Test-Time Adaptation of an Offline Multimodal Foundation Model for Simultaneous Speech Translation

Yi Xing, Manli Yu, Pengfei Liu*, and Helen Meng*
Centre for Perceptual and Interactive Intelligence (CPII), Hong Kong SAR, China
{yxing,mlyu,pfliu,hmmeng}@cpii.hk

Abstract

End-to-end simultaneous speech-to-text translation (SimulST) systems typically rely on complex architectures and sophisticated training strategies. In contrast, we propose a simple approach that combines conventional pause-based segmentation for streaming audio input with a strong off-the-shelf multimodal foundation model *adapted at test-time* for translation. To achieve simultaneity, we adopt a variant of the classic wait-k read-write policy to control the interaction between audio input and translation output, and use a multi-turn conversation format with response prefilling and key-value caching for coherent translation and computational efficiency. Experiments on the official development sets of the IWSLT 2026 SimulST shared task show that our system achieves a better quality–latency trade-off than the cascaded baseline across all language directions and latency regimes, highlighting the effectiveness of this simple yet powerful approach.

1 Introduction

Simultaneous speech translation (SimulST) aims to translate source-language speech into target-language text in real time, minimizing delay to enhance user comprehension (Papi et al., 2025). SimulST is challenging due to the inherent quality–latency trade-off and limited availability of task-specific training data (Ko et al., 2023).

Conventional end-to-end SimulST systems are generally divided into two modules: one to handle audio segmentation and the other to determine when to read or write (Liu et al., 2024). These systems typically employ sophisticated adaptive segmentation and/or read-write strategies and architectures as well as specialised training techniques (Liu et al., 2024). An example of using an adaptive strategy for performing audio segmentation is Tsiamas et al. (2022), whereas Dong et al. (2022), Zhang and Feng (2023) and Zhang et al. (2022) are representative cases of learning an adaptive segmentation module. Similarly, adaptive read-write policies were explored by studies like Ma et al. (2020), Polák et al. (2023), Papi et al. (2023), and Ouyang et al. (2025). Despite their effectiveness, these methods typically entail considerable architectural complexity and training cost.

Motivated by the need for practical and efficient SimulST systems, we participate in the speech-to-text track without extra context of the IWSLT 2026 (Adelani et al., 2026) SimulST shared task, and propose a simple yet effective end-to-end SimulST system based on an offline multimodal foundation model. Inspired by Papi et al. (2022) and Deng et al. (2025), our system uses conventional pause-based audio segmentation to process streaming input and an instruction-tuned multimodal foundation model to perform translation. To enable an offline model to operate in a simultaneous setting at test time—a goal shared by Macháček and Polák (2025), we adopt a variant of the classic wait-k read-write policy (Ma et al., 2019). Following Ouyang et al. (2025), we formulate the translation process as a multi-turn conversation with response prefilling. Our contributions are summarised as follows:

- We propose an end-to-end SimulST framework based on pause-based speech segmentation and a multimodal foundation model.
- We adapt the foundation model to simultaneous translation using a deterministic wait-k policy and a multi-turn conversation format.
- We use response prefilling and key-value caching to improve coherence and computational efficiency.
- To the best of our knowledge, we are the first to combine these elements into a single functioning SimulST system, which has achieved a better quality–latency trade-off than a cascaded baseline across multiple language directions

*Corresponding authors.

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 238–246

without fine-tuning or specialised training for simultaneous translation.

2 System Architecture

As illustrated in Figure 1, the proposed system consists of two main modules: one for speech segmentation and the other for speech translation. The segmented audio chunks from the first Segmentation module are passed to the second Translation module, which leverages a powerful, instruction-tuned, off-the-shelf multimodal foundation model, Qwen3-Omni. Qwen3-Omni is a pretrained multilingual model based on a Mixture-of-Experts (MoE) architecture with state-of-the-art capabilities in cross-modal understanding and generation (Xu et al., 2025). Because of its native support for both text and audio modalities, proven strong offline speech recognition and translation performance, and ability to process long-form inputs, it is a suitable candidate for adaptation to the SimulST task.

Translation is performed in a *multi-turn conversation format*, where each turn includes a user message (audio chunk) and an assistant response (translation). The system enforces strict generation control with a variant of the wait-k read-write policy to regulate the lag between speech input and translation output, as well as repetition handling mechanisms. It also uses response prefilling and key-value caching to improve coherence and reduce redundant computation. These measures work together to deliver high-quality, incremental translations. This simple yet robust architecture enables real-time speech translation without the need for specialised training or adaptive modules.

