emitting chunk c. Chunks without newly committed target content act as `read`/`wait` steps; chunks containing target content act as `write` steps that jointly emit $(\mathbf{Y}_c^{\text{text}}, \mathbf{Y}_c^{\text{code}})$. Equation (2) represents the resulting chunk-factorized training objective under this teacher trajectory. Notably, offline S2ST is simply the special case where $C = 1$ and $g_1 = |\mathbf{X}|$, whereas streaming generation corresponds to $C > 1$.

3.3 Backbone Choices

To isolate the effect of the target-code prediction path under a limited-data regime, we design a controlled comparison between a two-stream architecture and a unified decoder. As shown in Figure 2, both systems are matched in their speech encoder, tokenizer $\mathcal{Q}$, frozen acoustic backend, and training data. They differ exclusively in how they parameterize the chunk-factorized joint distribution $p(\mathbf{Y}_c^{\text{text}}, \mathbf{Y}_c^{\text{code}} \mid \mathcal{C}_c)$ from Equation (2).

Thinker–Talker. The two-stream approach factorizes target-text planning and target-code prediction. The joint probability of the chunk is decomposed as:

$$p\Big(\mathbf{Y}_c^{\text{text}}, \mathbf{Y}_c^{\text{code}} \mid \mathcal{C}_c\Big) = p_\theta\big(\mathbf{Y}_c^{\text{text}} \mid \mathcal{C}_c\big)\, p_\phi\Big(\mathbf{Y}_c^{\text{code}} \mid \mathcal{C}_c, \mathbf{Y}_c^{\text{text}}, \mathbf{H}_\theta\Big), \tag{4}$$

where $\mathbf{H}_\theta$ denotes the deterministic sequence of intermediate hidden states produced by the Thinker θ during text generation. The Thinker θ and Talker ϕ are initialized from Qwen2.5-Omni (Xu et al., 2025a), with the codec embedding and output head re-initialized for the DualCodec (Li et al., 2025) vocabulary. The Talker routes code prediction conditioning on the Thinker's hidden states alongside the aligned target-text token embeddings.

Dec-Only Baseline. The unified-decoder baseline unifies this process by appending $|\mathcal{V}_{\text{code}}| = 16,384$ code tokens to the Thinker's vocabulary, modeling the joint distribution with a single autoregressive head:

$$p_\theta\Big(\mathbf{Y}_c^{\text{text}}, \mathbf{Y}_c^{\text{code}} \mid \mathcal{C}_c\Big), \quad \text{where} \quad \mathcal{V} = \mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{code}}. \tag{5}$$

Within each chunk c, the model autoregressively generates the target text tokens followed immedi-

ately by the target code tokens (App. B.2).

4 Experiments

4.1 Experimental Protocol

Training Data. Our training recipe combines large-scale ASR, S2TT, MT, and TTS supervision with a limited paired-S2ST set (Table 2)—the core low-resource regime studied in this work. Both Thinker–Talker and Dec-only observe the exact same audio data pool. We filter candidate trajectories via length and difficulty quotas using the Normalized Inversion Rate (NIR) (Yu et al., 2025) and balance them. Paired-data statistics are in Appendix A; filtering details are in App. A.3 and A.4.

Matched Three-Stage Training. Both backbones share identical data mixtures, optimizers, and token budgets across three stages.*Stage 1 (Warmup)*: Thinker–Talker trains only the Talker on TTS data, while Dec-only uses a dual-branch LoRA (Hu et al., 2021) warmup to prevent modality underfitting, drawing inspiration from modality-specialized parameter partitioning (Xu et al., 2025c).*Stage 2 (Joint Pretraining)*: Both train on an ASR/S2TT/MT/TTS/S2ST mixture at a 0.2:1:0.5:1:1.5 ratio. *Stage 3 (Streaming Finetuning)*: We merge Stage-2 adapters, attach fresh LoRAs, freeze embeddings and prediction heads, and finetune on augmented streaming trajectories consisting of ASR/S2TT/MT/S2ST tasks. To support all inference latency tiers with a single checkpoint, training trajectories are augmented by sampling a latency multiplier $m \in \{1, \ldots, 12\}$, which merges m source chunks into a single read step. Overall, Dec-only consumes roughly $\mathbf{2}\times$ the GPU-hours of Thinker–Talker (Appendix Table 8).

Inference Configuration. Offline S2ST decodes full source utterances using greedy sampling, whereas streaming evaluation follows trajectory-defined latency tiers obtained by sweeping the latency multiplier m from a single checkpoint. Streaming decoding uses a rolling-window KV-cache and chunked flow-matching with cross-chunk audio context for speech. Following (Ouyang et al., 2025b), we use *num_beams=4* for the main streaming results and greedy decoding for comparison (full hyperparameters in Appendix C).

Evaluation Benchmarks and Metrics. We evaluate offline S2ST on CVSS-T (Jia et al., 2022). For streaming evaluation, we use RealSI (Cheng et al., 2024) for sentence-level/long-form S2TT/S2ST, and ACL60/60-dev (Salesky et al., 2023) for long-form En→Zh streaming S2TT. Crucially, all streaming test sets consist of real-world human recordings; synthesized speech is used strictly to construct the paired-S2ST training data and is never used as an evaluation target. Unless noted otherwise, all benchmarks cover both En→Zh and Zh→En. We report BLEU (Post, 2018) and ASR-BLEU for translation quality, SIM-O and AutoPCP (A.PCP) (Andrews et al., 2022; Cheng et al., 2025b) for speech quality, and Length-Adaptive Average Lagging (LAAL) (Papi et al., 2022) or Stream-LAAL (Papi et al., 2024) for streaming latency. Protocols are in Appendix D; metrics and baselines are in Appendix D.3. We additionally re-score with a second ASR system (Appendix G).

4.2 Offline S2ST Evaluation

We first evaluate the model's offline S2ST performance on the CVSS-T dataset. This reflects the model's basic speech translation and generation abilities aside from the added complexities of streaming, allowing for a clean comparison between the Thinker–Talker and Dec-only architectures. We evaluate them against state-of-the-art public baselines, including GPT-4o (OpenAI, 2024), Qwen-Omni2.5 (Xu et al., 2025a), Seamless-M4T (Communication et al., 2023a), and UniSS (Cheng et al., 2025b) (Table 1); see also App. E.1.

As shown in Table 1, the two-stream Thinker–Talker architecture substantially improves translation quality over the matched Dec-only baseline, successfully matching the performance of the previous state-of-the-art, UniSS(Q). In both directions, the Talker yields higher scores in both Text-BLEU and ASR-BLEU. This confirms that forcing a unified decoder to predict dense semantic codes introduces modality interference that harms intermediate text planning. Furthermore, on En→Zh, the Talker's +4.0 ASR-BLEU gain notably outpaces its +2.14 Text-BLEU gain. This outsized content-realization improvement indicates that the two-stream design provides a dual benefit: it protects text reasoning and establishes a much more faithful path for target-speech generation. However, the Dec-only baseline maintains a clear edge in non-text acoustic quality, scoring higher on A.PCP and SIM-O and even outperforming UniSS. This highlights a fundamental intelligence-versus-quality trade-off: the Dec-only model leverages

Table 1: **Offline S2ST on CVSS-T.** Bold marks the best result among S2ST baselines and our models. †Scores quoted from (Cheng et al., 2025b); ‡Re-evaluated under our unified protocol. – denotes undisclosed.

Category	Model	#	ASR-BLEU↑		Text-BLEU↑		A.PCP↑		SIM-O↑	
			En→Zh	Zh→En	En→Zh	Zh→En	En→Zh	Zh→En	En→Zh	Zh→En
MLLM	GPT-4o†	—	25.42	**31.64**	—	—	**2.66**	2.58	—	—
	Qwen-Omni2.5†	7B	8.04	22.66	34.85	24.39	1.90	1.92	—	—
	Step-Audio 2 mini	7B	**32.81**	25.35	—	—	—	—	—	—
S2ST	Seamless-m4t-v2-large‡	2.3B	20.86	22.25	20.75	22.60	2.43	2.67	0.26	0.11
	Seamless-expressive‡	1.7B	23.70	15.89	25.45	16.67	2.73	**2.76**	0.42	0.34
	UniSS (P)‡	1.5B	30.09	23.77	30.77	24.63	2.73	2.75	0.40	0.43
	UniSS (Q)‡	1.5B	**32.04**	24.72	32.95	25.51	2.71	2.75	0.40	0.42
Ours	Ours (Dec-only)	3B	27.12	23.41	31.59	24.90	**2.96**	2.75	**0.45**	**0.59**
	Ours (Thinker–Talker)	3B	31.12	**25.18**	**33.73**	**26.41**	2.70	2.64	0.35	0.45

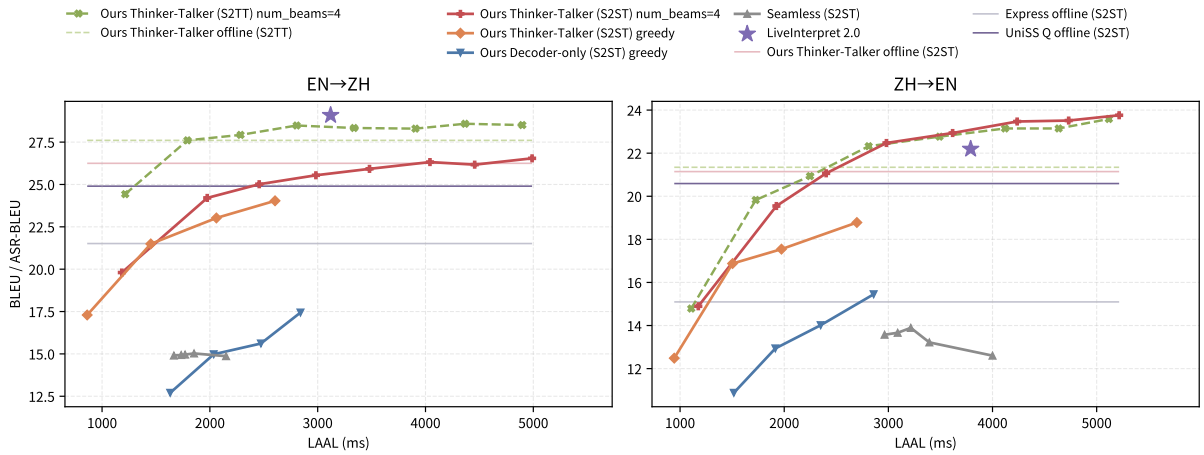

Figure 3: **RealSI sentence-level content trade-off.** Dashed curves denote S2TT text BLEU, solid curves denote S2ST ASR-BLEU, and horizontal lines mark offline reference lines.

Table 2: Training data statistics by task type. Token counts are in millions. Audio duration aggregates ASR, S2TT, Stage-1 TTS, and paired-S2ST audio.

Task	Tokens (M)	Audio duration (h)
TTS	4,531.3	89,595.8
MT	4,011.7	—
ASR	293.1	2,811.9
S2TT	2597.2	24,359.1
S2ST	307.4	2,104.8
total	11,740.7	118,871.6

its entire 3B backbone for code generation, which strengthens its ability to capture fine-grained audio features but severely harms linguistic intelligence. In contrast, the two-stream system restricts code generation to a lightweight 0.4B Talker, sacrificing some acoustic richness to protect overall translation fidelity prior to trajectory finetuning.

4.3 Sentence-Level Simultaneous Translation

Sentence-level simultaneous translation is a bounded setting for evaluating simultaneous S2TT or S2ST. In the resulting frontier plots, the horizontal axis represents the LAAL delay, while the vertical axis denotes translation quality, measured by text BLEU (for S2TT) or ASR-BLEU (for S2ST). A better-performing system yields curves positioned closer to the top-left region, achieving both lower latency and higher translation quality. We evaluate our system on RealSI (Cheng et al., 2024) against SeamlessStreaming (Communication et al., 2023b), LiveInterpret 2.0 (Cheng et al., 2025a)[1] , and offline models.

Figure 3 illustrates the sentence-level content

[1]Quality scores use audio obtained by using its official Volcengine API script (concatenated, non-chunked audio), scored with our unified pipeline. Latency uses the LAAL reported in its paper rather than an API-measured value, to avoid service-side queuing delay; refer to Appendix. F for LAAL_CA analysis.

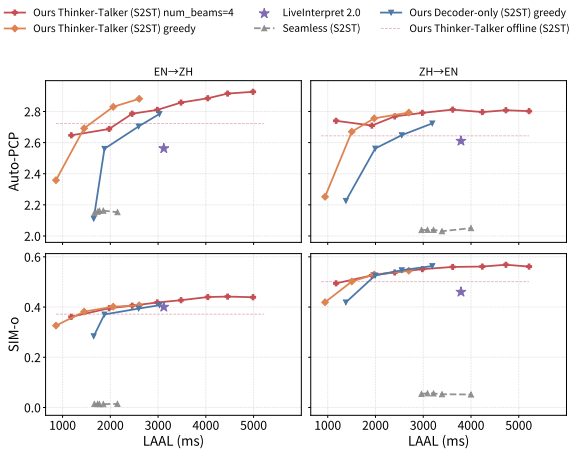

Figure 4: **RealSI sentence-level acoustic quality trade-off:** A.PCP and SIM-O against LAAL for En→Zh and Zh→En.

trade-off. The two-stream Talker vastly outperforms both Seamless streaming and the matched Dec-only baseline across all latency tiers in both directions. Crucially, this massive ASR-BLEU lead confirms that trajectory finetuning alone cannot resolve a unified decoder's internal text/code conflict, cementing the structural necessity of two-stream factorization for continuous streaming. This persists under CA/RTF analysis, where Thinker–Talker remains stronger and more efficient than Dec-only (App. F). Furthermore, under beam search, the Talker scales so effectively that it eventually surpasses its own offline S2ST reference at higher latency tiers, and from tier $m3$ onward its streaming operating points already exceed UniSS-Q (Cheng et al., 2025b), a strong open-source *offline* S2ST system, in both directions—despite UniSS-Q having access to the full utterance while our system decodes under a streaming constraint.

