SimulMEGA: MoE Routers are Advanced Policy Makers for Simultaneous Speech Translation

Chenyang Le
nethermanpro@sjtu.edu.cn

Bing Han
hanbing97@sjtu.edu.cn

Jinshun Li
muzi-jingshun@sjtu.edu.cn

Songyong Chen
chensy30@sjtu.edu.cn

Yanmin Qian *
yanminqian@sjtu.edu.cn

Auditory Cognition and Computational Acoustics Lab
MoE Key Lab of Artificial Intelligence, AI Institute
School of Computer Science, Shanghai Jiao Tong University, Shanghai, China

Abstract

Simultaneous Speech Translation (SimulST) enables real-time cross-lingual communication by jointly optimizing speech recognition and machine translation under strict latency constraints. Existing systems struggle to balance translation quality, latency, and semantic coherence, particularly in multilingual many-to-many scenarios where divergent read/write policies hinder unified strategy learning. In this paper, we present **SimulMEGA**(**Simul**taneous Generation by **M**ixture-of-**E**xperts **GA**ting), an unsupervised policy learning framework that combines prefix-based training with a Mixture-of-Experts refiner to learn effective read/write decisions in an implicit manner, without adding inference-time overhead. Our design requires only minimal modifications to standard transformer architectures and generalizes across both speech-to-text and text-to-speech streaming tasks. Through comprehensive evaluation on six language pairs, our 500 M-parameter speech-to-text model outperforms the Seamless baseline, achieving under 7% BLEU degradation at 1.5 s average lag and under 3% at 3 s. We further demonstrate SimulMEGA's versatility by extending it to streaming TTS with a unidirectional backbone, yielding superior latency–quality trade-offs. [2]

1 Introduction

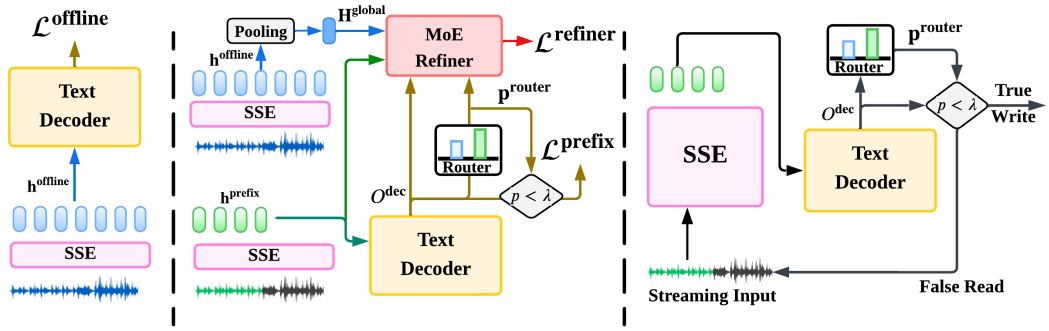
Simultaneous Speech Translation (SimulST) addresses the critical need for real-time cross-lingual communication by jointly optimizing speech recognition and machine translation under strict latency constraints. Unlike conventional offline systems that process complete utterances, SimulST operates on streaming audio input, incrementally generating translations while simultaneously decoding ongoing speech - a capability mirroring human interpreters' cognitive processing. This technology enables transformative applications in international diplomacy, live media localization, and low-latency dialogue systems. However, it faces fundamental challenges in reconciling three competing objectives: translation quality, computational latency, and semantic coherence, exacerbated by fragmented acoustic patterns and partial contextual dependencies.

Recent progress combines adaptive segmentation strategies[1, 2, 3] with streaming architectures like RNN-T transducers[4, 5, 6] and Monotonic Attention mechanisms[7, 8], alongside policy-based

*Corresponding Author

[2]The code can be found in `https://github.com/nethermanpro/simulmega`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) Offline Training (b) Prefix-based Simultaneous Training (c) Simultaneous Inference

Figure 1: Overview of the training and inference paradigm of SimulMEGA. SimulMEGA is composed of a Streaming Speech Encoder, a text decoder, an MoE routing Gate(Router), and an MoE refiner. In the first stage, the model is pre-trained on $\mathcal{L}^{\text{offline}}$. In the second stage, the model is trained on a combination of $\mathcal{L}_{\text{offline}}$, $\mathcal{L}^{\text{prefix}}$ and $\mathcal{L}^{\text{refiner}}$. SSE denotes Streaming Speech Encoder.

decision frameworks[9, 10, 11, 12, 13, 14]. Despite these advances, current systems struggle to achieve human-like efficiency-accuracy trade-offs across diverse linguistic contexts and acoustic conditions. A particularly underexplored challenge lies in multilingual many-to-many translation, where divergent read/write policies across language pairs complicate the learning of unified operational strategies.