2.1 Variable-Length Speech Segmentation

Incoming audio is segmented into variable-length chunks using the Silero VAD model (Silero Team, 2024), which estimates the speech probability of each 32-ms audio frame. Chunk boundaries are then determined with a simple hybrid approach that combines pause-based segmentation with fixed minimum and maximum chunk durations.

Audio accumulation is triggered when at least n_{a} frames within an n_{w}-frame window have speech probabilities greater than 0.5. Once the accumulated audio reaches the minimum chunk duration, the system searches for a natural pause and places the boundary at the start of the first frame whose speech probability falls below 0.5. If no such pause is detected before the accumulated audio reaches the maximum chunk duration, the boundary is instead placed at the frame with the lowest speech probability within the interval between the minimum and maximum durations. After a chunk is generated, audio accumulation is reset, and the system returns to the activation stage. This segmentation scheme aligns chunk boundaries with natural pauses whenever possible while limiting the latency caused by long stretches of continuous speech.

2.2 Adaptation for Speech Translation

Following Ouyang et al. (2025), we interface with the multimodal foundation model using a multi-turn conversation format. However, rather than fine-tuning the model to learn a read-write policy, we perform test-time adaptation through natural-language instructions and strict generation control. This enables the model to translate incoming speech incrementally while maintaining cross-turn coherence. The proposed test-time adaptation framework consists of four main components: a multi-turn conversation format, generation control with a read–write policy and repetition handling, as well as key-value caching for computational efficiency.

2.2.1 Multi-turn Conversation Format

The conversation format, as illustrated in Figure 2, consists of a system message containing an instruction and a sequence of *user–assistant* turns, where each user message is a chunk of input audio, and assistant responses are translation output. Below is the system instruction:

> You are a simultaneous speech translator. Translate the <source language> speech into <target language>. You will receive audio incrementally. Each response should be the *complete translation of all audio* received so far.

Crucially, the model is instructed to translate not only the latest chunk, but also previous ones, which is necessary because we control how much the model is allowed to generate in a turn (see Section 2.2.2), and there might be syntactical differences between the source and target languages. This instruction is coupled with *response prefilling* (Anthropic, 2026), a practice we have adapted from Koshkin et al. (2024), who implemented an LLM-based simultaneous text-to-text translator: the accumulated translation tokens from prior turns are injected as a prefix at the beginning of each assistant response after the first turn, so that generation resumes from