Figure 4 demonstrates that these massive translation gains do not compromise acoustic fidelity. Unlike in the offline setting, the streaming Talker consistently outperforms Dec-only on A.PCP. More importantly, as the latency multiplier m increases, the Talker's A.PCP scores actually exceed our offline S2ST baseline. This proves that chunked text-code generation via trajectory supervision actively improves local rhythm, rather than merely preserving it.

Finally, our system achieves highly competitive performance against state-of-the-art, closed-source S2ST systems. Across both directions, the Talker eclipses LiveInterpret 2.0 in speech quality at comparable latencies (Figure 4). In translation fidelity (Table 3), the Talker successfully matches or exceeds LiveInterpret 2.0 from tier $m4$ onward on Zh→En, and shrinks the gap to a highly competitive margin on En→Zh. To complement these automatic metrics with human judgments of naturalness, fidelity, and translation adequacy, we further report a human listening study and an LLM-as-judge (VIP) evaluation against LiveInterpret 2.0 and Seamless-Streaming in Appendix H, which corroborate the same latency-matched ranking.

4.4 Long-form Simultaneous Translation

Long-form streaming evaluates a model's ability to maintain discourse context and prevent low-latency error accumulation over unbounded, continuous source streams. We anchor this evaluation with two complementary comparisons: long-form streaming S2ST against LiveInterpret 2.0 on RealSI (Cheng et al., 2024), and streaming S2TT against published academic systems—such as CMU25 (Ouyang et al., 2025a), NAIST-25 (Tan et al., 2025), and the IWSLT baseline VAD—on ACL60/60-dev (Salesky et al., 2023)[2]. For S2TT, we report BLEU against StreamLAAL (Papi et al., 2024). We report ASR-BLEU for S2ST translation fidelity. (RealSI long-form S2TT trade-offs are reported in Appendix E.3.)

Table 3: **RealSI streaming S2ST ASR-BLEU.** Comparison against reported LiveInterpret 2.0 operating points. **Bold** marks the highest score per row. (The table is truncated to m2–m6 due to the page width limitation.)

		Ours					LiveInterpret
Type	**Direction**	**m2**	**m3**	**m4**	**m5**	**m6**	**ASR-BLEU**
Sent	En→Zh	24.21	25.01	25.54	25.92	26.32	**29.09**
	Zh→En	19.55	21.07	22.47	22.94	**23.47**	22.19
Doc	En→Zh	26.58	28.42	29.69	**30.04**	29.73	29.33
	Zh→En	22.40	22.62	24.37	26.53	25.60	**26.80**

As shown in Table 3 (bottom rows), the trajectory-finetuned Talker successfully scales to continuous S2ST without the catastrophic performance degradation typically associated with unbounded generation. On the RealSI long-form evaluation, our system remains highly competitive with LiveInterpret 2.0, a heavily optimized closed-source benchmark. Specifically, our higher latency tiers ($m4$–$m6$) successfully exceed the closed-source baseline on En→Zh and maintain

[2]CMU operating points are taken directly from the CMU paper; NAIST-25 and IWSLT VAD curves are digitized from the NAIST paper to serve as trend references.

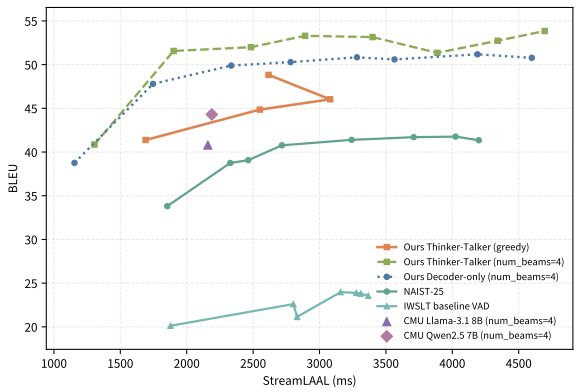

Figure 5: **ACL60/60-dev long-form En→Zh streaming S2TT trade-off.** BLEU against StreamLAAL.

a tight, approaching margin on Zh→En. This confirms that the joint text-code commitment path effectively manages long-form target speech generation.

Figure 5 illustrates long-form En→Zh streaming S2TT on ACL60/60-dev. Consistent with our sentence-level findings, the two-stream Talker strictly dominates the matched Dec-only baseline across all latency tiers. More importantly, the Talker substantially outperforms recent academic systems. Our greedy Talker configuration alone is already competitive with CMU's heavier 7B models operating under beam search at similar latencies (∼2.4 s StreamLAAL). Meanwhile, our standard beam-search configuration reaches 52.30 BLEU at matched tier m3, decisively outperforming all external baselines in the same latency region. This provides robust, cross-benchmark evidence that the trajectory-and-Talker recipe yields a state-of-the-art long-form quality–latency frontier.

4.5 Ablation Studies

Table 4: **Data quality and trajectory sampling strategy ablation on long-form RealSI S2TT.** Stream-side BLEU by latency tier.

Direction	Variant	m1	m2	m3	m4
En→Zh	RANDOM	4.59	24.86	26.22	26.54
	NIR	21.14	26.44	28.00	**28.53**
	NIR+LINEAR	**22.26**	**26.94**	**28.19**	28.33
	NIR+GAUSSIAN	22.08	26.91	27.60	27.90
Zh→En	RANDOM	3.56	14.55	17.32	17.08
	NIR	11.98	18.42	19.89	**21.73**
	NIR+LINEAR	**12.83**	**18.83**	**20.40**	21.63
	NIR+GAUSSIAN	11.95	17.78	19.87	21.46

Data quality and sampling strategy. Table 4 isolates the effects of trajectory data filtering and latency multiplier sampling on streaming translation quality, using a controlled subset of 60k trajectories. For data selection, RANDOM draws trajectories arbitrarily, whereas NIR applies our monotonicity-aware filter. Both use a uniform distribution for latency multipliers during training. To test the effect of multiplier sampling, we also evaluate NIR+LINEAR and NIR+GAUSSIAN, which apply shaped distributions to bias training toward specific latency tiers (see Appendix B).

Without monotonicity filtering, low-latency ($m1$) performance collapses entirely, plummeting to 4.59 and 3.56 BLEU in the two directions, respectively. Introducing NIR-based filtering immediately recovers this gap, proving that high-quality, difficulty-controlled trajectory supervision is the primary driver of low-latency robustness. Furthermore, varying the multiplier sampling schedules yields negligible differences compared with the uniform baseline. This confirms that curating a stable, filtered data pool is far more critical than the specific multiplier sampling schedule.

Table 5: RealSI sentence-level paired-S2ST budget ablation by latency tier. Values are ASR-BLEU. All results use greedy decoding.

Direction	Model	Paired-S2ST budget	m1	m2	m3	m4
En→Zh	Talker	Full	**17.30**	**21.49**	**23.02**	**24.04**
	Talker	10%	15.50	20.00	21.49	23.47
	Talker	10% w/o aux	9.47	18.46	21.12	21.90
	Dec-only	Full	12.70	**14.98**	**15.61**	**17.44**
	Dec-only	10%	**13.02**	14.07	15.38	16.52
	Dec-only	10% w/o aux	12.96	13.97	14.22	15.93
Zh→En	Talker	Full	**12.49**	**16.88**	**17.55**	**18.78**
	Talker	10%	9.76	15.57	17.53	18.59
	Talker	10% w/o aux	8.45	14.62	16.91	18.24
	Dec-only	Full	**10.88**	**12.94**	14.01	**15.45**
	Dec-only	10%	10.29	12.82	**14.16**	15.12
	Dec-only	10% w/o aux	10.24	11.68	13.83	15.12

Dataset size and augmented tasks Table 5 isolates the effect of reducing the paired-S2ST supervision to just 10%, with and without the fixed auxiliary mixture (simul ASR/S2TT/MT). All models are trained under the NIR+LINEAR trajectory sampling setting. The Talker exhibits remarkable data efficiency: when auxiliary data is retained, the 10% model stays highly competitive with the full-data baseline. However, removing this auxiliary data (`w/o aux`) triggers a severe degradation, dropping by nearly 8.0 ASR-BLEU at $m1$ on En→Zh. This demonstrates that the auxiliary mixture successfully anchors the model, allowing it to scale up and maintain streaming S2ST using only a fraction of paired speech. Conversely, the Dec-only base-

line remains strictly bounded below the Talker's performance ceiling across all subsets. Notably, removing auxiliary data from the 10% Dec-only model yields negligible differences. This indicates that the unified decoder is fundamentally bottlenecked by its internal text/code conflict and cannot effectively leverage auxiliary multitask supervision to improve streaming performance.

5 Conclusion

This work demonstrates that robust, long-form streaming speech-to-speech translation (S2ST) is achievable even under realistic low-resource conditions where paired S2ST audio is scarce. Through a controlled evaluation, we show that the two-stream Thinker–Talker factorization utilizes limited paired data significantly more effectively than unified-decoder baselines, yielding superior offline and simultaneous translation quality. By combining this architecture with a joint text-code trajectory supervision mechanism, our system successfully scales to long-form streaming with controllable latency tiers. Crucially, our ablations confirm that state-of-the-art low-latency performance requires the synergy of both reliable trajectory filtering and the structural decoupling of linguistic and acoustic generation. Future work will extend this framework to duplex architectures for real-time interactive conversation and scale paired data across scenarios.

Limitations

Our experiments are limited to Chinese–English because suitable multilingual streaming S2ST data are still scarce. Extending the same trajectory-based training pipeline to broader multilingual settings is an important direction for future work.

In particular, the paired streaming S2ST data used in this work are constructed through synthesis, alignment, and filtering, rather than collected from naturally recorded simultaneous interpretation. As a result, they may not fully capture all phenomena in real-world interpreting scenarios. Collecting real simultaneous-interpretation speech pairs for training and evaluation is an important direction for future work.

In addition, this work mainly studies the language-model and token-prediction backbone, while keeping the speech encoder and speech-generation backend fixed. The current chunk-wise Flow Matching decoder is not yet fully streaming-native, which may limit inter-chunk coherence and inference efficiency. Extending the acoustic backend like streaming Flow Matching or Multi token predictor to explicitly model streaming history is left to future work.

Ethical Considerations

S2ST systems that synthesize target speech raise voice-identity and consent concerns. Although this work does not introduce a new voice-cloning model, generated target speech could still be misused for impersonation if deployed without appropriate safeguards. Any release of models, data, or demos should therefore document synthesis sources, avoid unauthorized voice cloning, and make generated audio identifiable where appropriate.

Acknowledgments

This work is partially supported by NSFC Grant 62376237; the 2023 Shenzhen Stability Science Program; and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2023ZT10X044).

References

AI Challenger. 2017. Ai challenger 2017: English-chinese machine translation task. https://github.com/AIChallenger/AI_Challenger_2017. Official dataset and competition platform; accessed via the AI Challenger 2017 repository and platform documentation.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, and 27 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *Preprint*, arXiv:2406.02430.

Pierre Andrews, Guillaume Wenzek, Kevin Heffernan, Onur Çelebi, Anna Sun, Ammar Kamran, Yingzhe Guo, Alexandre Mourachko, Holger Schwenk, and Angela Fan. 2022. stopes - modular machine translation pipelines. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 258–265, Abu Dhabi, UAE. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 4218–4222, Marseille, France. European Language Resources Association.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Dekun Chen, Xueyao Zhang, Yuancheng Wang, Kenan Dai, Li Ma, and Zhizheng Wu. 2026. Flexivoice: Enabling flexible style control in zero-shot tts with natural language instructions. *arXiv preprint arXiv:2601.04656*.

Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, Yabin Li, Xiang Lv, Jiaqing Liu, Haoneng Luo, Bin Ma, Chongjia Ni, Xian Shi, Jialong Tang, Hui Wang, and 17 others. 2025. Minmo: A multimodal large language model for seamless voice interaction. *Preprint*, arXiv:2501.06282.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.*

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024a. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. *arXiv preprint arXiv:2410.06885*.

Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, Yu Qiao, and Jing Shao. 2024b. Octavius: Mitigating task interference in mllms via lora-moe. *Preprint*, arXiv:2311.02684.

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

Shanbo Cheng, Zhichao Huang, Tom Ko, Hang Li, Ningxin Peng, Lu Xu, and Qini Zhang. 2024. Towards achieving human parity on end-to-end simultaneous speech translation via llm agent. *arXiv preprint arXiv:2407.21646*.