This paper introduces SimulMEGA, an unsupervised policy learning framework that synergizes prefix-based training with Mixture-of-Experts (MoE) mechanisms to address these limitations. Our approach enables high-performance multilingual SimulST through three key innovations: (1) An auxiliary MoE refiner module enabling implicit policy learning without inference-time overhead, (2) Minimal architectural modifications to standard transformers for broad applicability, and (3) A unified framework supporting both speech-to-text and text-to-speech streaming tasks. Experimental results demonstrate state-of-the-art performance across six languages, with our 500M-parameter speech-to-text model outperforming SeamlessM4T[15] by achieving <7% BLEU degradation at 1.5s average lagging (AL) and <3% at 3s AL. What's more, we extended our methods to streaming text-to-speech (TTS) task via CosyVoice2's unidirectional backbone[16], the system achieves superior latency control compared to conventional interleaved streaming approaches.

Our principal contributions are as follows.

- **Introduction of SimulMEGA**: An unsupervised policy learning framework that combines prefix-based training with a Mixture-of-Experts (MoE) refiner module, enabling efficient and effective simultaneous speech translation without adding inference-time overhead.
- **Multilingual Advancement**: SimulMEGA achieves SOTA performance for quality-latency trade-offs in many-to-many translation, demonstrating cross-lingual robustness through comprehensive 6-language evaluation.
- **Universal Streaming Framework**: SimulMEGA supports both speech-to-text and text-to-speech streaming tasks within a single framework, making it broadly applicable and easily integrable into existing models for multilingual SimulST tasks.
- **General Conversational System**: The integrated solution combining low-latency speech-to-text and text-to-speech conversion, validated through real-world dialog interpretation scenarios.

2 Related Works

SimulST Early works in SimulST focus on finding the input segmentation boundary and then applying a variant of the wait-k policy[1, 17, 2, 3]. Later work seeks to find a more sophisticated dynamic policy. [18] designs a learnable matrix between source and target for an integrate-and-fire(IF) policy, and [19, 20] use attention score for a similar approach. [11] proposes a policy based on divergence between the prefix and full input. [12] employs an extra CTC[21] head for policy control. Another approach includes using an inherently streaming architecture like monotonic

attention[7, 22, 8, 23], transducer[4, 5, 6] and Hidden Markov[24]. [25] explores non-autoregresive generation for SimulST.

Streaming TTS Many works on TTS claim to be streamable[26, 27], yet they are not optimized for fine-grained(chunk or token level) streaming and suffer from degraded quality due to a lack of future information. [28] adapts a streaming model from an offline model via restricted attention. [29] uses a fixed number of text tokens for look-ahead. [30] processes each text chunk in a separate AR loop with look-ahead. [16] interleaves text and speech tokens at a fixed ratio, which potentially incurs larger latency.

3 SimulMEGA: Simultaneous Translation via Mixture-of-Experts Routing

In this section, we introduce SimulMEGA (**Simul**taneous Generation by **M**ixture-of-**E**xperts **GA**ting), an innovative framework that effectively converts an offline autoregressive model into a simultaneous system with minimal computational overhead and negligible performance degradation.

The core principle of simultaneous training lies in enabling the model to autonomously determine whether the currently available input suffices for generating the next output token. Previous approaches typically rely on either artificial policy signals, which constrain the model's self-learning capacity, or architectural modifications that compromise generation quality. In contrast, our framework leverages a Mixture-of-Experts (MoE) architecture to achieve unsupervised policy learning while maintaining an identical structure to the offline model, thereby preserving translation performance. Our approach employs a prefix-based training strategy that closely mimics real-world streaming inference scenarios.

3.1 Architecture Overview

Figure 1 illustrates the overall architecture of our SimulMEGA framework, which consists of four key components: (1) a streaming speech encoder, (2) a text decoder, (3) a global routing gate, and (4) a Mixture-of-Experts (MoE) refiner module.

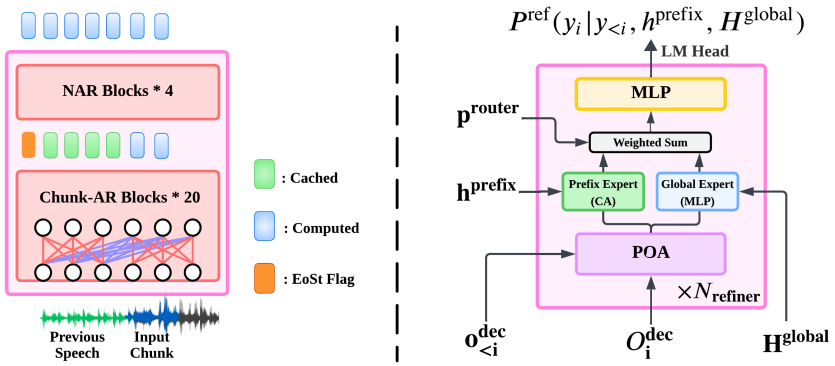
The streaming speech encoder and text decoder adopt the Transformer architecture, maintaining compatibility with conventional speech-to-text translation (S2TT) systems. As illustrated in Figure 2(a), the encoder combines chunk-wise autoregressive (Chunk-AR) blocks with non-autoregressive (NAR) blocks in a hybrid design. The Chunk-AR blocks optimize inference efficiency through a cached key-value mechanism, while the NAR blocks preserve translation quality by capturing global context. To handle streaming inputs lacking an explicit end-of-sequence (EOS) token, we prepend a learnable end-of-stream (EoSt) flag (a binary embedding) to the NAR blocks' input, signaling whether the current chunk terminates the audio stream. The text decoder follows a standard autoregressive Transformer architecture.