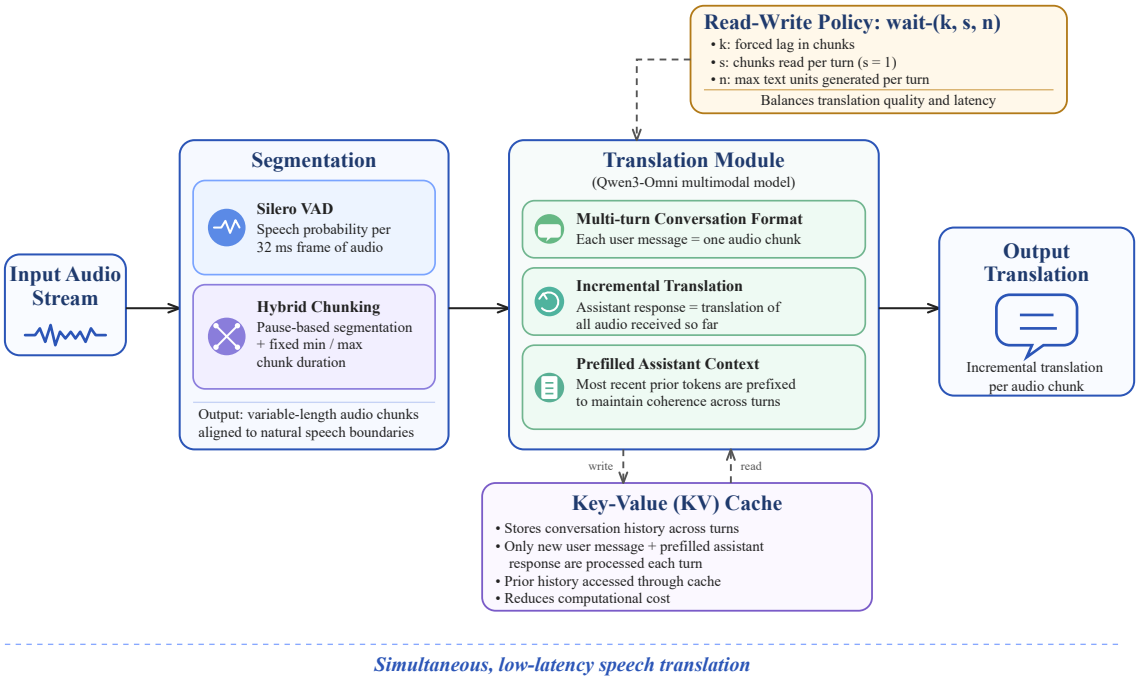

Figure 1: Overview of the proposed SimulST system via test-time adaptation of a multimodal foundation model.

exactly where the previous turn left off, ensuring coherence across turns. In order to reduce overhead, we prefill only a sliding window of the n_{p} most recent tokens rather than the entire translation history.

2.2.2 Read-Write Policy

We adapt the deterministic wait-(k, s, n) policy of Nguyen et al. (2021), in which the system waits for k acoustic frames before writing the first output token and subsequently alternates between reading s frames and writing n tokens. In our setting, where the input is segmented into variable-length audio chunks, the model first reads k chunks of audio. In each subsequent turn, it reads one additional chunk, corresponding to $s = 1$, and is allowed to generate at most n text units. The unit is word for English, German and Italian, and character for Chinese, excluding punctuation and space in both cases. The value of n is estimated based on the expected speaking rate and a language-direction-specific expansion factor, so that the generated output remains consistent with the desired read-write schedule.

In order to enforce this policy, we use a custom inference loop—we run the autoregressive decoding procedure step by step by ourselves instead of using the original end-to-end generation facility, and break out of the loop when n units of text have been generated or an end-of-sequence token is produced, whichever comes first. If we stop a turn due to reaching the n-unit limit, we force-inject an end-of-sequence token to close this turn. At the end of the audio stream, the model is allowed to generate without restrictions.

Theoretically, in this policy, k controls how many audio chunks the translation is forced to lag behind the speech, and with $s = 1$ and the chunk duration configurable, the latter determines translation granularity. n should be set to an appropriate value according to the chunk duration so that the k-chunk lag holds. We find this policy to be a straightforward and easy-to-control mechanism to achieve a good trade-off between quality and latency, without the need to resort to complex adaptive policies or training techniques.

2.2.3 Repetition Handling

An important part of our system is dedicated to preventing repetitions in the model's outputs and, if they still occur, detecting and discarding them. First, we penalise tokens already generated in a turn by subtracting their counts multiplied by a *penalty factor* α_{p} from their logits. We also check for literal repetitions of token n-grams at every decoding step: if there are more than two consecutive occurrences of an n-gram, we discard the last two and regenerate from the point where the first occurrence ends, at the same time setting the repeated tokens' log-

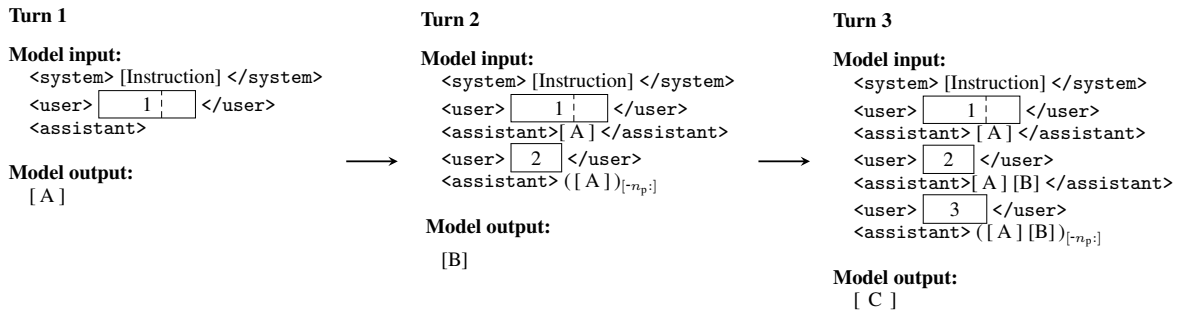

Figure 2: A conversation format with three user–assistant turns. Each turn consists of a user message containing a chunk of input audio (numbered rectangles) and an assistant translation response (square-bracketed capitals), whose prefix is forced to be the last n_{p} tokens of previous translations, and each turn's output can only have at most n units of text (excluding the prefix). A special case is the first turn, whose input consists of k audio chunks merged into one (in this figure, $k = 2$).