Sitong Cheng, Weizhen Bian, Xinsheng Wang, Ruibin Yuan, Jianyi Chen, Shunshun Yin, Yike Guo, and Wei Xue. 2025b. Uniss: Unified expressive speech-to-speech translation with your voice. *arXiv preprint arXiv:2509.21144*.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, and 49 others. 2023a. SeamlessM4T: Massively Multilingual & Multimodal Machine Translation. *arXiv preprint*.

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

Keqi Deng, Wenxi Chen, Xie Chen, and Phil Woodland. 2025. Simuls2s-llm: Unlocking simultaneous inference of speech llms for speech-to-speech translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16718–16734.

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.

Amirbek Djanibekov, Luisa Bentivogli, Matteo Negri, and Sara Papi. 2026. Simulu: Training-free policy for long-form simultaneous speech-to-speech translation. *arXiv preprint arXiv:2603.16924*.

Qianqian Dong, Zhiying Huang, Qiao Tian, Chen Xu, Tom Ko, Yunlong Zhao, Siyuan Feng, Tang Li, Kexin Wang, Xuxin Cheng, Fengpeng Yue, Ye Bai, Xi Chen, Lu Lu, Zejun Ma, Yuping Wang, Mingxuan Wang, and Yuxuan Wang. 2023. Polyvoice: Language models for speech to speech translation. *Preprint*, arXiv:2306.02982.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jingren Zhou. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *Preprint*, arXiv:2412.10117.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. *Preprint*, arXiv:2406.18009.

Qingkai Fang, Shaolei Zhang, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. Can we achieve high-quality direct speech-to-speech translation without parallel speech data? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7264–7277.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. 2022. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*. ISCA.

Shoutao Guo, Xiang Li, Mengge Liu, Wei Chen, and Yang Feng. 2025. Streamuni: Achieving streaming speech translation with a unified large speech-language model. *arXiv preprint arXiv:2507.07803*.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *Preprint*, arXiv:2407.05361.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Hirofumi Inaguma, Sravya Popuri, Ilia Kulikov, Peng-Jen Chen, Changhan Wang, Yu-An Chung, Yun Tang, Ann Lee, Shinji Watanabe, and Juan Pino. 2023. UnitY: Two-pass direct speech-to-speech translation with discrete units. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15655–15680, Toronto, Canada. Association for Computational Linguistics.

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.

Ye Jia, Michelle Tadmor Ramanovich, Quan Wang, and Heiga Zen. 2022. CVSS corpus and massively multilingual speech-to-speech translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6691–6703, Marseille, France. European Language Resources Association.

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

Tom Labiausse, Laurent Mazaré, Edouard Grave, Patrick Pérez, Alexandre Défossez, and Neil Zeghidour. 2025. High-fidelity simultaneous speech-to-speech translation. *Preprint*, arXiv:2502.03382.

Chenyang Le, Bing Han, Jinshun Li, Songyong Chen, and Yanmin Qian. 2025. Simulmega: Moe routers are advanced policy makers for simultaneous speech translation. *arXiv preprint arXiv:2509.01200*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Preprint*, arXiv:2306.15687.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Yossi Adi, Juan Pino, Jiatao Gu, and Wei-Ning Hsu. 2022. Textless speech-to-speech translation on real data. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 860–872, Seattle, United States. Association for Computational Linguistics.

Jiaqi Li, Xiaolong Lin, Zhekai Li, Shixi Huang, Yuancheng Wang, Chaoren Wang, Zhenpeng Zhan, and Zhizheng Wu. 2025. Dualcodec: A low-frame-rate, semantically-enhanced neural audio codec for speech generation. *arXiv preprint arXiv:2505.13000*.

Zhengrui Ma, Yang Feng, and Min Zhang. 2024. Overcoming non-monotonicity in transducer-based streaming generation. *arXiv preprint arXiv:2411.17170*.

Eliya Nachmani, Alon Levkovitch, Yifan Ding, Chulayuth Asawaroengchai, Heiga Zen, and Michelle Tadmor Ramanovich. 2024. Translatotron 3: Speech to speech translation with monolingual data. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10686–10690. IEEE.

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

OpenAI. 2024. Gpt-4o.

Siqi Ouyang, Xi Xu, and Lei Li. 2025a. Cmu's iwslt 2025 simultaneous speech translation system. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 309–314.

Siqi Ouyang, Xi Xu, and Lei Li. 2025b. Infinisst: Simultaneous translation of unbounded speech with large language model. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3032–3046.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

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

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.

Peter Polák and Ondřej Bojar. 2024. Long-form end-to-end speech translation via latent alignment segmentation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 1076–1082. IEEE.

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.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

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

Jiatong Shi, Chunlei Zhang, Jinchuan Tian, Junrui Ni, Hao Zhang, Shinji Watanabe, and Dong Yu. 2025. Balancing speech understanding and generation using continual pre-training for codec-based speech llm. *arXiv preprint arXiv:2502.16897*.

Hubert Siuzdak. 2023. Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis. *arXiv preprint arXiv:2306.00814*.

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

Haotian Tan, Ruhiyah Faradishi Widiaputri, Jan Meyer Saragih, Yuka Ko, Katsuhito Sudoh, Satoshi Nakamura, and Sakriani Sakti. 2025. NAIST simultaneous speech translation system for IWSLT 2025. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 369–378, Vienna, Austria (in-person and online). Association for Computational Linguistics.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. SALMONN: towards generic hearing abilities for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. Covost 2 and massively multilingual speech translation. In *Interspeech*, pages 2247–2251.

Minghan Wang, Thuy Vu, Yuxia Wang, Ehsan Shareghi, and Gholamreza Haffari. 2025. Conversational simulmt: Efficient simultaneous translation with large language models. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 93–105.

Tianrui Wang, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei. 2023. Viola: Unified codec language models for speech recognition, synthesis, and translation. *Preprint*, arXiv:2305.16107.

Wenfu Wang, Chenxing Li, Liqiang Zhang, Yiyang Zhao, Yuxiang Zou, Hanzhao Li, Mingyu Cui, Hao Zhang, Kun Wei, Le Xu, Zikang Huang, Jiajun Xu, Jiliang Hu, Xiang He, Zeyu Xie, Jiawen Kang, Youjun Chen, Meng Yu, Dong Yu, and 7 others. 2026. Covo-audio technical report. *Preprint*, arXiv:2602.09823.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2024. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *arXiv preprint arXiv:2409.00750*.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, Mingrui Chen, Peng Liu, Wang You, Xiangyu Tony Zhang, Xingyuan Li, Xuerui Yang, Yayue Deng, Yechang Huang, Yuxin Li, and 90 others. 2025. Step-audio 2 technical report. *Preprint*, arXiv:2507.16632.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. 2024. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *Preprint*, arXiv:2410.13848.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and

Junyang Lin. 2025a. Qwen2.5-omni technical report. *Preprint*, arXiv:2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025b. Qwen3-omni technical report. *Preprint*, arXiv:2509.17765.

Zhiyang Xu, Minqian Liu, Ying Shen, Joy Rimchala, Jiaxin Zhang, Qifan Wang, Yu Cheng, and Lifu Huang. 2025c. Modality-specialized synergizers for interleaved vision-language generalists. In *International Conference on Learning Representations*, volume 2025, pages 13055–13080.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Rong Ye, Chengqi Zhao, Tom Ko, Chutong Meng, Tao Wang, Mingxuan Wang, and Jun Cao. 2023. Gigast: A 10,000-hour pseudo speech translation corpus. *Preprint*, arXiv:2204.03939.

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*.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *Preprint*, arXiv:2412.02612.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. 2022. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. *Preprint*, arXiv:2110.03370.

Shaolei Zhang, Qingkai Fang, Shoutao Guo, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. StreamSpeech: Simultaneous Speech-to-Speech Translation with Multi-task Learning. *arXiv preprint*.

Zhisheng Zheng, Xiaohang Sun, Tuan Dinh, Abhishek Yanamandra, Abhinav Jain, Zhu Liu, Sunil Hadap, Vimal Bhat, Manoj Aggarwal, Gerard Medioni, and David Harwath. 2026. Rosettaspeech: Zero-shot speech-to-speech translation without parallel speech. *Preprint*, arXiv:2511.20974.

Han Zhu, Lingxuan Ye, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhifeng Han, Weiji Zhuang, Long Lin, and Daniel Povey. 2026. Omnivoice: Towards omnilingual zero-shot text-to-speech with diffusion language models. *arXiv preprint arXiv:2604.00688*.

A Data Sources, Pair Construction, and Filtering

Table 6: Training data sources by task type. Sample counts are in thousands (k). Audio duration is aggregated over speech-containing splits. `short_audio_supplement`, `emilia_paired_*`, and *S2ST* rows are derived internal subsets or filtered paired pools built from the cited source corpora.

Dataset	Samples (k)	Hours
ASR		
common_voice_en	1143.3	1,808.2
common_voice_zh	29.6	42.6
librispeech	281.2	961.1
aishell_2	1,009	1000.0
S2TT		
gigast	7,665.2	9,807.1
mustc_en-zh	358.9	596.1
covost_en-zh	288.2	428.0
covost_zh-en	7.1	11.0
wenetspeech	14,203.7	9,732.0
emilia_en-zh	598.4	1,540.1
emilia_zh-en	875.9	2,244.8
MT		
ai_challenger	10,000.0	—
WMT	23,206.6	—
TTS		
emilia_en	15,533.1	40,714.4
emilia_zh	16,954.1	42,540.5
short_audio_supplement	2,036.0	1,083.2
TTS-Paired		
emilia_paired_en	999.0	2,704.3
emilia_paired_zh	999.0	2,553.4
S2ST		
s2st_en-zh	685.9	1,193.3
s2st_zh-en	484.4	911.4

A.1 Data Source Statistics

Our training data combine large publicly available supervision for speech understanding and text translation with a substantially smaller pool of paired-S2ST examples. The full multitask mixture draws from LibriSpeech (Panayotov et al., 2015), Common Voice (Ardila et al., 2020), MuST-C (Di Gangi et al., 2019), CoVoST 2 (Wang et al., 2021), GigaST (Ye et al., 2023), WenetSpeech (Zhang et al., 2022), Emilia (He et al., 2024), and large-scale text-parallel corpora such as AI Challenger (AI Challenger, 2017) and WMT (Bojar et al., 2017), together with several short-speech supplements derived from public ASR datasets. Their task roles are reported in the main text; this appendix focuses on paired-S2ST construction and the trajectory-specific filters that matter for the controlled comparisons.

Table 6 summarizes the multitask training mixture and the filtered paired-S2ST pools used for trajectory construction. Text-only MT corpora do not contribute audio hours and are therefore marked as not applicable.

A.2 Paired-S2ST Candidate Construction

Before entering the candidate pool, each source speech corpus is first cleaned at the transcript level: we cross-check two independent ASR engines (Qwen3-ASR-1.7B and Whisper-large-v3) against the corpus's reference transcript under strict word-error-rate and length thresholds, retaining roughly 80–95% of utterances depending on corpus quality. This keeps the source transcript—and hence the eventual translation target—accurate before any further construction.

The paired-S2ST pool is built to provide $(\mathbf{X}, \mathbf{Y}^{\text{text}}, \mathbf{Y}^{\text{wav}})$ supervision for the offline S2ST stage. In addition to directly available S2TT-style parallel corpora, we expand the candidate pool with publicly available speech corpora for which target text can be generated reliably. Concretely, we sample additional Chinese and English speech from publicly available speech resources. A strong text translation model, Qwen3.6 35BA3B (Yang et al., 2025), produces target-language text when parallel targets are not already available, yielding a pool of roughly 20,000 hours where every candidate has an accurate source transcript and target text. Each candidate then contains source speech $\mathbf{X}$ and target text $\mathbf{Y}^{\text{text}}$, after which target speech $\mathbf{Y}^{\text{wav}}$ is synthesized.

Target speech is synthesized with OmniVoice (Zhu et al., 2026) under the same cross-lingual voice-cloning setting used in our evaluation pipeline. This keeps the paired-S2ST supervision aligned with the downstream S2ST objective: the target side must be correct in content while remaining usable for semantic-code learning and speech reconstruction.

A.3 Monotonicity-Aware Trajectory Selection

For streaming training, each reliable full-utterance pair must be converted into incremental read–write supervision. To reduce unstable long-distance reorderings in the trajectory pool, we apply monotonicity-aware selection based on the normalized inversion rate (NIR) (Yu et al., 2025). Let the source positions aligned to target words form the sequence $\hat{\mathbf{A}} = [\hat{a}_1, \hat{a}_2, \ldots]$. If $I_{\hat{\mathbf{A}}}$ is the inversion

number of this sequence, NIR is defined as

$$\mathrm{NIR} = \frac{2I_{\hat{\mathbf{A}}}}{|\hat{\mathbf{A}}|(|\hat{\mathbf{A}}|-1)} \times 100\%. \tag{6}$$

Lower NIR indicates weaker reordering and therefore more stable read–write supervision. In the main trajectory pool, we bucket examples by NIR and sentence length. The main sampling policy uses difficulty weights $\{\text{high} : 0.1, \text{mid_high} : 0.3, \text{mid_low} : 0.4, \text{low} : 0.2\}$ and length weights $\{\text{short} : 0.1, \text{medium} : 0.5, \text{long} : 0.4\}$.