The core innovation lies in the routing gate and MoE refiner, which enable unsupervised learning of simultaneous translation policies without architectural modifications. The MoE refiner follows a transformer-like architecture and shares language model head with the text decoder, which combines prefix information and global information and predicts the target translation sequence. It is only activated during training, introducing zero additional computational overhead during inference.

3.2 Unsupervised Policy Learning.

This section introduces the Mixture of Experts (MoE) refiner module, whose architecture is shown in Figure 2(b). The module comprises N_{refiner} blocks, each containing two specialized experts: a prefix expert (E_p) and a global expert (E_g). The merging weights of these experts implicitly define a policy that determines whether the current input prefix contains sufficient information for generating the target token. Besides the dual-experts module, each block also contains a previous-output attention module (similar to self-attention) and an MLP module, which is analogous to a standard transformer decoder block.

Global Routing Gate For decision consistency, each refiner layer employs a shared global gate implemented as a two-layer MLP with a Sigmoid head. The gate projects the text decoder's final hidden state (O^{dec}) into a scalar value $p \in [0, 1]$, which determines the expert weights: $P_{E_g}^{\text{Router}} = p$ and $P_{E_p}^{\text{Router}} = 1 - p$.



(a) Streaming Speech Encoder (b) MoE Refiner

Figure 2: (a) Structure and inference example of the streaming speech encoder of SimulMEGA. It comprises 20 chunkwise autoregressive (Chunk-AR) blocks and 4 non-autoregressive(NAR) blocks. After each read, the Chunk-AR blocks only compute the new chunk while the NAR blocks recompute the whole sequence. An End-of-Stream (EoSt) flag is given before NAR blocks. (B) The structure of the MoE Refiner, in which the gate decides over the mixture proportion of the Prefix Expert and the Global Expert. This proportion reflects the model's confidence in the prefix sequence, leading to a natural read/write policy. The self-attention module is replaced by a previous output attention(POA) module to prevent global information leakage.

Dual-Expert Architecture As shown in Figure 1(b), during training, the encoder processes both the complete offline speech and a randomly truncated prefix, producing corresponding hidden states h^{offline} and h^{prefix}. Then the gate and the dual-expert architecture will decide which hidden states to use. While the router typically favors the more informative h^{offline}, we introduce an information bottleneck to balance this preference. Specifically, we apply temporal mean-pooling to h^{offline}, yielding a condensed global embedding H^{global} for E_g. E_g is a two-layer MLP module whose calculation is as follows:

$$\widehat{\mathbf{x}^{\text{in}}_{\mathbf{i}}} = \text{LayerNorm}\left(\mathbf{x}^{\text{in}}_{\mathbf{i}}\right) \tag{1}$$

$$\mathbf{x}^{\mathbf{E_g}}_{\mathbf{i}} = \mathbf{W}^{\text{E}_\text{g}}_{\text{out}} \cdot \text{ReLU}\left(\mathbf{W}^{\text{E}_\text{g}}_{\text{in}} \cdot \left[\widehat{\mathbf{x}^{\text{in}}_{\mathbf{i}}}; \mathbf{H}^{\text{global}}\right]\right) \tag{2}$$

where x^{in}_i denotes the i-th position's input, $[\cdot;\cdot]$ represents vector concatenation, $\mathbf{W}^{\text{E}_\text{g}}_{\text{in/out}}$ are learnable projection weights (biases omitted). The prefix expert E_p employs a standard cross-attention:

$$\mathbf{x}^{\mathbf{E_P}}_{\mathbf{i}} = \mathbf{W}^{\text{E}_\text{P}}_{\text{O}} \cdot \text{MHA}\left(\mathbf{W}^{\text{E}_\text{P}}_{\text{Q}}\widehat{\mathbf{x}^{\text{in}}_{\mathbf{i}}},\ \mathbf{W}^{\text{E}_\text{P}}_{\text{K}} h^{\text{prefix}},\ \mathbf{W}^{\text{E}_\text{P}}_{\text{V}} h^{\text{prefix}}\right)$$

Where MHA denotes multi-head attention. The final output combines both experts' contributions through a gated residual connection:

$$\mathbf{x}^{\text{out}}_{\mathbf{i}} = \mathbf{x}^{\text{in}}_{\mathbf{i}} + P^{\text{Router}}_{i,E_g} \cdot \mathbf{x}^{\mathbf{E_g}}_{\mathbf{i}} + P^{\text{Router}}_{i,E_p} \cdot \mathbf{x}^{\mathbf{E_P}}_{\mathbf{i}}$$

Global Information Leakage In standard transformer architectures, the self-attention mechanism inherently leaks global information across the sequence. This poses a critical issue for our design: even when the gate assigns $P^{\text{Router}}_{E_g} = 0$ for a certain position, the hidden state at that position may still access global context via self-attention, thereby undermining the intended expert specialization. To enforce strict isolation of global information, we replace self-attention with a previous-output attention mechanism. Instead of attending to hidden states within the same layer, each position only attends to the decoder's prior outputs($o^{dec}_{<i}$), effectively preventing unintended information flow while preserving sequential dependencies.