its to negative infinity to prevent them from being generated again. In addition to strict literal checks on tokens, we perform fuzzy matching on decoded text periodically and when repetitions are found, discard and regenerate them likewise.

2.2.4 Key-Value Caching

Since the conversation context grows monotonically as the audio stream progresses, we maintain the model's key-value (KV) cache throughout conversation turns. At each turn, only the new user message and the prefilled assistant response are processed, while the previous conversation history remains accessible through the cache. This avoids redundant computation and keeps the per-turn inference cost approximately proportional to the newly added audio input. Under the proposed multi-turn conversation format, KV caching allows the model to compute attention only over the latest turn's audio chunk and the assistant prefix, thereby substantially reducing computational cost.

3 Experiments

3.1 Experimental Datasets

We conduct experiments based on the development set for English-to-German/Chinese/Italian translation named MCIF (Papi et al., 2026), which consists of talks from the Association for Computational Linguistics (ACL) conferences, and that for Czech-to-English translation is a collection of recordings of the Chamber of Deputies' meetings in the Czech Parliament[1]. The statistics of the development sets are given in Table 1. It is worth mentioning that the recordings in the development set are substantially shorter than the maximum input audio duration supported by the Qwen3-Omni model (~40 minutes). Therefore, we do not address support for infinite input streams in this work, although this can be easily done by evicting older conversation history from the model input.

	MCIF	IWSLT26 Czech
# Recordings	21	43
# Segments	919	679
Total Duration (min)	118	109
Total Segment Duration (min)	105	108
Average Segment Duration (sec.)	6.84	9.52
# Words	16,342	13,059
Average Words per Segment	17.8	19.2

Table 1: Statistics of the development sets.

3.2 Experimental Setup

3.2.1 Evaluation Metrics

Evaluation is conducted on unsegmented long-form audio, and the primary evaluation metrics are xCOMET (Guerreiro et al., 2024) for quality and LongYAAL (Polák et al., 2025) for latency. Systems are divided into two latency regimes based on computation-unaware (CU) LongYAAL: **low latency** (0–2 seconds) and **high latency** (2–4 seconds), and we target both in our submission.

3.2.2 Baseline Method

We compare our proposed system with the official baseline of IWSLT 2026[2], which is a cascaded system based on Qwen3-ASR-1.7B (Shi et al., 2026)

[1] http://ufallab.ms.mff.cuni.cz/~polak/iwslt26-cs-dev.zip

[2] https://github.com/owaski/iwslt-2026-baselines

for automatic speech recognition on fixed-size audio chunks and Qwen3-4B-Instruct-2507 (Yang et al., 2025) with the local agreement policy (Liu et al., 2020) for machine translation.

3.2.3 System Parameters

Parameter	Symbol	Value
Penalty factor	α_{p}	0.2
Assistant prefix window size	n_{p}	100
Activation window size	n_{w}	10
# activation frames	n_{a}	3
Min chunk duration (ms)	d_{min}	960
Max chunk duration (ms)	d_{max}	$[1600, 3520]$
# awaited chunks	k	$\{1, 2, 3\}$

Table 2: Main parameters of proposed SimulST system.

The main parameters of our system are listed in Table 2, where some are fixed and others are given in ranges. In particular, d_{max}, k and n are to be determined by hyperparameter search. Among these three, n's choice is related to the language direction and the chunk duration, and we determine a suitable range for each language direction based on the configured range of chunk duration, the source language's speaking rate and the direction's *expansion factor* (Ni et al., 2022): the expansion factor from Language A to B is the average ratio of the number of units in a piece of text in A to that in a translation in B. We assume a speaking rate of 100–150 words per minute for English (Barnard, 2022), and 94–143 for Czech (Psutka et al., 2003)[3], and the expansion factors are listed in Table 3. Then a range can be determined by multiplying the three.