Figure 6 and Table 7 compare this NIR-stratified trajectory pool against a random-selection baseline drawn from overlapping SimAlign-processed candidates. Metrics are computed on pre-trajectory word alignments (`simalign_inter`). Random sampling concentrates mass near NIR=0 and higher Spearman ρ, reflecting predominantly monotonic alignments. NIR-stratified selection deliberately retains harder reordering buckets (10% high-NIR quota), so its pooled distribution spreads to larger NIR values even though the policy is monotonicity-aware. Dashed vertical lines in the figure mark the direction-wise means; summary moments appear in the table.

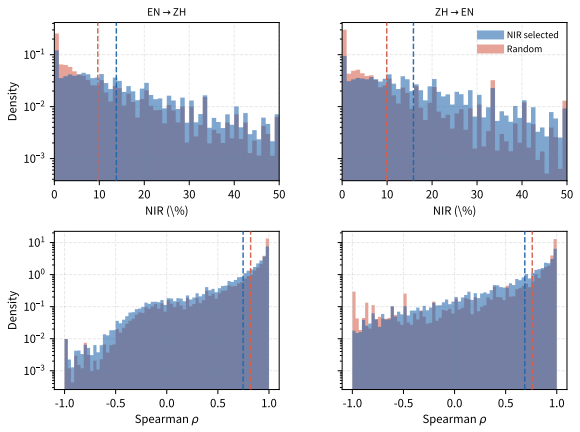

Figure 6: Pooled NIR and Spearman distributions for NIR-stratified vs. random trajectory candidates (log-scaled density). Top row: NIR (%); bottom row: Spearman ρ. Columns: En→Zh (left) and Zh→En (right). NIR histograms use NIR $\leq$ 50%; dashed lines are means.

A.4 ASR-Based Pair Filtering

Before paired examples enter S2ST training, we filter synthesized targets with ASR-based content checks. For the trajectory-ready pool, the En→Zh direction contains 1,470,930 candidates (1,589.62 hours) and the Zh→En direction contains 1,206,014 candidates (816.06 hours). We apply language-specific error-rate thresholds of 5% for Chinese and 7% for English. After filtering, 78.76% of En→Zh candidates and 80.19% of Zh→En candidates are retained. This step removes content errors and duration mismatches before semantic-code supervision is created. We keep these thresholds deliberately above zero rather than requiring near-perfect ASR match: voice-cloned synthesis legitimately carries source-accent coloring and disfluencies (e.g., filled pauses), and an overly strict threshold would trade this prosodic diversity for ASR-"clean" but flat speech.

Table 7: Direction-wise mean and variance of NIR and Spearman ρ for NIR-stratified vs. random selection. NIR moments use NIR $\leq$ 50%; Spearman uses all retained samples.

Selection	**Dir.**	**NIR (%)**		**Spearman ρ**	
		μ	σ^2	μ	σ^2
NIR stratified	En→Zh	13.79	146.64	0.747	0.103
NIR stratified	Zh→En	15.86	160.69	0.687	0.147
Random	En→Zh	9.66	131.46	0.821	0.081
Random	Zh→En	9.92	147.47	0.760	0.176

B Training Implementation Details and Hyperparameters

Table 8 summarizes the stage-level training budgets and lists the key hyperparameters for each stage. All runs use eight A800 GPUs. For each run, we select the checkpoint with the lowest held-out evaluation loss on per-dataset validation splits sampled at 0.5%, capped at 1,000 examples each. For both models, the speech encoder and Thinker are initialized from Qwen2.5-Omni (Xu et al., 2025a), and the same semantic-code pipeline (Li et al., 2025), flow-matching acoustic model (Chen et al., 2026; Le et al., 2023), and vocoder are adopted.

B.1 Thinker–Talker

Stage 1: Talker warmup. We initialize the Talker from the Qwen2.5-Omni talker module (see Appendix E.2 for the impact of different initialization methods). We then replace the Talker embedding and output head with the DualCodec vocabulary (16,384) and add four special tokens: `codec_bos`, `codec_eos`, `codec_pad`, and `codec_mask`. We unfreeze the Talker while freezing the Thinker, training the Talker using only TTS data for 2 epochs.

Table 8: Key hyperparameters for the main Thinker–Talker line and matched Dec-only baseline. "Tok/GPU" denotes tokens per GPU; "GPU Time" reports A800 GPU-hours for the corresponding stage.

Line	Stage	Trainable scope	Tok/GPU	GPU Time	LR schedule	Key setting
Thinker–Talker	Stage 1	– / Full	10k	416	$4 \times 10^{-4} \rightarrow 4 \times 10^{-5}$	2 epoch TTS
	Stage 2	LoRA r32 / Full	6.4k	136	$1 \times 10^{-4} \rightarrow 2 \times 10^{-5}$	multitask mixture
	Stage 3	LoRA r8 / LoRA r8	5.2k	176	$3 \times 10^{-5} \rightarrow 2 \times 10^{-6}$	1 epoch trajectory data
Dec-only	Stage 1	LoRA r32/r32	8k / 10k	1,248	$1 \times 10^{-4} \rightarrow 1 \times 10^{-5}$	1 epoch S2T&MT/ 2 epoch TTS
	Stage 2	LoRA r64	6.4k	112	$1 \times 10^{-4} \rightarrow 2 \times 10^{-5}$	same multitask mixture
	Stage 3	LoRA r8	5.2k	136	$3 \times 10^{-5} \rightarrow 2 \times 10^{-6}$	1 epoch trajectory data

Stage 2: multitask joint training. We then switch to joint multitask training on ASR, S2TT, MT, TTS, and S2ST, with a sampling ratio of ASR:S2TT:MT:TTS:S2ST = 0.2:1:0.5:1:1.5. To mitigate the scarcity of ASR data, we randomly replace the S2TT target with the ASR task with a probability of 10%. In this stage, we add LoRA modules to the Thinker and jointly train the Thinker's LoRA and the Talker. This aligns the Talker's hidden space while strengthening the Thinker's intelligence. The training target is set to 30k steps.

Stage 3: streaming trajectory finetuning. We merge the previous adapters into the base weights and instantiate fresh low-rank adapters for both the Thinker and Talker. We freeze the embeddings and prediction heads, and continue training on quality-filtered multi-task trajectories (including simultaneous ASR, S2TT, MT, and S2ST) for 1 epoch. In the default main-line setting, latency multipliers are sampled using probabilities that linearly decay from 1 to $M_{\max}$, with $M_{\max}$=12.

B.2 Dec-only

We first attempted to train the unified decoder with exactly the same data mixture and optimization recipe used by the Talker Stage 2 setup. In practice, this recipe did not allow the Dec-only model to reach comparable performance, due to inherent representational conflicts between multi-modal understanding and generation tasks (Shi et al., 2025; Wu et al., 2024). Increasing the proportion of understanding-oriented data improved translation and recognition behavior but weakened speech generation, whereas increasing the proportion of generation-oriented data had the opposite effect. To resolve this optimization conflict under fixed data and compute budgets, we introduced a dual-branch warmup stage—separating understanding-heavy and generation-heavy adaptations before performing parameter fusion, drawing inspiration from separate modality-specialized adapter architectures (Xu et al., 2025c).

Stage 1: dual-branch LoRA warmup. We initialize two different sets of LoRA parameters for the two branches. The understanding branch emphasizes MT, S2TT, ASR, and related tasks, which account for roughly 70% of its training exposure, and is trained for 1 epoch on all data. The generation branch emphasizes TTS, S2ST, and related generation tasks, which account for roughly 90% of its exposure, and is trained for 2 epochs, exhausting the TTS data.

Stage 2: multitask joint training. We then merge the understanding and generation branches and continue with full multitask training. The weight-fusion procedure is described in Section B.2. After fusion, Stage 2 uses the same multitask configuration as the Talker Stage 2 (Section B.1).

Stage 3: streaming trajectory finetuning. For streaming finetuning, we follow the same Stage 3 trajectory recipe used by the Talker line (Section B.1). We first merge the previous-stage adapters into the base model, then instantiate fresh low-rank adapters and freeze the embedding and prediction-head parameters.

Latency multiplier sampling. During streaming finetuning, each trajectory is additionally viewed under a sampled latency multiplier $m \in \{1, \ldots, M_{\max}\}$ that controls how aggressively source chunks are merged before emission. The RANDOM and NIR ablations in Table 4 both draw m uniformly, i.e., $P(m) = 1/M_{\max}$, and differ only in whether the trajectory pool is randomly selected or NIR-stratified. The NIR+LINEAR variant keeps the same NIR-stratified pool but replaces uniform sampling with a truncated linear schedule that front-loads low-latency supervision:

$$\tilde{w}_m = \max(a - b(m-1), 0)$$

and $P(m) = \tilde{w}_m / \sum_{k=1}^{M_{\max}} \tilde{w}_k$. We use an intercept of a=0.18, a slope of b=0.017, and $M_{\max}$=12. The NIR+GAUSSIAN variant also keeps the same NIR-stratified pool, but samples multipliers from a discretized Gaussian weighting centered on lower-latency tiers. We set the Gaussian mean to 3.5 and the variance to 4.0, then normalize the resulting weights over $m \in \{1, \ldots, M_{\max}\}$. The main-line Stage 3 recipe also uses uniform multiplier sampling over $\{1, \ldots, M_{\max}\}$.

Initialization fairness. Both lines are initialized from the same Qwen2.5-Omni family of publicly released weights: Dec-only inherits the Qwen2.5-Omni LLM body, and Thinker–Talker additionally inherits the Qwen2.5-Omni talker decoder body. In both lines, the codec embedding and output head—the only parameters that interact directly with DualCodec vocabulary—are freshly initialized. Initializing the Talker from Qwen serves as a modest optimization aid rather than the primary source of the performance gain (ablations on the initialization method are provided in Appendix E.2). Furthermore, in terms of compute, the Dec-only Stage 1 dual-branch LoRA warmup uses substantially more GPU-hours (Table 8), making the overall comparison conservative with respect to our system.

Weighted fusion of Thinker LoRA. Some Dec-only runs use branch-focused LoRA adapters that are fused before continued multitask or trajectory finetuning. Given N adapters $\{(A_i, B_i)\}_{i=1}^{N}$ and normalized weights $\{w_i\}_{i=1}^{N}$, we construct a single merged adapter $(\bar{A}, \bar{B})$ that exactly preserves the weighted sum of the individual updates, i.e., $\bar{B}\bar{A} = \sum_{i=1}^{N} w_i B_i A_i$. This is achieved by block-concatenating the scaled factors:

$$\begin{aligned} \bar{A} &= \begin{bmatrix} \sqrt{|w_1|}\, A_1 \\ \vdots \\ \sqrt{|w_N|}\, A_N \end{bmatrix}, \\ \bar{B} &= \begin{bmatrix} C_1 & \cdots & C_N \end{bmatrix}, \\ C_i &= \operatorname{sgn}(w_i)\sqrt{|w_i|}\, B_i, \qquad i = 1, \ldots, N. \end{aligned} \tag{7}$$

The newly merged adapter can then be finetuned further under the standard schedule.

C Inference and Decoding

The offline S2ST task takes full source audio utterances as input and outputs the full target audio. For both architectures, we set `repetition_penalty=1.2` for text decoding and `repetition_penalty=1.4` for code decoding.

Streaming evaluation uses finite source context and latency tiers induced by trajectory multipliers. Following InfiniSST (Ouyang et al., 2025b), we preserve system prompts and keep a bounded history window across turns. InfiniSST implements this strategy with a sinking-cache mechanism built on attention sinks that retain pre-RoPE key-value states. In contrast, we directly preserve the dynamic key-value cache from the previous turn. Once the history window limit is reached, we retain the latest $\lfloor 16/m \rfloor$ history turns and perform a full forward pass to obtain a new KV cache, where m is the latency multiplier.

For both architectures, streaming text decoding uses `no_repeat_ngram_size=5` and `repetition_penalty=1.2`. Beam search is additionally applied where explicitly mentioned. Semantic-code decoding uses greedy decoding with `repetition_penalty=1.4` and `no_repeat_ngram_size=5`, following the repetition-suppression strategy used in InfiniSST. These decoding settings are kept consistent between the Talker and Dec-only models.

Directly decoding a mel spectrogram from a single chunk with the flow-matching model and vocoder introduces audible discontinuities across chunk boundaries and produces unstable prosody and rhythm in long-form audio. To reduce these boundary artifacts, we preserve the latest 4 s of source audio and the latest 2 s of generated audio as prompt context for the flow-matching model. This improves timbre consistency and smooths transitions across chunks.

D Evaluation Protocols

D.1 Offline TTS Evaluation

Evaluation datasets. **Seed-TTS** Supplementary zero-shot TTS evaluation uses English and Chinese test sets following the Seed-TTS protocol (Anastassiou et al., 2024). These diagnostics are used only to assess speech-generation quality apart from the main S2ST benchmarks.

Evaluation metrics.

- **WER** measures content consistency between synthesized speech and the target text. English recognition uses Whisper-Large-V3 (Radford et al., 2022), while Chinese recognition uses Paraformer (Gao et al., 2022).

- **SIM-O** measures speaker similarity using WavLM-Large (Chen et al., 2022) speaker embeddings and cosine similarity against the reference target speech.

D.2 Offline Speech Understanding Evaluation

Evaluation datasets. LibriSpeech (English ASR). We report English speech understanding on LibriSpeech *test-clean* (Panayotov et al., 2015).