Training Protocol SimulMEGA leverages a pre-trained offline S2TT model as its foundation. The training process consists of two stages:

1. **Offline Pretraining:** We first train the model with the standard offline S2TT objective $\mathcal{L}^{\text{offline}}$(incorporating streaming chunk masks in the encoder) until convergence, as illustrated in Figure 1(a).

2. **Simultaneous Training:** We then introduce two additional loss functions to enable simultaneous capabilities as illustrated in Figure 1(b):

$$\mathcal{L}^{\text{refiner}} = -\sum_{i} \log p^{\text{ref}}(y_i|y_{<i}, h^{\text{prefix}}, H^{\text{global}}) \tag{3}$$

$$\mathcal{L}^{\text{prefix}} = -\sum_{i:p_i<\lambda} \log p^{\text{dec}}(y_i \mid y_{<i}, h^{\text{prefix}}) \tag{4}$$

where p^{ref} and p^{dec} are the output distribution of the MoE refiner and the text decoder. λ is a pre-defined hyperparameter that restricts the losses to the confident position.$\mathcal{L}^{\text{refiner}}$ learns the read/write policy while $\mathcal{L}^{\text{prefix}}$ strengthens the prefix-based translation capability. In this stage, the total loss is the weighted sum of the above three losses:

$$\mathcal{L}^{\text{simul}} = \mathcal{L}^{\text{offline}} + w_{\text{r}} \cdot \mathcal{L}^{\text{refiner}} + w_{\text{p}} \cdot \mathcal{L}^{\text{prefix}} \tag{5}$$

In our experiment, we set $w_{\text{r}} = w_{\text{p}} = 0.2$ to prioritize the offline training.

Inference Policy During inference, SimulMEGA employs a straightforward threshold-based policy:

$$\text{Action} = \begin{cases} \text{Write} & \text{if } p_{t,i} < \lambda \\ \text{Read} & \text{otherwise} \end{cases} \tag{6}$$

where $p_{t,i}$ is the gating score at input time t and target position i.

3.3 Score Distribution Control

The raw output scores from the routing gate lack inherent interpretability, as they emerge solely from neural network optimization through gradient descent. These scores exhibit inconsistent statistical properties, both in mean and variance, between different tasks, language pairs, and training data. This variability poses significant challenges during inference, necessitating task-specific or language-specific threshold tuning that fundamentally compromises the generalizability of our approach. To overcome this limitation, we introduce the following techniques:

Score Normalization We introduce a heuristic that aligns the average gating score with the relative information content between prefix and global contexts. Since sequence length serves as a practical proxy for information quantity, we formulate a normalization loss based on the prefix length (l_p) and full sequence length (l_g):

$$L_{\text{norm}} = \text{SmoothL1Loss}\left(\overline{p},\ \frac{\min(l_p, l_b) \times 0.5 + (l_g - l_p)}{l_g}\right) \tag{7}$$

where $\overline{p}$ represents the sequence's mean gate output, and l_b is a buffer hyperparameter (set to 1.5 seconds in our experiments) that prevents abrupt score changes near prefix boundaries. The 0.5 weighting factor accounts for the partial information available at prefix edges.

Pre-Sigmoid Gaussian Noise Following prior work [7], we incorporate Gaussian noise before Sigmoid activation to promote discretization of gate output. While this discretization does not directly improve model performance, it significantly enhances robustness by reducing sensitivity to threshold selection across diverse tasks. In our implementation, we apply zero-mean unit-variance Gaussian noise ($\sigma_R = 1$) to the pre-Sigmoid logits.

Combined with our normalization technique, this approach enables precise control over both the mean and variance of gating scores. The complete training objective is:

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{simul}} + w_{\text{n}} \cdot \mathcal{L}^{\text{norm}} \tag{8}$$

where we set a small normalization weight $w_{\text{n}} = 0.01$.

3.4 Streaming TTS

A common practice in speech-to-speech translation (S2ST) systems involves integrating a speech-to-text translation (S2TT) module with a text-to-speech (TTS) component. For simultaneous translation scenarios, however, a streaming TTS system is essential to preserve the low-latency requirement of the overall pipeline. While specialized streaming TTS systems exist, we demonstrate that SimulMEGA

offers a universal and straightforward method to convert a standard transformer-based autoregressive TTS system into a streaming-capable variant.