Direction	Expansion Factor
en → de	0.80–0.95
en → zh	1.50–1.80
en → it	1.15–1.25
cs → en	1.10–1.20

Table 3: Expansion factors for the four language translation directions (Trusted Translations, Inc., 2026).

We use the Qwen3-Omni-30B-A3B-Instruct[4] model as the multimodal foundation model.

3.3 Results and Analysis

3.3.1 Speech Segmentation Analysis

Our audio segmentation module is designed to generate chunks that respect natural speech boundaries whenever possible. To assess its effectiveness, we applied the module to the MCIF dataset and analysed the resulting distribution of chunk durations, as shown in Figure 3. The results show that all chunk durations fall within the predefined minimum and maximum limits, while still exhibiting variable lengths. In addition, 79.8% of the chunks were segmented at frames where the speech probability was below 0.5, confirming the module's ability to align chunk boundaries with natural speech pauses.

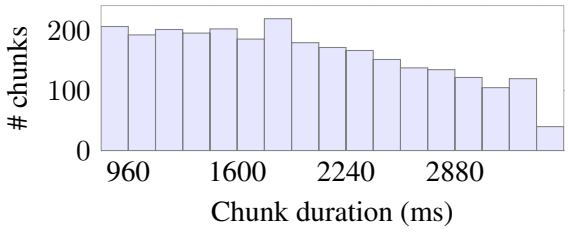

Figure 3: Distribution of chunk durations when $d_{\text{min}} = 960$ ms and $d_{\text{max}} = 3520$ ms on the MCIF dataset.

3.3.2 Hyperparameter Search and Analysis

We performed a coarse-grained hyperparameter search over 20 combinations per language direction, restricting k to $2, 3$ and using a step size of 320 ms for d_{max} and odd values for n. Since this initial search predominantly yielded high-latency configurations for English→German, English→Chinese, and English→Italian, we conducted a second round comprising 26 additional combinations for these three directions, this time focusing on $k = 1$ and smaller values of d_{max}.

After completing inference over these ~160 hyperparameter combinations, we selected one configuration per language direction and latency regime based on the quality–latency trade-off and generalisation ability. As described in Section 3.2.1, we measured quality using corpus-level COMET scores and latency using LongYAAL; in addition, we computed recording-level COMET scores and used their standard deviation as a measure of generalisation ability. We then divided the results into low- and high-latency regimes and, within each regime, shortlisted the candidates with the highest COMET scores. The final configuration for each language direction and latency regime was chosen by jointly considering all three metrics, prioritising translation quality and generalisation over latency, as shown in Table 4.

As depicted in Figure 4, higher latency is generally correlated with better quality, except for some

[3] In this study, we use the above range instead of 64–173 wpm in the original paper to speed up the search.