WenetSpeech (Mandarin ASR). We report Mandarin speech understanding on the WenetSpeech TestNet split (Zhang et al., 2022).

CoVoST 2 (offline S2TT). We report offline speech-to-text translation on the En→Zh and Zh→En directions of CoVoST 2 (Wang et al., 2021).

Evaluation metrics.

- **WER / CER** are used for English and Mandarin ASR respectively. English ASR uses the LibriSpeech evaluation protocol, and Mandarin ASR uses the WenetSpeech evaluation protocol.
- **BLEU** is used for offline S2TT on CoVoST 2 under the same text normalization and SacreBLEU (Post, 2018) setup described below for translation quality metrics.

D.3 Offline and Streaming S2ST Evaluation Details

Evaluation datasets. CVSS-T (offline S2ST). We use the Chinese (ZH) and English (EN) test sets from CVSS-T (Jia et al., 2022), which contain 4,897 utterance pairs (8.2 hours for Chinese and 6.3 hours for English). Both En→Zh and Zh→En directions are evaluated.

RealSI (streaming S2TT and S2ST). RealSI (Cheng et al., 2025a) is a long-form Chinese–English simultaneous interpretation benchmark comprising natural, casual speech across ten domains. It provides sentence-level and long-form evaluation settings. The sentence-level setting contains 51:12 minutes / 431 segments for En→Zh and 44:17 minutes / 347 segments for Zh→En.

ACL60/60-dev (long-form S2TT). We additionally evaluate long-form En→Zh streaming S2TT on ACL60/60-dev (Salesky et al., 2023), which consists of 5 real-world ACL 2022 oral-presentation talks in English, containing roughly one hour of audio in total under the IWSLT realistic-condition setting.

Evaluation inputs and references. Each evaluation example contains a reference translation text, a reference speech waveform used for speech-side comparison, and the model-generated target speech. Text-based metrics compare model text or ASR transcripts against the target reference. Speech similarity metrics compare the generated waveform against the paired reference waveform provided by the evaluation manifest.

Translation quality metrics. Unless otherwise stated, our offline S2ST comparison follows the UniSS evaluation protocol so that CVSS-T results remain directly comparable with recent offline S2ST systems.

- **Text-BLEU** evaluates generated translation text with *corpus-level* SacreBLEU (Post, 2018) following UniSS preprocessing. English text is lowercased and stripped of punctuation while preserving apostrophes. Chinese text is converted to simplified Chinese, punctuation is removed, and characters are separated by spaces. We use SacreBLEU's `zh` tokenizer for Chinese and `13a` for English.
- **ASR-BLEU** evaluates whether the synthesized speech preserves translation content. Generated speech is transcribed with Whisper-Large-V3 (Radford et al., 2022) for English and Paraformer-zh (Gao et al., 2022) for Chinese, then scored against the target reference under the same normalization used for Text-BLEU.

Speech similarity metrics.

- **SIM-O** is computed with WavLM-Large (Chen et al., 2022) speaker embeddings and cosine similarity against the paired real reference target speech.
- **A.PCP** (AutoPCP) follows the automatic prosodic-consistency protocol used in expressive multilingual S2ST evaluation. We use the Stopes (Andrews et al., 2022) implementation on paired 16 kHz waveforms with the released **AutoPCP-multilingual-v2** comparator on hidden states from layer 9 of a frozen **Wav2Vec2-Large-XLSR-53** encoder to compare the output against the real source audio.

Streaming metrics. Streaming experiments use SimulEval-style latency metrics together with task-specific quality metrics.

- **Sentence-level S2TT/S2ST** uses LAAL (Papi et al., 2022) and computation-aware LAAL (LAAL_CA). Sentence-level S2TT quality is reported with SimulEval text BLEU; sentence-level S2ST quality is reported with ASR-BLEU, A.PCP, and SIM-O from the generated speech.
- **Long-form S2TT** uses StreamLAAL (Papi et al., 2024) and StreamLAAL_CA. The main trade-off plots use the stream-side BLEU emitted by the StreamLAAL evaluator so that quality and latency come from the same long-form run.
- **Long-form S2ST** supplementary results report ASR-BLEU on the generated speech using corpus-level SacreBLEU.
- **RTF** is reported for sentence-level S2ST systems with instance logs, using active compute time divided by source-speech duration.

E Supplementary Evaluation Results

E.1 Offline Speech Understanding and Translation

Table 9 compares public spoken-language models with our Dec-only baseline and Talker main line on offline speech understanding and speech-to-text translation. We report English ASR on LibriSpeech *test-clean*, Mandarin ASR on WenetSpeech TestNet, and CoVoST 2 En→Zh / Zh→En BLEU under the preprocessing described in Section D. Published baseline numbers are taken from the original papers when available; the public comparison set includes SALMONN (Tang et al., 2024), GLM-4-Voice-Base (Zeng et al., 2024), MinMo (Chen et al., 2025), Qwen-Omni2.5 (Xu et al., 2025a), Step-Audio 2 (Wu et al., 2025), Qwen3-Omni (Xu et al., 2025b), and Covo-Audio (Wang et al., 2026). This table isolates understanding and translation quality from speech-generation behavior, which is evaluated separately on SEED in the next subsection.

Overall, the two architectures remain in a similar offline regime relative to public SLMs, suggesting that introducing an independent Talker does not materially degrade the shared thinker's speech understanding or translation ability. The observed differences are modest and task-dependent rather than systematic, so we treat offline understanding as preserved and focus on architecture-specific speech-generation behavior on SEED below.

Table 9: Offline speech understanding and speech-to-text translation. ASR columns report WER/CER (lower is better); S2TT columns report BLEU (higher is better). **Bold** marks the best result in each column.

		ASR		S2TT	
Model	**#**	En	Zh	En→Zh	Zh→En
SALMONN	7B	2.10	—	33.10	—
GLM-4-Voice-Base	9B	2.82	—	—	—
Minmo	7B	1.70	6.80	46.70	26.00
Qwen-Omni2.5	7B	1.80	5.90	41.40	29.40
Step-Audio 2	30B	**1.17**	**4.67**	49.01	**29.51**
Qwen3-Omni	30A3	1.22	4.69	48.72	21.50
Covo-Audio	7B	1.45	7.23	**49.84**	26.77
Ours (Dec-only)	3B	2.10	7.43	45.70	24.65
Ours (Talker)	3B	2.78	6.79	45.09	26.68

E.2 Zero-Shot Speech Generation on SEED

We evaluate the zero-shot speech-generation ability of our models on the Seed-TTS benchmark. All TTS results use the in-context-learning (ICL) protocol on SEED, focusing on content consistency (WER) and speaker similarity (SIM-O). Table 10 compares public models and our two structures. The public TTS and speech-language baselines include MaskGCT (Wang et al., 2024), E2 TTS (Eskimez et al., 2024), F5-TTS (Chen et al., 2024a), CosyVoice 2 (Du et al., 2024), GLM-4-Voice-Base (Zeng et al., 2024), MinMo (Chen et al., 2025), and Qwen-Omni2.5 (Xu et al., 2025a).

Table 10: Zero-shot speech generation on SEED. WER↓ and SIM-O↑ are reported for the English and Chinese subsets; **bold** marks the best result in each column. Missing entries are not reported in the original paper.

	SEED EN		SEED ZH	
Model	WER↓	SIM-O↑	WER↓	SIM-O↑
MaskGCT	2.62	0.714	2.27	**0.774**
E2 TTS	2.19	0.710	1.97	0.730
F5-TTS	**1.83**	0.647	**1.56**	0.741
CosyVoice 2	2.57	0.652	1.45	0.748
GLM-4-Voice-Base 9B	2.91	—	2.10	—
Minmo 7B	2.90	—	2.48	—
Qwen-Omni2.5 7B	2.72	**0.747**	1.70	0.752
Ours (Dec-only)	2.73	0.665	2.34	0.741
Ours (Talker)	1.93	0.652	1.87	0.723

The SEED comparison shows the same structural pattern as the offline and streaming S2ST results: separating code prediction from the unified decoder mainly improves content consistency, while speaker similarity remains in a comparable range. This serves as supplementary evidence that the Talker primarily helps the model say the correct

content more reliably.

Initialization control. To test whether initialization materially affects TTS performance, we run a matched TTS-only control on Emilia and evaluate on SeedTTS after one epoch. We compare a Talker initialized from the reused Qwen talker body, a Talker initialized from scratch, all with the Thinker frozen, and a Dec-only baseline with a TTS LoRA branch. Table 11 shows that initialization has only a minor effect within the Talker architecture: Qwen initialization provides a small advantage, but the from-scratch Talker remains close and still outperforms the Dec-only TTS branch. We therefore treat Qwen initialization as a modest optimization aid rather than the main source of the gain, which is better explained by the architectural separation itself than by training duration or initialization alone.

Table 11: SeedTTS TTS-only initialization control. Lower WER is better; higher SIM-O is better. Both rows use one-epoch TTS-only warmup on Emilia.

	SEED EN		SEED ZH	
Model	WER↓	SIM-O↑	WER↓	SIM-O↑
Talker (Qwen-initialized)	2.03	0.647	1.80	0.723
Talker (From-scratch)	2.41	0.643	1.85	0.723
Dec-only (TTS-LoRA)	3.98	0.657	3.22	0.738

E.3 RealSI Long-form S2TT Trade-off

The main text anchors our long-form performance claims using RealSI long-form S2ST against LiveInterpret 2.0 (Table 3) and ACL60/60 against academic systems. For completeness, Figure 7 reports the RealSI long-form streaming S2TT trade-off, with Seamless streaming as an additional reference. Consistent with our S2ST findings, the Thinker–Talker model forms a strong quality–latency frontier above the external baseline. On both En→Zh and Zh→En, our beam-search Talker (NB4) substantially outperforms Seamless streaming, maintaining leads of roughly 10 to 13 BLEU across all comparable latency tiers.

Crucially, this evaluation reaffirms the structural advantage of the two-stream Talker over the unified decoder for long-form continuous generation. Across both directions, the Talker strictly dominates the matched Dec-only baseline. This architectural advantage is most pronounced on the challenging Zh→En direction at ultra-low latencies, where the unified Dec-only model suffers a catastrophic collapse (plummeting to roughly 7.5 BLEU), whereas the Talker maintains robust translation fidelity. Furthermore, while greedy decoding provides a fast, competitive baseline at low latencies, transitioning to beam search (NB4) yields a massive and necessary quality boost to sustain optimal long-form generation as the read window relaxes.

F Computation-Aware Simultaneous Evaluation

This subsection examines whether our main latency–quality conclusions remain stable after accounting for active inference cost. We reuse the same RealSI runs as in the main paper and evaluate them with three complementary views: (i) LAAL_CA / StreamLAAL_CA, which augment standard latency metrics with computation-aware delay; and (ii) active compute RTF, defined as the total non-idle inference time divided by source duration. These metrics are intended to separate listener-side latency from pure wall-clock computation.

F.1 Sentence-level S2ST LAAL_CA.

Figure 8 replots the sentence-level S2ST frontier, shifting the horizontal axis to computation-aware latency (LAAL_CA). Introducing computation awareness adds an average delay of approximately 2 seconds across all tiers compared to the ideal, non-CA latency. Notably, the latency gap on the x-axis between our beam-search (NB4) and greedy Talker configurations is remarkably small. This indicates that the frozen flow-matching acoustic backend consumes the vast majority of the wall-clock processing time, making the translation quality gains of beam search highly worthwhile.

Furthermore, the DEC-ONLY structure exhibits a severe latency penalty, shifting significantly to the right. Predicting dense acoustic codes directly from a single 3B-parameter backbone creates a major computational bottleneck, severely inflating its delay compared with the lightweight 0.4B Talker. Finally, since LiveInterpret 2.0's own paper does not report a computation-aware latency, we re-implement its official API request as a SimulEval agent and measure its LAAL_CA under the same wall-clock instrumentation as our own systems. Under this measurement, accounting for computation awareness shifts LiveInterpret 2.0's own operating point considerably further right as well, into a latency regime comparable to our $m4$–$m6$ tiers rather

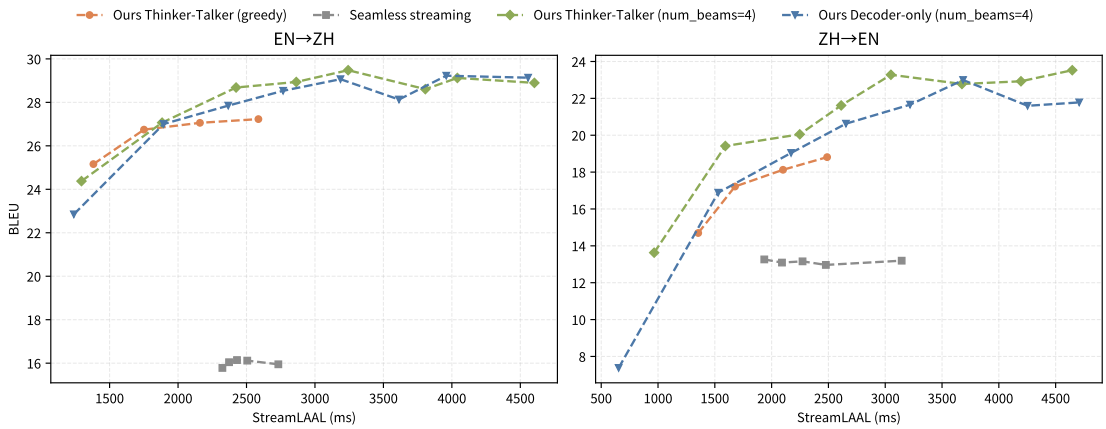

Figure 7: RealSI long-form streaming S2TT trade-off for both directions.