For our experiments, we adopt CosyVoice 2 as the base model, which consists of a decoder-only language model and a flow-based acoustic model. During adaptation, we retain the flow model and fine-tune the language model using the techniques and loss functions outlined in Sections 3.2 and 3.3. Randomly initialized routing gate and the MoE refiner are incorporated during training. Since the backbone lacks an encoder, we derive the prefix and offline hidden states (h^{prefix} and h^{offline}) from the final hidden representations of the text tokens. A detailed architectural diagram is provided in the appendix. We compare our approach against CosyVoice2's default dual-stream mode.

Through this work, we not only develop a more comprehensive S2ST system capable of real-time speech-to-speech translation but also validate the compatibility of SimulMEGA with diverse tasks and model architectures. We leave an end-to-end S2ST system with SimulMEGA for future work.

4 Experiments and Results

4.1 Experiment Settings

Task & Data We collect data and train on many-to-many speech-to-text translation across six popular languages: EN, ZH, DE, ES, FR, and IT. We collect various open-source speech recognition datasets covering the six languages, including LibriSpeech [31], Multilingual Librispeech(MLS) [32], VoxPopuli [33], Common Voice[34], WenetSpeech[35], KeSpeech[36] and Emilia [37]. Then we create pseudo translation labels for these data by translating the transcription into different languages through a cloud text-to-text translation API. The dataset consists of approximately 100K hours of training data. For TTS experiment, we use a subset of ST training set that only contains Chinese and English data due to the language compatibility of the CosyVoise 2.

Model Our offline base model for ST is derived from the whisper medium model. For inference efficiency, we transformed the encoder into a streaming encoder and pruned half of the layers in the decoder. This results in an encoder with 20 chunk-AR blocks and 4 NAR blocks, and a decoder with 12 layers. Offline base model for TTS is CosyVoice 2 as mentioned in section 3.4. In both the ST and TTS experiment, the MoE refiner consists of $N_{\text{refiner}} = 6$ layers, with the hidden size matching that of the base model.

Training Each experiment is trained in FP16 with 8 Nvidia H800 GPUs. In stage 1 offline training of the ST experiment, to retain the capability of the whisper encoder, we employ Low-Rank Adaptation(LoRA)[38] ($\alpha = 64$) at the chunk-AR blocks of the encoder. We train the offline model for 1M steps, which takes around 1 week. In stage 2 training, the chunk-AR blocks are frozen. The router and MoE Refiner module are randomly initialized. The stage 2 training takes about 2 days. We use AdamW optimizer[39] and linear learning rate scheduler with 5000 steps of warmup. The maximum learning rate is 1e-4 in stage 1 training and 1e-5 in stage 2 training.

Evaluation In S2TT evaluation, we employ case-sensitive BLEU[40] with punctuation[3] as our primary quality metric and average lagging (AL)[4] [43] for latency assessment. Our evaluation spans two benchmark datasets: CoVoST2 [44] and Fleurs [45]. For CoVoST2, we report averaged scores across 5 non-English to English (X-EN) translation pairs and 2 English to non-English (EN-X) pairs. The Fleurs dataset enables comprehensive many-to-many evaluation through its parallel data across six languages, yielding results for all 30 possible language pair combinations. We present separate averages for 5 X-EN pairs, 5 EN-X pairs, and 30 X-X pairs.

For TTS evaluation, we combine LibriSpeech-PC *test-clean* [31, 46] and Seed-TTS *test-zh* [47] with CoVoST2 data. Our analysis includes recognition word error rate (WER), speaker similarity (SIM), and text-unit-to-speech-unit average lagging(AL). In S2ST evaluation on CoVoST2, we measure ASR-BLEU against speech-to-speech-unit AL. For English metrics, we utilize Whisper-Large-V3, while Mandarin evaluations employ Paraformer [48]. And for SIM, we employ a speaker verification model based on WavLM-large [49] to extract speaker embeddings. These embeddings are then used to compute the cosine similarity between synthesized speech and ground truth speech.

[3] https://github.com/mjpost/sacrebleu[41]

[4] Code is borrowed from https://github.com/facebookresearch/SimulEval[42]

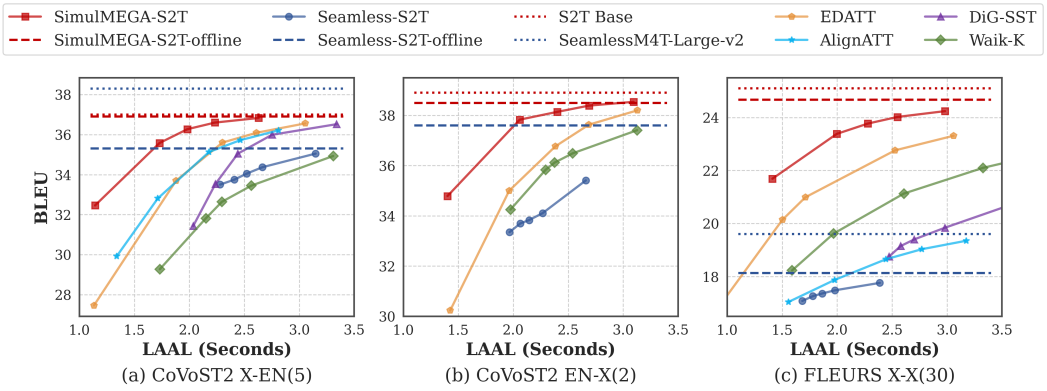
Table 1: The offline BLEU score(%) of different models on CoVoST2 and Fleurs testset, where all results are based on greedy search. Only parameters involved in the S2TT inference are calculated. The value in the bracket denotes the performance degradation compared to the offline base model.