[4] https://huggingface.co/Qwen/Qwen3-Omni-30B-A3B-Instruct

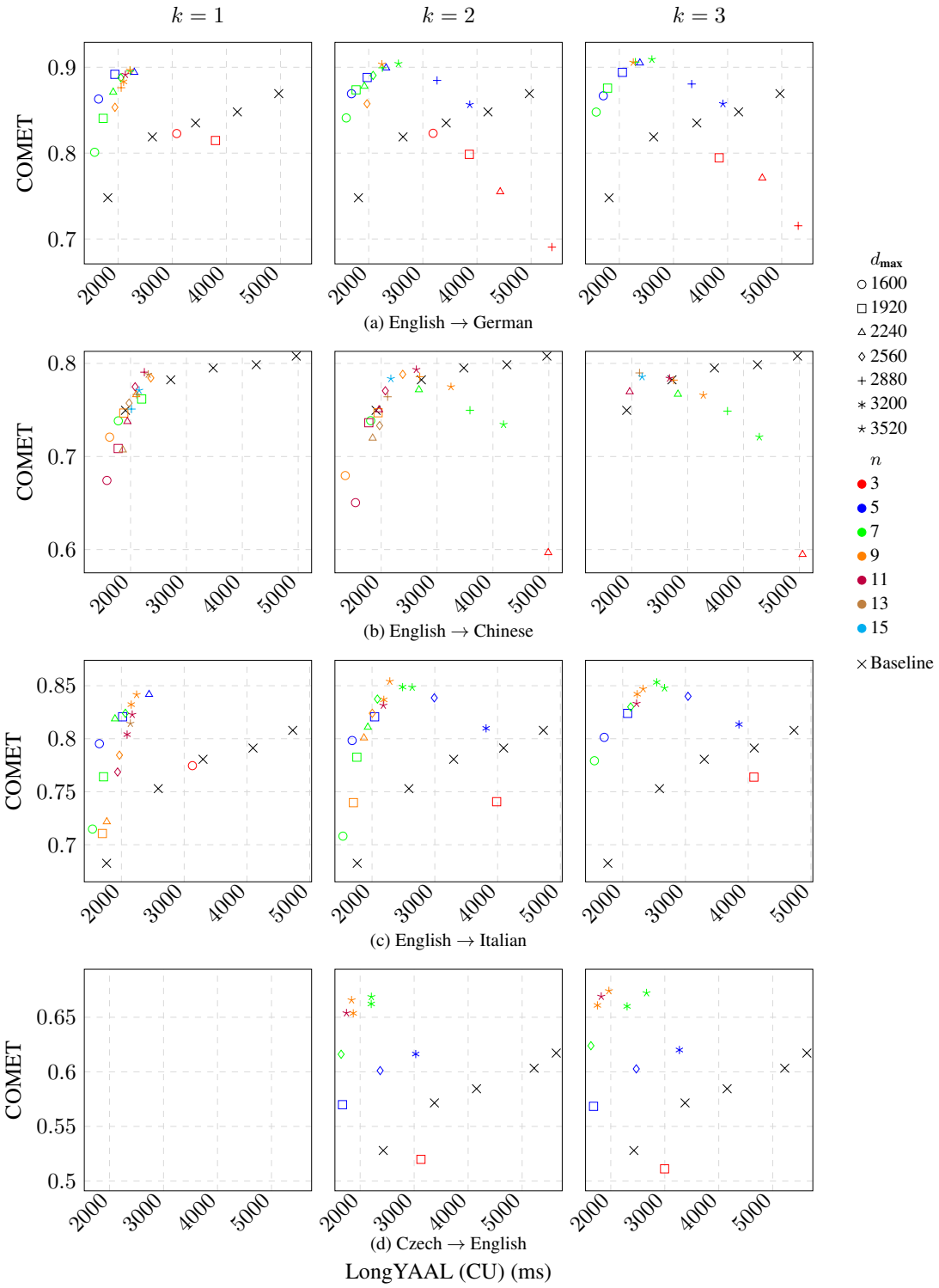

Figure 4: Quality–latency trade-off analysis for the four language directions under different hyperparameter combinations ($d_{\max}, k, n$). The hyperparameters should all be inside appropriate ranges to make the system achieve good performance; either too small or too large values can lead to suboptimal results. Note that the baseline results for the first three language directions are official,[5] whereas the Czech-to-English baseline was generated from our own experiments. The first subfigure for the Czech-to-English direction is left blank because we omitted inference at $k = 1$ due to time constraints.

outliers with very small n or large $d_{\max}$. This is because too small an n can 'suffocate' the model and make the translation fall far behind the speech's progress (the red/blue points in the bottom right

[5] https://github.com/user-attachments/files/26411361/outputs.zip

corner of each subfigure of Figure 4). These hyperparameters themselves are also correlated—for example, a large $d_{\max}$ must be paired with an n large enough to make the model keep up with the speech, or the quality drops even with higher latency (green stars in Figure 4(b)); also, if $d_{\max}$ and n are in suitable ranges, a larger k can lead to better quality at the cost of higher latency. When all of the three hyperparameters are in appropriate ranges and well-coordinated, the system can achieve good quality with reasonable latency (points in the upper-left corner of each subfigure in Figure 4).