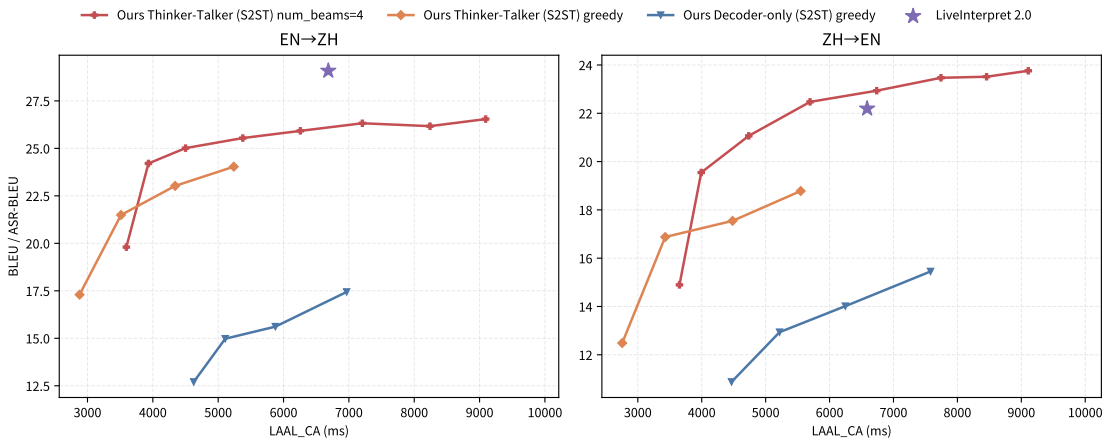

Figure 8: RealSI Sentence-level simultaneous S2ST under LAAL_CA. ASR-BLEU is plotted against computation-aware latency on RealSI sentence test sets.

than the low-latency end of the frontier. Within this comparable regime, our system is competitive: on Zh→En it matches or slightly exceeds LiveInterpret 2.0 at an equal or lower computation-aware latency, while on En→Zh a moderate quality gap of roughly 3 ASR-BLEU points remains at comparable latency.

F.2 Sentence-level S2ST Compute RTF

Figure 9 reports active compute RTF against LAAL_CA. RTF is computed as $\mathrm{RTF} = \sum_i (e_i - \max(d_i, e_{i-1}))/T_{\text{src}}$, where d_i and e_i are the per-chunk input-ready and completion times. Unlike LAAL_CA, which measures listener delay, this isolates how much active wall-clock compute is required per second of source speech.

Two major trends emerge that align with our previous architectural conclusions. First, the DEC-ONLY structure's RTF is substantially higher than the Talker's at every comparable tier, confirming that the unified decoder is too heavy for efficient continuous audio generation. Second, decoding strategy heavily influences compute at the extremes: the greedy Talker substantially reduces RTF at low latencies compared with NB4 (e.g., 1.54 vs. 1.86 RTF at En→Zh $m1$). However, as the latency tier increases, the RTF of greedy and NB4 rapidly converge. This reveals that the computational overhead of beam search is primarily driven by the high frequency of chunk-level decisions required at ultra-low latencies. Once chunks are larger (from $m4$ onward), the system becomes usable in real scenarios because the duration of generated audio can cover the compute overhead (RTF <1).

F.3 RealSI Long-form StreamLAAL_CA

Figure 10 demonstrates that long-form performance rankings remain stable when accounting for active computation time (StreamLAAL_CA). Across both directions, the Thinker–Talker model

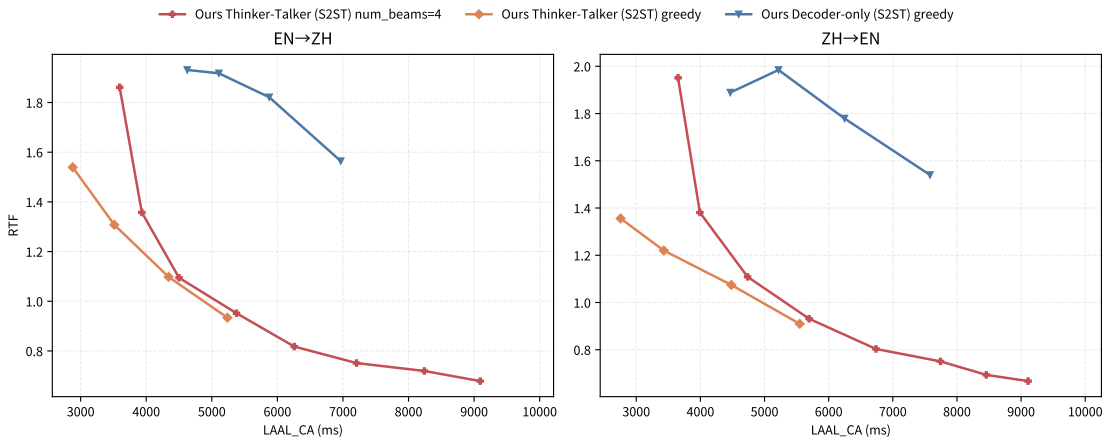

Figure 9: RealSI sentence-level simultaneous S2ST: active compute RTF vs. LAAL_CA. RTF counts only non-idle inference time per source second.

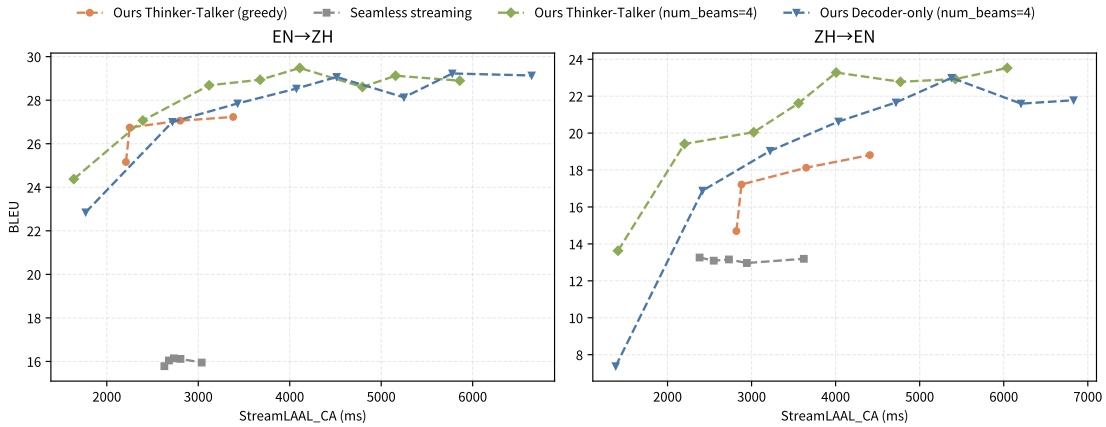

Figure 10: RealSI long-form simultaneous S2TT under StreamLAAL_CA.

(NB4) maintains a large quality advantage over Seamless streaming at every comparable tier. Crucially, the structural advantage of the two-stream architecture persists in long-form generation: the Talker (NB4) consistently yields higher translation fidelity at lower computation-aware latencies than the unified Dec-only baseline. This confirms that the Talker's superiority is a fundamental architectural benefit, not an artifact of a specific latency definition.

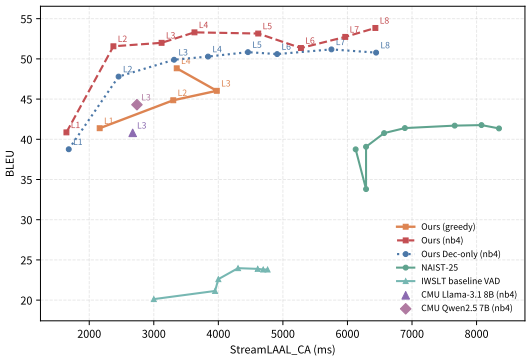

Figure 11: ACL60/60-dev Long-form simultaneous S2TT under StreamLAAL_CA.

F.4 ACL60/60-dev Long-form StreamLAAL_CA

Figure 11 plots StreamLAAL_CA on ACL60/60-dev, confirming that our system retains its state-of-the-art status even under strict computation-aware evaluation. The Talker (NB4) continues to strictly dominate the Dec-only baseline and vastly outperforms external systems such as NAIST-25 and the IWSLT baseline. Most notably, our computationally inexpensive greedy Talker configuration achieves a quality–latency trade-off highly competitive with CMU's heavier 7B and 8B models operating under beam search. This proves that our trajectory-and-Talker recipe provides a highly compute-efficient, state-of-the-art frontier for un-

bounded streaming S2TT.

G Reproducibility and ASR Sensitivity

G.1 Training-Seed Reproducibility

Our main results report single training runs. Rerunning the full-budget main setting across multiple seeds is prohibitively expensive at our training scale, so we instead pick the cheaper 10% w/o aux paired-S2ST budget ablation (the setting used in Table 5) as a representative proxy: to confirm the reported Talker–Dec-only gaps are not an artifact of random seed noise, we rerun this ablation for both models, in both directions, across 3 random seeds.

Table 12: Training-seed reproducibility of the 10% w/o aux paired-S2ST budget ablation. Values are ASR-BLEU, mean ± std over 3 training seeds.

Direction	Model	m1	m2	m3	m4
En→Zh	Talker	9.04±0.54	18.67±0.19	20.90±0.20	21.99±0.16
	Dec-only	12.83±0.12	14.16±0.20	14.96±0.64	16.40±0.42
Zh→En	Talker	8.88±0.38	14.89±0.25	17.31±0.41	18.41±0.48
	Dec-only	10.34±0.68	12.34±0.62	13.90±0.22	15.24±0.18

Table 12 shows that the seed std is small (≤0.68 BLEU) across all tiers, and these 3-seed averages agree closely with the single-run numbers already reported in Table 5, confirming that our training is stable rather than driven by random seed noise. The Talker–Dec-only gaps at $m2$–$m4$ (up to ∼4.5 BLEU) far exceed this variance, so they reflect genuine architectural differences rather than statistical fluctuation.

G.2 ASR Backend Sensitivity

ASR-BLEU transcribes synthesized target speech with an ASR system and scores the transcript against the reference translation, so the metric can inherit systematic biases of the chosen ASR backend, particularly when the generated speech carries accents, prosodic variation, or synthesis artifacts. The main paper reports ASR-BLEU with Whisper-large-v3 (English) and Paraformer-zh (Chinese), following Seed-TTS-eval (Anastassiou et al., 2024) and recent S2ST work (Cheng et al., 2025b; Le et al., 2025). To probe this potential bias, we retranscribe the identical generated audio—both ours and LiveInterpret 2.0's—with another strong ASR system, Qwen3-ASR-1.7B, and recompute ASR-BLEU under an otherwise identical protocol, at the same $m2$–$m6$ operating points as Table 3. Because LiveInterpret 2.0 is only accessible through its official API, we query it 3 times and report both LiveInterpret columns below as the mean ± std over these runs, transcribed with Qwen3-ASR-1.7B and with the original Whisper-large-v3 / Paraformer-zh backend, respectively.

Table 13: **RealSI streaming S2ST ASR-BLEU under an alternative ASR backend** (Qwen3-ASR-1.7B), compared against the main-paper backend (Whisper-large-v3 / Paraformer-zh) in Table 3. **Bold** marks the highest score per row.

		Ours (Qwen3-ASR)					LiveInterpret	
Type	Direction	m2	m3	m4	m5	m6	Qwen3-ASR	Whisper (orig.)
Sent	En→Zh	25.47	26.40	26.70	27.30	27.35	**29.38±0.36**	28.70±0.37
	Zh→En	19.84	20.93	22.28	23.06	**23.26**	22.34±0.20	22.22±0.14
Doc	En→Zh	28.35	30.30	31.01	**31.43**	31.13	30.59±0.71	29.99±0.72
	Zh→En	22.25	22.56	24.06	26.55	25.51	**28.03±0.30**	26.32±0.42

As shown in Table 13, the relative ranking is largely stable across the two backends: in 3 of 4 settings the comparison is unchanged (LiveInterpret ahead on Sent En→Zh, ours ahead on Sent Zh→En and Doc En→Zh). The only backend-sensitive case is long-form Zh→En, where our score barely moves while LiveInterpret's rises by roughly 1.7 BLEU (26.32 → 28.03), widening its lead in that single setting. Overall, the ASR backend shifts absolute scores somewhat without changing the paper's main conclusions.

H Human Evaluation and LLM-as-Judge Assessment

Automatic metrics such as ASR-BLEU can be biased by the ASR backend used for transcription. We therefore complement our automatic evaluation with a human listening study and an LLM-as-judge adequacy protocol (VIP), both on RealSI sentence-level samples against LiveInterpret 2.0 and Seamless-Streaming.