Models	Param	CoVoST2		Fleurs		
		X-EN	EN-X	X-EN	EN-X	X-X
Offline Models						
SeamlessM4T Medium	821M	34.4	35.9	25.6	27.1	16.1
SeamlessM4T Large-v2	1.5B	**38.3**	**40.8**	**29.8**	31.5	19.6
S2T Base (Ours)	561M	37.0	38.9	26.4	**32.4**	**25.1**
Simultaneous Models						
Seamless-S2T	2.0B	35.3(-7.8%)	37.6(-7.8%)	**28.0**(-6.0%)	28.9(-8.3%)	18.1(-7.7%)
SimulMEGA-S2T (Ours)	561M	**36.9**(-0.3%)	**38.5**(-1.0%)	26.3(-0.4%)	**31.4**(-3.1%)	**24.7**(-1.7%)

Baselines In simultaneous S2TT, our method is compared against the Seamless model family [15] and four custom baselines: Wait-K [43], Dig-SST [11], EDATT [19] and AlignATT [20], all implemented using our offline base model architecture. For streaming TTS comparisons, we benchmark against the native streaming implementation in CosyVoice2. In simultaneous S2ST evaluation, Seamless serves as our primary comparator. Both the Seamless framework and our proposed system incorporate voice preservation capabilities. We also compare against two publicly available models, StreamSpeech[12] and NAST-S2S[25], whose parameter size and data size are smaller.

4.2 Main Results

Offline Multilingual S2TT Performance The results are summarized in Table 1. Leveraging high-quality pseudo-labeled data, our offline base model achieves performance comparable to SeamlessM4T large-v2 across our in-domain six languages, despite the latter having three times the parameter count. The Seamless models, constrained by an English-centric data distribution, slightly underperform ours in the FLEURS many-to-many evaluation. For simultaneous models, we report BLEU score degradation (in brackets) relative to their respective base models. Our analysis reveals that SimulMEGA exhibits minimal degradation, ranging from **0.3%** to **3%**, while Seamless suffers a more significant **7%** drop. Consequently, SimulMEGA outperforms Seamless in 4 out of the 5 evaluated scenarios.



(a) CoVoST2 X-EN(5) (b) CoVoST2 EN-X(2) (c) FLEURS X-X(30)

Figure 3: The multilingual simultaneous speech-to-text translation quality (BLEU) against the latency metrics (LAAL) on different testsets. The reported BLEU and LAAL are the average of all language splits in the testsets(5 in CoVoST X-EN, 2 in CoVoST2 EN-X and 30 in Fleurs X-X)

Simultaneous Multilingual S2TT Performance To evaluate generalizability across language pairs, we apply the same threshold configuration to all methods under identical settings. As illustrated in Figure 3, SimulMEGA demonstrates robust performance across diverse language pairs and outperforms baseline methods in all three evaluation scenarios. Compared to the offline base

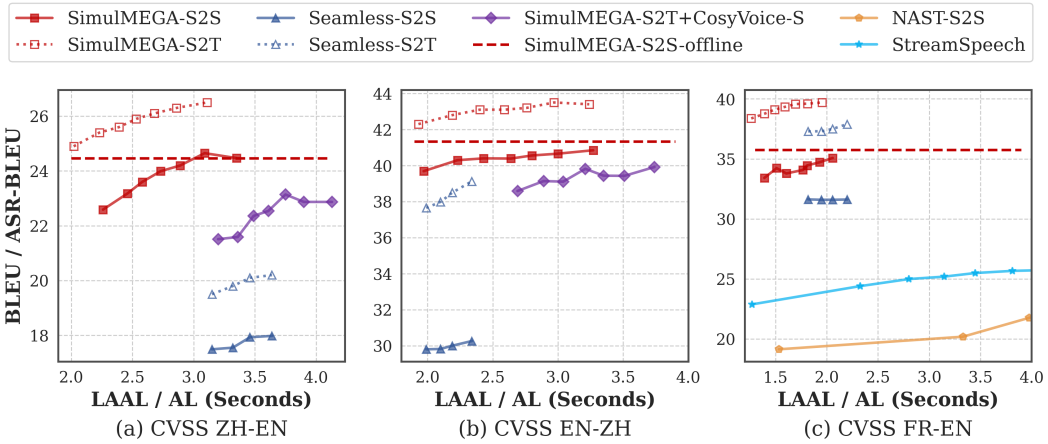
Table 2: TTS evaluation results. ZS denotes zero-shot mode (prompt added in AR stage), and CL denotes cross-lingual mode (prompt not added in AR stage). ALs are calculated by the number of input text tokens. We use the generated text as the WER label for CoVoST2 evaluation.