Direction	Latency	Setting			Metrics		
		$d_{\max}$	k	n	COMET ↑	LongYAAL (CU) (ms) ↓	σ_{COMET} ↓
en → de	Low	1920	1	5	0.8918	1940	0.0330
		1920	2	5	0.8881	1972	0.0283
		2240	2	7	0.8779	1920	0.0469
	High	3520	3	7	0.9091	2601	0.0280
		2880	3	7	0.9060	2296	0.0256
		3520	3	9	0.9057	2258	0.0190
en → zh	Low	2240	3	11	0.7696	1962	0.0433
		2560	1	13	0.7574	1974	0.0559
		2240	2	11	0.7502	1961	0.0502
	High	3520	2	11	0.7934	2629	0.0352
		2880	1	11	0.7906	2247	0.0364
		2880	3	13	0.7898	2138	0.0344
en → it	Low	2240	1	7	0.8188	1897	0.0413
		2240	2	7	0.8109	1933	0.0631
		1600	3	5	0.8013	1706	0.0380
	High	3520	2	9	0.8541	2283	0.0347
		3200	3	7	0.8532	2542	0.0368
		3200	2	7	0.8487	2486	0.0392
cs → en	Low	3520	3	9	0.6742	1965	0.131
		3520	3	11	0.6691	1823	0.134
	High	3520	3	7	0.6723	2662	0.142
		3520	2	7	0.6688	2208	0.133

Table 4: Hyperparameter settings for each language direction and latency regime. The chosen settings of our submitted system are highlighted in blue.

3.3.3 Comparison with Baseline

Due to the lack of access to the test sets, we employ k-fold cross validation to evaluate the performance of our proposed system and compare it with the baseline's. We first divided the development sets into k folds, then used each of the folds as held-out validation data and the rest for hyperparameter search[6]: for each language direction and latency regime, we selected the hyperparameter combination that has the highest average COMET score on the $k-1$ folds of recordings, then evaluated its quality and latency on the held-out fold. Finally, we computed the mean of the k sets of metrics, as shown in Table 5, under $k=5$.

As shown in Table 5, our system achieves a better overall quality–latency trade-off than the baseline across the evaluated language directions, with particularly large gains for English-to-German, English-to-Italian and Czech-to-English translation, possibly because of the three pairs' genealogical affinity. For English-to-Chinese translation, the improvement is more modest: our system attains higher COMET in the low-latency setting and substantially lower latency in the high-latency setting, although with a small drop in COMET. This may reflect the inherent difficulty of translating in this language direction, as English and Chinese belong to distant language families and have considerable differences lexically and syntactically.

Direction	Latency	System	COMET ↑	LongYAAL (CU) (ms) ↓
en → de	Low	Baseline	0.7656	**1809**
		Ours	**0.8884**	1982
	High	Baseline	0.8616	3797
		Ours	**0.8995**	**2511**
en → zh	Low	Baseline	0.7555	**1873**
		Ours	**0.7593**	2001
	High	Baseline	**0.8063**	3769
		Ours	0.7670	**2504**
en → it	Low	Baseline	0.6873	**1827**
		Ours	**0.8231**	1950
	High	Baseline	0.7936	3639
		Ours	**0.8467**	**2439**
cs → en	Low	Baseline	0.4978	2366
		Ours	**0.6356**	**1957**
	High	Baseline	0.5464	3547
		Ours	**0.6482**	**2711**

Table 5: Performance comparison of our system with the baseline using five-fold cross validation.

4 Conclusion

This paper presented our submission to the main track of the IWSLT 2026 SimulST shared task. Our system first applies a hybrid VAD-based approach for audio segmentation, and then performs translation with an offline multimodal foundation model adapted to the task at test time. For test-time adaptation, we design a fixed read–write policy inspired by wait-k, together with a multi-turn conversation format that incorporates response prefilling and KV caching. We conducted a hyperparameter search over representative combinations of three key hyperparameters and selected suitable configurations for each language direction and latency regime. Experiments on the development sets showed that our system outperforms the cascaded baseline in all evaluated language directions, and its decent zero-shot performance on Czech-to-English translation is particularly noteworthy. These results demonstrated the efficacy and efficiency of our approach.

[5] https://github.com/user-attachments/files/26411361/outputs.zip

[6] The baseline system has only one hyperparameter—the chunk duration.

Limitations

First of all, the multi-turn conversation format can lead to error accumulation: once a single turn goes awry, for example, by refusing to follow the instruction, subsequent turns might follow suit. Also, KV cache management can be further optimised to handle infinite audio streams by introducing a sliding window mechanism. In addition, hyperparameter search is coarse-grained and only performed on two small and single-domain datasets. Last but not least, inference costs of a large multimodal foundation model are significant because of its relatively large size, resulting in higher computation-aware latency.