H.1 Human Listening Study

Protocol. Five evaluators recruited from among the paper authors and other members of the authors' laboratory rate a sampled subset of RealSI sentence-level examples (N=15 per direction). Participation was voluntary and uncompensated, and no external crowdsourcing platform was used. All evaluators were informed that their ratings would be used for research and publication and voluntarily consented to participate; only aggregated results are reported. Evaluators rate each example along three dimensions on a 1–5 scale, with Overall defined as the mean of the three:

- **Translation Adequacy:** how completely and correctly the spoken translation conveys the meaning of the source, judged by listening to the speech output.

- **Naturalness:** whether the delivery keeps a rhythm close to the source audio and to a real human simultaneous interpreter, i.e., whether pacing and rhythm track the source speaker and whether intonation is expressive rather than robotic or monotone.

- **Fidelity (audio quality):** whether the audio is clean and clear, free of noise, distortion, glitches, muffling, or artifacts.

For each sample, raters are given the source audio, its source transcript, and the generated target speech under test. Systems are anonymized and shuffled per sample (Figure 12). For a latency-matched comparison, we evaluate our **m5** tier and Seamless-Streaming's **thr = 0.9** tier, whose LAAL is comparable to LiveInterpret 2.0's reported latency; we additionally report our lower-latency **m2** tier to characterize the quality–latency trade-off as latency tightens.

Results. As shown in Table 14, at matched latency (m5), our system's Naturalness and Fidelity are top-tier in both directions, on par with or above LiveInterpret 2.0 (Overall 4.15 vs. 4.12 on En→Zh, 4.64 vs. 4.01 on Zh→En). At the lower-latency m2 tier scores drop, mainly due to cross-chunk discontinuities and a tighter delivery that feels less natural, though both tiers stay well above Seamless-Streaming. On Translation Adequacy, our m5 tier rates highest in both directions; since our raters are not professional interpreters, these scores may carry some bias, motivating the VIP LLM-as-judge protocol in Appendix H.2 as a complementary, less rater-dependent measure.

Statistical significance and agreement. Table 15 reports paired Wilcoxon signed-rank tests comparing the Overall score of Ours (m5) against each external system per direction. Even at N=15, the rankings are not noise: on par with LiveInterpret 2.0 on En→Zh (n.s.) and significantly better on Zh→En (p=0.003), and significantly above Seamless-Streaming in both directions ($p \leq 0.001$, surviving Holm correction). Inter-annotator agreement is fair at the single-rater level (Krippendorff's α=0.43 Overall, highest on Adequacy at 0.60) and substantial for the aggregated 5-rater mean we actually compare (ICC(2,k)=0.76 Overall; 0.77/0.72 for En→Zh/Zh→En), indicating the ranking above reflects genuine quality differences rather than rater noise.

Due to time and resource constraints, we conduct only a small-scale human evaluation (N=15 per direction, ∼75 min per rater for the full 4-system session). Larger-scale evaluation with a broader and more diverse evaluator pool is left to future work.

H.2 VIP: LLM-as-Judge Adequacy Evaluation

To assess translation quality at scale, we adopt **VIP**, an adequacy protocol originally proposed in the CLASI paper (Cheng et al., 2024) for professional interpreters; we implement it with an LLM judge (DeepSeek-V4-Pro) over the ASR-transcribed output, rating Key Information Recognition, Correctness Assessment, and Expressiveness Assessment, and report the resulting pass rate (higher is better).

As shown in Table 16, VIP follows the same trend as our objective metrics and human MOS: quality rises steadily as latency loosens. On En→Zh, a gap to LiveInterpret 2.0 remains (0.361 vs. 0.436 at m5), matching the ASR-BLEU trend in Table 3. On Zh→En, we reach parity from m5 (0.633 vs. 0.646) and surpass LiveInterpret 2.0 at higher tiers (up to 0.731 at m8).

I Case Study

We complement the aggregate curves above with utterance-level visualizations of two RealSI sentence-level examples under two latency settings (Figures 13, 14, 15, and 16).

Each figure aligns the source waveform (top) with the generated target waveform (bottom) on a shared time axis. Source words are aligned and grouped into m-second read windows (e.g., one second per step at m=1). Target segments denote partial translation commits. Pink *WAIT* intervals indicate steps where the agent receives additional source audio but emits no speech, successfully accumulating context before the next write.

At the finest-grained regime (m=1), the agent reads one source second per step, resulting in frequent but short target commits. As shown in the En→Zh example, the system correctly predicts *WAIT* steps early in the utterance, reflecting uncertainty at clausal boundaries during a disfluent

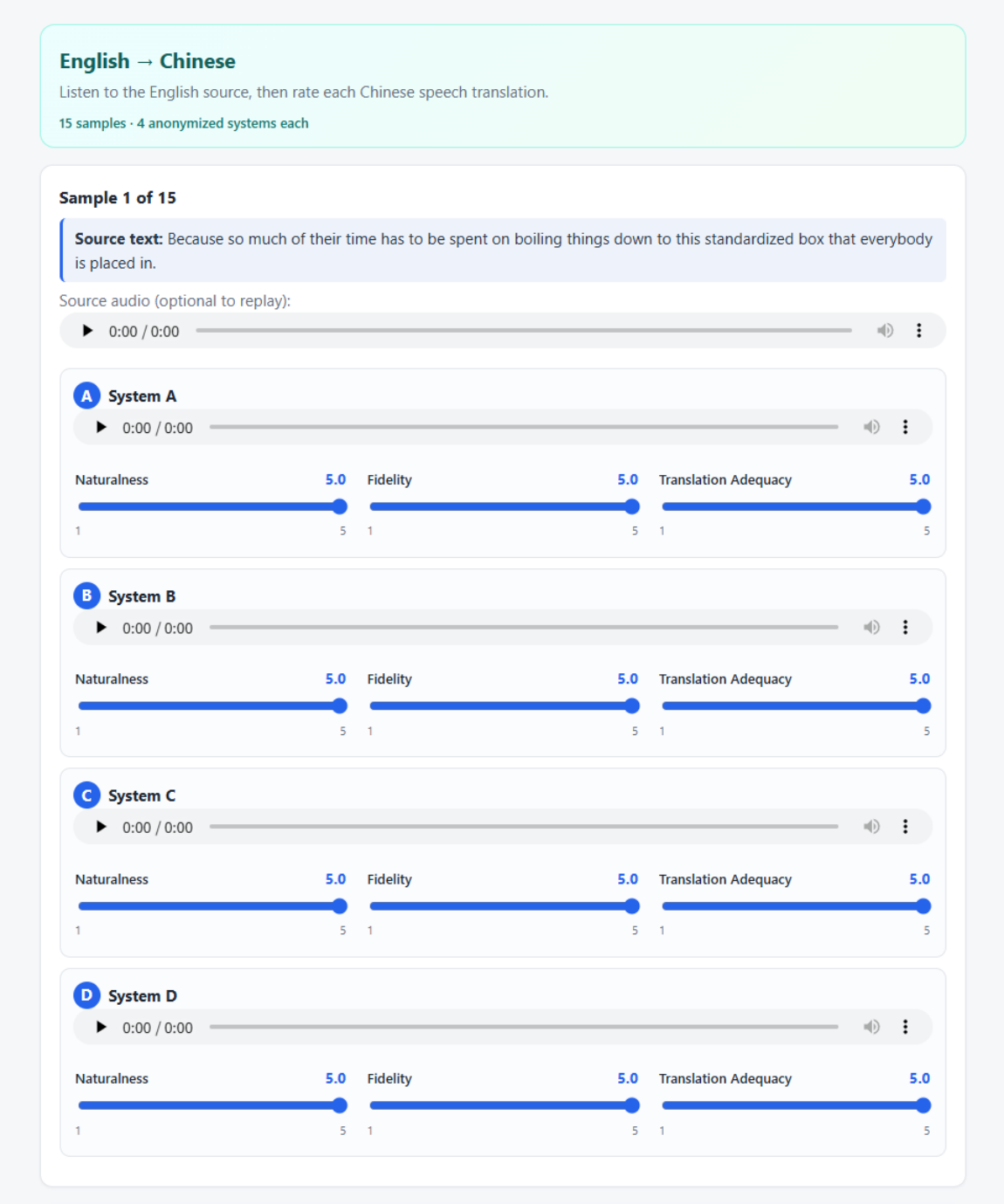

Figure 12: Anonymized rating interface used in the human listening study. Systems are randomly assigned to slots A–D and re-shuffled per sample so that raters cannot infer system identity; raters score Naturalness, Fidelity, and Translation Adequacy for each slot on a 1–5 slider.

Table 14: Human MOS (1–5, higher is better) on RealSI sentence-level samples (N=15 per direction). ± denotes the sample standard deviation over the 15 rated items. Ours (m5) is latency-matched to LiveInterpret 2.0; Ours (m2) is a lower-latency additional operating point.

Direction	System	Naturalness	Fidelity	Adequacy	Overall
En→Zh	Ours (m5)	4.04±0.37	4.24±0.32	4.18±0.46	4.15±0.32
	Ours (m2)	3.66±0.41	3.84±0.33	3.95±0.44	3.81±0.32
	LiveInterpret	4.22±0.27	4.18±0.35	3.96±0.74	4.12±0.37
	Seamless-Streaming	3.02±0.25	3.27±0.39	2.76±0.67	3.02±0.37
Zh→En	Ours (m5)	4.53±0.44	4.63±0.36	4.75±0.27	4.64±0.33
	Ours (m2)	3.84±0.38	4.06±0.31	4.31±0.54	4.07±0.31
	LiveInterpret	4.12±0.46	3.90±0.34	4.01±1.17	4.01±0.50
	Seamless-Streaming	3.95±0.29	4.04±0.30	4.04±0.56	4.01±0.34

Table 15: Paired Wilcoxon signed-rank test on the Overall human MOS, comparing Ours (m5) against each external system per direction.

Direction	Comparison	Ours (m5)	Other	p
En→Zh	vs. LiveInterpret	4.15	4.12	0.93 (n.s.)
En→Zh	vs. Seamless-Streaming	4.15	3.02	0.001
Zh→En	vs. LiveInterpret	4.64	4.01	0.003
Zh→En	vs. Seamless-Streaming	4.64	4.01	0.001

source input (filled pauses and false starts). Raising the multiplier to m=2 changes the rhythm as expected: read windows widen, the number of target commits drops, and individual writes carry more context per step. This demonstrates the intended latency–quality trade-off of our framework.

J Bad Cases and Failure Analysis

We identify three recurring failure modes encountered while constructing the paired streaming data and while decoding at tight latency. None of them affects the operating points reported in the main paper; rather, they characterize the current limits of our pipeline and motivate directions for future work.

J.1 Boundary Concatenation Artifacts

Figure 17 shows a representative silence→speech join in a causally re-timed target. The artifact surfaces as a short transient click or spectral discontinuity at chunk boundaries, most audible at the utterance onset; it is primarily perceptual and only marginally raises ASR WER. It arises because voiced segments are hard-cut at systematically tight forced-alignment (FA) word boundaries and spliced against exact numerical silence, so the resulting join is an abrupt energy step rather than a natural onset.

J.2 Forced-Alignment Error under Transcript–Audio Mismatch

Figure 18 shows a clip whose clean reference transcript omits a spoken false start (*"That I I… "*), aligned once against the clean text and once against the ASR transcript. When aligned on the clean text, the aligner collapses the first source word *"I"* into an 0.08 s sliver at the very onset and misattributes the false start to the following word; because this FA end-time becomes the cross-lingual `ready_time`, the first target word is unlocked at 0.08 s and emitted almost immediately, before any real source content has arrived. This is a data issue rather than a modeling failure: when the transcript omits spoken material, FA must absorb it into neighboring words, and the scheduler then faithfully trusts these incorrect boundaries. We mitigate this by re-transcribing with ASR and filtering mismatched samples upstream.

J.3 Premature Emission / Wrong Write

Figure 19 decodes the same Zh→En utterance at a tight (m=2) and a looser (m=4) latency tier. At m=2, the agent commits *"It's called a podium."* at 6.0 s and then stalls in a WAIT; at m=4, it correctly emits *"It's called the sage on the stage."* At the tighter tier, the agent writes after hearing only the leading noun *jiangtan* ("lectern"), before the disambiguating *shengzhe* ("sage") arrives, and cannot repair already-committed audio. This reflects the inherent read–write latency trade-off rather than a modeling defect.

K Reproducibility and Versioning

All controlled comparisons share the same semantic-code tokenizer, acoustic backend, and evaluation scripts unless explicitly marked otherwise. The main manuscript reports the task mix-

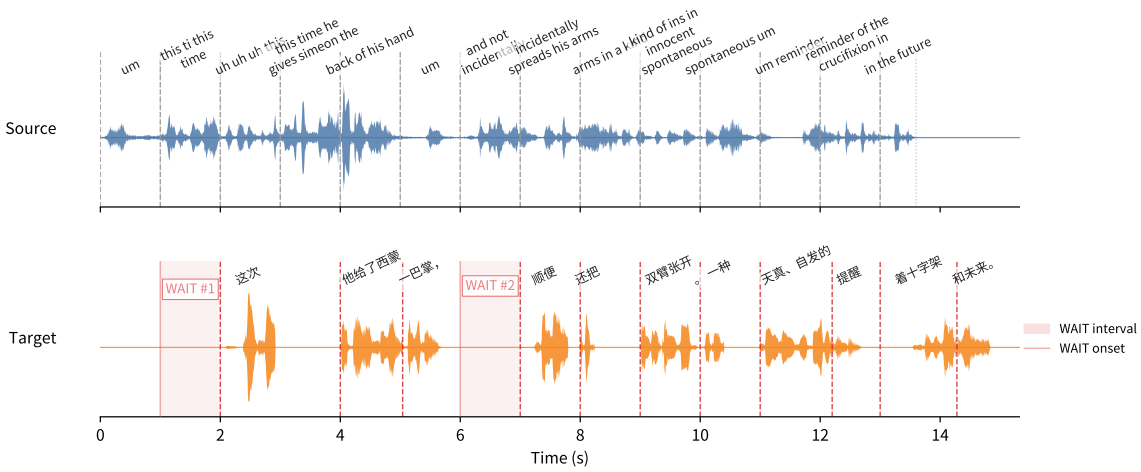

Figure 13: RealSI S2ST case study: En→Zh, m=1.