	LibirSpeech			SeedTTS_zh			CoVoST2_zh-en		CoVoST2_en-zh	
	WER	SIM	AL	WER	SIM	AL	WER	SIM	WER	SIM
CosyVoice2-ZS	2.44	0.658	22.3	1.62	0.757	23.1	-	-	-	-
CosyVoice2-S-ZS[2]	5.31	0.651	18.5	7.98	0.760	20.9	-	-	-	-
CosyVoice2-S-CL[2]	3.13	0.535	8.4	4.33	0.707	8.4	19.94	0.388	7.74	0.352
Seamless	-	-	-	-	-	-	20.05	0.292	18.86	0.301
SimulMEGA-TTS	2.54	0.661	1.2	1.90	0.755	0.5	12.26	0.391	4.51	0.405

model, SimulMEGA exhibits minimal degradation—**3%**, **3%**, and **5%** degradation at an **AL of 2 seconds**—while Seamless suffers significantly higher degradation (**12%**, **17%**, and **9%**). Notably, SimulMEGA achieves low-latency performance with **<7%** degradation at **1.5 seconds AL** and nears offline model quality (**<3%** degradation) at **3 seconds AL**.

In contrast, Seamless, despite its self-learning design, struggles with a substantial performance gap due to limitations in multi-head monotonic attention(MMA). The static wait-K strategy incurs high latency and degradation across all settings. Meanwhile, DiG-SST, ED-ATT and AlignATT exhibits instability, failing on specific language pairs. In our experiment ED-ATT is most stable among baselines but still shows more than 0.5s additional LAAL compare to SimulMEGA under same quality.

Streaming TTS & Simultaneous S2ST Our experiments on LibriSpeech and SeedTTS_zh evaluate SimulMEGA-TTS under extreme streaming conditions, processing one text unit at a time. For the CoVoST2 cross-lingual setting, the system receives incremental text chunks generated by SimulMEGA-S2T. We compare against CosyVoice2-S operating at its default text-to-speech ratio. As shown in Table 2, SimulMEGA-TTS demonstrates comparable speaker similarity to both CosyVoice2 variants, attributable to their shared flow model. While showing marginally lower alignment latency (AL) than CosyVoice systems, SimulMEGA-TTS maintains equivalent speech intelligibility (WER) to the original CosyVoice2. In cross-lingual evaluation, SimulMEGA-TTS surpasses Seamless by 10 SIM points while achieving more than 40% lower WER. Figure 4 presents our simultaneous speech-to-speech translation (S2ST) results. SimulMEGA-S2S maintains tight latency control, adding less than 200 ms AL compared to speech-to-text translation (S2TT). The quality degradation from S2TT to S2ST shows a 7% (ZH-EN) and 6% (EN-ZH) BLEU reduction for our system, lower than Seamless's 10% and 20% reductions, respectively. Furthermore, SimulMEGA-S2S outperforms synthesizing by CosyVoice2's streaming mode in both output quality and latency metrics.



(a) CVSS ZH-EN (b) CVSS EN-ZH (c) CVSS FR-EN

Figure 4: Bi-directional Simultaneous S2TT and S2ST Result between Mandarin and English in CoVoST2 test set. The translation quality metric is BLEU for S2TT and ASR-BLEU for S2ST. We use the same threshold group between S2ST and S2TT. Results of NAST-S2S and StreamSpeech are taken from original paper.

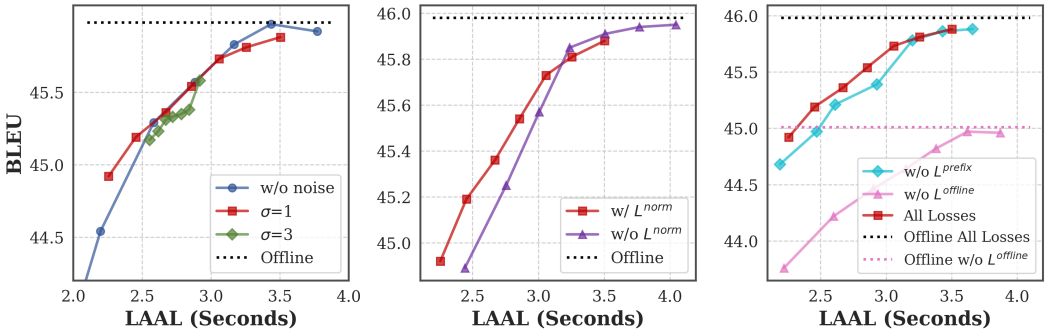
4.3 Ablation Study

We conduct ablation experiments on three key designs in SimulMEGA: 1) pre-Sigmoid noise scale, 2) router score normalization, and 3) training losses. For these ablations, we employ the same base model and conduct stage-2 simultaneous training as in the main experiment. Training is performed on a subset of the main dataset containing only the Common Voice English data, with evaluation on the CoVoST2 EN-ZH test set.