Acknowledgements

This work was supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd., a CUHK-led InnoCentre under the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

References

David Ifeoluwa Adelani, Victor Agostinelli, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Sebastien Bratières, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, Marcello Federico, Marco Gaido, Mahendra Gupta, HyoJung Han, Ali Hatami, David Javorský, Yejin Jeon, Marek Kasztelnik, Antoine Laurent, and 33 others. 2026. Speech translation and metrics in 2026: Findings of the iwslt campaign. In *Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026)*, San Diego, California, US. Association for Computational Linguistics.

Anthropic. 2026. Prompting best practices: Migrating away from prefilled responses.

Dom Barnard. 2022. Average Speaking Rate and Words per Minute.

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, Vienna, Austria. Association for Computational Linguistics.

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

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Yuka Ko, Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Tagged end-to-end simultaneous speech translation training using simultaneous interpretation data. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 363–375, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Roman Koshkin, Katsuhito Sudoh, and Satoshi Nakamura. 2024. TransLLaMa: LLM-based Simultaneous Translation System. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 461–476, Miami, Florida, USA. Association for Computational Linguistics.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Interspeech 2020*, pages 3620–3624.

Xiaoqian Liu, Guoqiang Hu, Yangfan Du, Erfeng He, YingFeng Luo, Chen Xu, Tong Xiao, and Jingbo Zhu. 2024. Recent advances in end-to-end simultaneous speech translation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8142–8150. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

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

Dominik Macháček and Peter Polák. 2025. Simultaneous translation with offline speech and LLM models in CUNI submission to IWSLT 2025. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 389–398, Vienna, Austria (in-person and online). Association for Computational Linguistics.

Ha Nguyen, Yannick Estève, and Laurent Besacier. 2021. An Empirical Study of End-To-End Simultaneous Speech Translation Decoding Strategies. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7528–7532.

Jingwei Ni, Zhijing Jin, Markus Freitag, Mrinmaya Sachan, and Bernhard Schölkopf. 2022. Original or translated? a causal analysis of the impact of translationese on machine translation performance. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5303–5320, Seattle, United States. Association for Computational Linguistics.

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

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Does simultaneous speech translation need simultaneous models? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 141–153, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sara Papi, Matteo Negri, and Marco Turchi. 2023. Attention as a Guide for Simultaneous Speech Translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada. Association for Computational Linguistics.

Sara Papi, Peter Polák, Dominik Macháček, and Ondřej Bojar. 2025. How "real" is your real-time simultaneous speech-to-text translation system? *Transactions of the Association for Computational Linguistics*, 13:281–313.

Sara Papi, Maike Züfle, Marco Gaido, Beatrice Savoldi, Danni Liu, Ioannis Douros, Luisa Bentivogli, and Jan Niehues. 2026. MCIF: Multimodal crosslingual instruction-following benchmark from scientific talks. In *The Fourteenth International Conference on Learning Representations*.

Peter Polák, Sara Papi, Luisa Bentivogli, and Ondřej Bojar. 2025. Better Late Than Never: Evaluation of Latency Metrics for Simultaneous Speech-to-Text Translation. *arXiv preprint*. ArXiv:2509.17349 [cs].

Peter Polák, Brian Yan, Shinji Watanabe, Alex Waibel, and Ondřej Bojar. 2023. Incremental Blockwise Beam Search for Simultaneous Speech Translation with Controllable Quality-Latency Tradeoff. In *Interspeech 2023*, pages 3979–3983.

Josef Psutka, Pavel Ircing, J.V. Psutka, Vlasta Radova, William J. Byrne, Jan Hajic, Jiri Mirovsky, and Samuel Gustman. 2003. Large vocabulary ASR for spontaneous czech in the MALACH project. In *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, pages 1821–1824.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-ASR Technical Report. *arXiv preprint*. ArXiv:2601.21337 [cs].

Silero Team. 2024. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. `https://github.com/snakers4/silero-vad`.

Trusted Translations, Inc. 2026. Translation Quote per Word.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022. SHAS: Approaching optimal Segmentation for End-to-End Speech Translation. In *Interspeech 2022*, pages 106–110.

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. 2025. Qwen3-Omni Technical Report. *arXiv preprint*. ArXiv:2509.17765 [cs].

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. *arXiv preprint*. ArXiv:2505.09388 [cs].

Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2022. Learning Adaptive Segmentation Policy for End-to-End Simultaneous Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7862–7874, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2023. End-to-End Simultaneous Speech Translation with Differentiable Segmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7659–7680, Toronto, Canada. Association for Computational Linguistics.