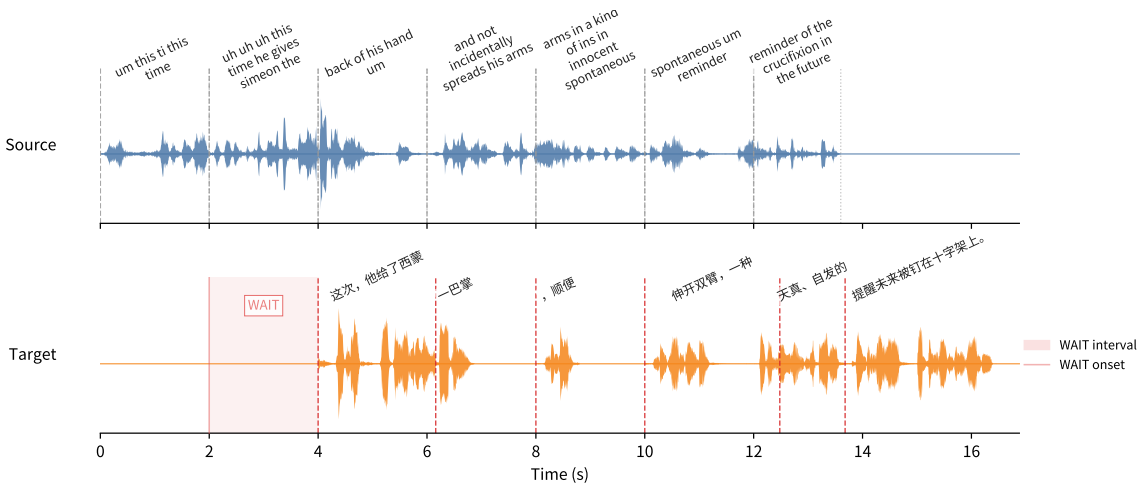

Figure 14: RealSI S2ST case study: En→Zh, m=2.

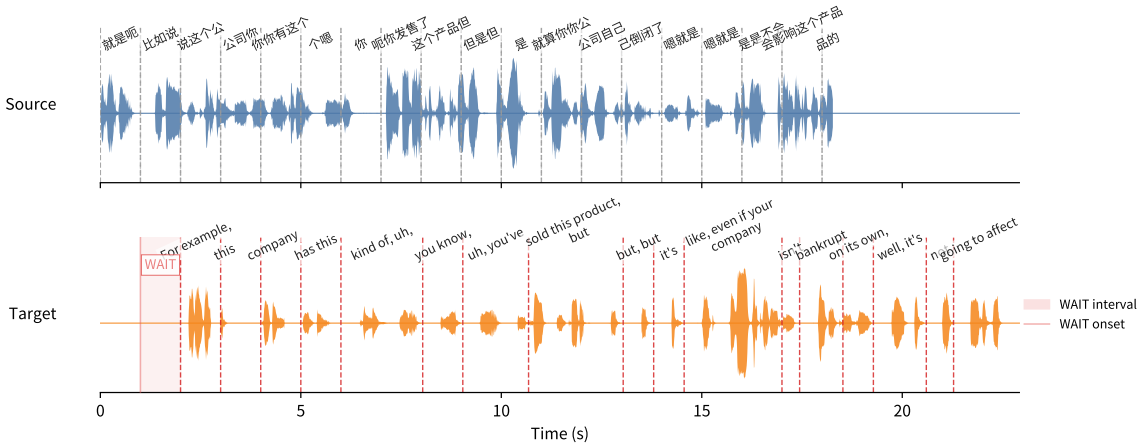

Figure 15: RealSI S2ST case study: Zh→En, m=1.

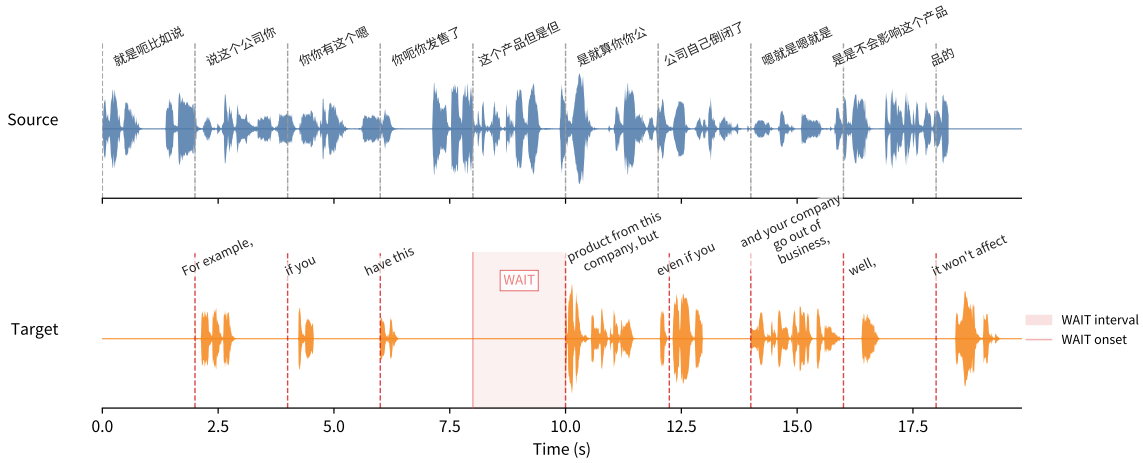

Figure 16: RealSI S2ST case study: Zh→En, m=2.

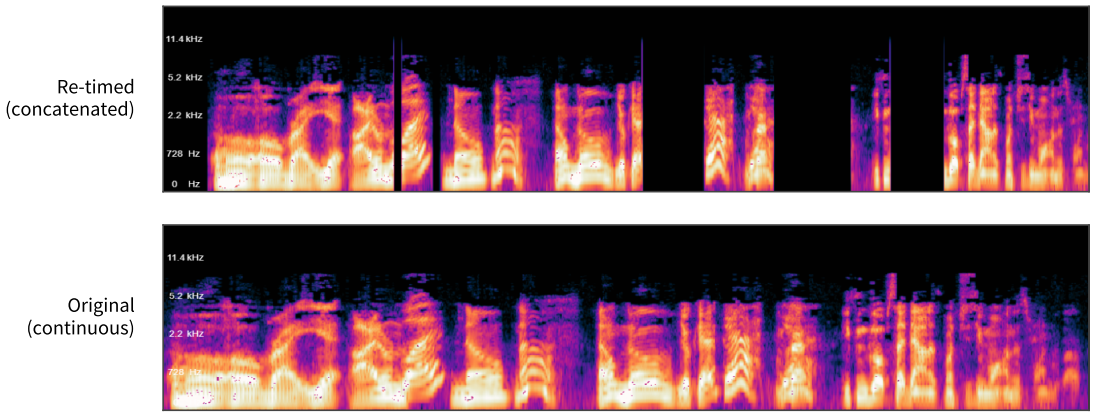

Figure 17: Boundary concatenation artifact (mel spectrogram). *Top:* the causally re-timed target, showing hard cuts and silent gaps at chunk boundaries. *Bottom:* the continuous original recording of the same content, with smooth onsets.

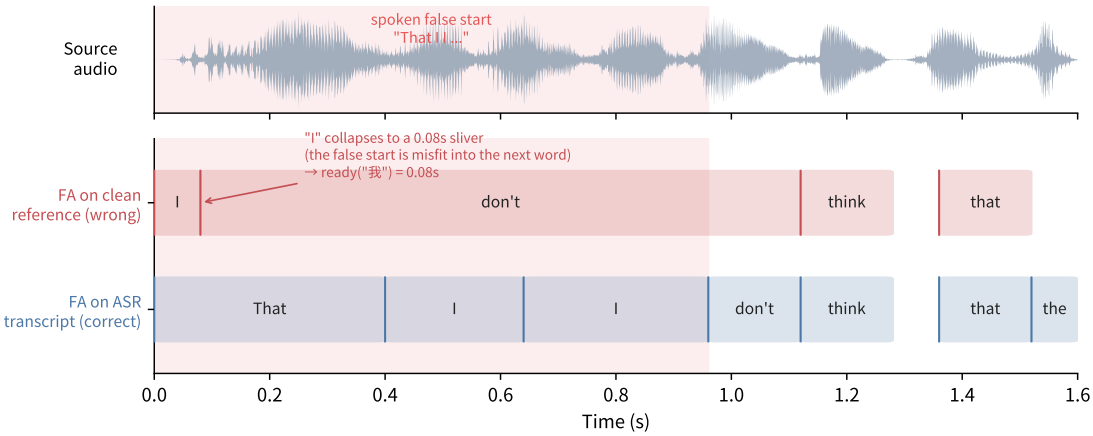

Figure 18: FA under transcript–audio mismatch (source clip; the false start *"That I I"* is shaded). Aligning on the clean reference (top) collapses *"I"* to an 0.08 s sliver, so the target `ready_time` fires at 0.08 s; aligning on the ASR transcript (bottom) matches the audio, where the real content only begins after ∼0.96 s.

Table 16: VIP pass rate (LLM-as-judge adequacy protocol; higher is better) across latency tiers $m1$–$m8$. LiveInterpret is the mean $\pm$ std over 3 official-API runs.

Direction	m1	m2	m3	m4	m5	m6	m7	m8	LiveInterpret
En→Zh	0.095	0.211	0.304	0.289	0.361	0.335	0.347	0.405	0.436±0.022
Zh→En	0.318	0.476	0.582	0.613	0.633	0.689	0.719	0.731	0.646±0.012

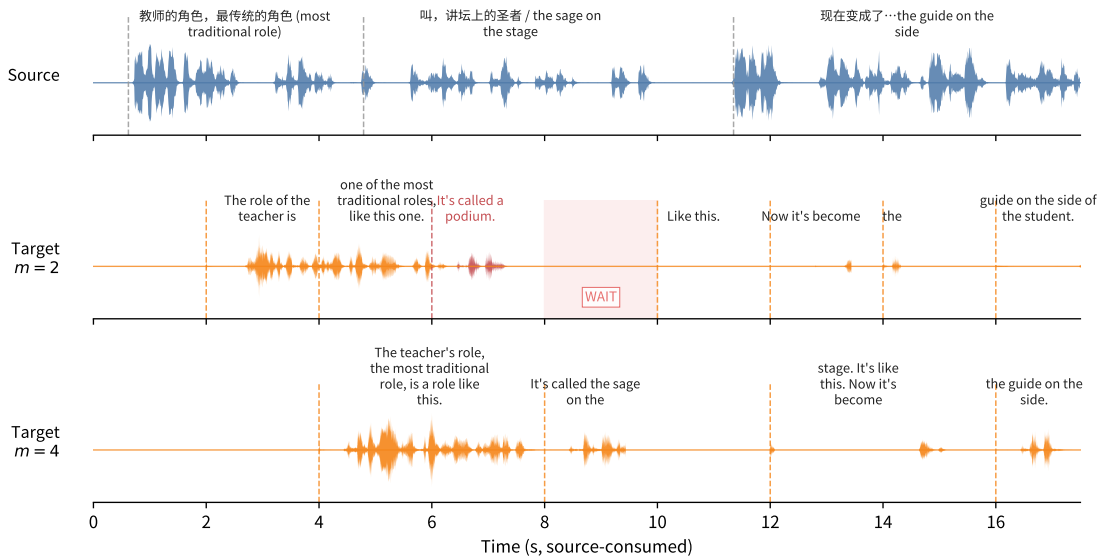

Figure 19: Premature emission / wrong write (Zh→En), source-consumed axis. *Top:* source. *Middle* (m=2)*:* the agent commits *"It's called a podium."* at 6.0 s—before the disambiguating source ("sage") arrives—then stalls in a WAIT. *Bottom* (m=4)*:* the wider read window lets it hear the full phrase first and correctly emit *"It's called the sage on the stage."*

ture, paired-S2ST scale, decoding settings, and metric definitions; this appendix records the training stages, evaluation protocol, and representative configuration files needed to reconstruct the reported comparisons. Exact final release packaging, run identifiers, and artifact manifests will be listed in the public code release.

All public artifacts are used according to their original licenses and release terms.

An AI assistant was used during the paper-writing phase strictly for copy-editing, grammatical polishing, and LaTeX formatting of our drafted text. All core scientific ideas, experimental designs, and data analyses were conceptualized and executed entirely by the human authors.