Pre-Sigmoid Noise We examine three noise settings: 1) No noise, 2) $\sigma = 1$, and 3) $\sigma = 3$. Figure 5(a) indicates that noise magnitude has minimal effect on average performance. However, increasing noise leads to more deterministic router scores, thus limiting the flexibility of the latency-performance curve. Conversely, the absence of noise results in an overly wide score range and unstable low-latency performance. A moderate noise level ($\sigma = 1$) achieves an optimal balance between a manageable dynamic range and stable translation performance.

Score Normalization Figure 5(b) shows that without normalization, router scores are disproportionately concentrated between thresholds 0.5 and 0.8, complicating threshold selection across different tasks or language pairs. Incorporating normalization yields a smoother latency-performance curve, reflecting a more balanced score distribution within the [0.2, 0.8] range, thereby facilitating easier and more consistent threshold tuning.

Training Loss We investigate the effects of removing either $\mathcal{L}^{\text{offline}}$ or $\mathcal{L}^{\text{prefix}}$ during simultaneous training. Results presented in Figure 5(c) demonstrate that removing $\mathcal{L}^{\text{offline}}$ reduces overall performance by approximately 1 BLEU point, highlighting the necessity of offline loss to maintain simultaneous translation quality. Conversely, omitting $\mathcal{L}^{\text{prefix}}$ leads to negligible performance degradation, suggesting its optional nature. We hypothesize that the generalization to prefix translation tasks provided by $\mathcal{L}^{\text{offline}}$ and $\mathcal{L}^{\text{refiner}}$ mitigates the need for truncation-specific training.



(a) Ablation on Noise Scale (b) Ablation on Score Norm (c) Ablation on Losses

Figure 5: Ablation Studies. The threshold for all ablation experiment evaluation is 0.8, 0.7, ..., 0.2 from left to right.

5 Conclusion and Discussion

SimulMEGA addresses multilingual SimulST challenges through an unsupervised policy learning framework integrating prefix-based training and Mixture-of-Experts refinement. By enabling minimal architectural modifications to standard transformers, it achieves state-of-the-art quality-latency tradeoffs across six languages while supporting unified speech/text streaming. The integrated system advances real-time cross-lingual communication with miniature BLEU degradation at low latency and robust conversational performance, establishing a versatile foundation for low-latency multimodal translation systems.

Limitation Though SimulMEGA-S2S is a strong simultaneous S2ST system, it suffers from the drawback shared by cascaded systems: inconsistent text tokens(between Whisper tokens in S2TT and

[2]Using the officially released code and checkpoint (Commit fbab274), CosyVoice 2 exhibits hallucination issues in streaming mode, leading to increased WER and deviations from reported results.

Qwen2 tokens in TTS), extra latency(around 0.1 seconds) and so on. What's more, SimulMEGA-TTS only supports two languages by now, which hinders its utility. In the future, we plan to add more language support, as well as integrate into a more unified end-to-end system. Additionally, currently SimulMEGA only support 30 seconds of maximum input duration. Therefore, it still rely on an VAD model to cut input stream into less than 30s. In the future we will explore continuously generation without segmentation by sliding window or history selection.

Broader Impacts This work proposes new streaming techniques and is dedicated to providing a faithful translation of the original speech. However, with the voice cloning capability of SimulMEGA-TTS, the system may be subject to some misuse.

6 Acknowledgment

This work was supported by China STI 2030-Major Projects under Grant No. 2021ZD0201500

References

[1] X. Ma, J. Pino, and P. Koehn, "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*, K.-F. Wong, K. Knight, and H. Wu, Eds. Suzhou, China: Association for Computational Linguistics, 2020, pp. 582–587.

[2] Q. Dong, Y. Zhu, M. Wang, and L. Li, "Learning When to Translate for Streaming Speech," 2022.

[3] S. Zhang and Y. Feng, "End-to-End Simultaneous Speech Translation with Differentiable Segmentation," 2023.

[4] J. Xue, P. Wang, J. Li, M. Post, and Y. Gaur, "Large-Scale Streaming End-to-End Speech Translation with Neural Transducers," 2022.

[5] K. Deng and P. C. Woodland, "Label-Synchronous Neural Transducer for E2E Simultaneous Speech Translation," 2024.

[6] J. Zhao, N. Moritz, E. Lakomkin, R. Xie, Z. Xiu, K. Zmolikova, Z. Ahmed, Y. Gaur, D. Le, and C. Fuegen, "Textless Streaming Speech-to-Speech Translation using Semantic Speech Tokens," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[7] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *International conference on machine learning*. PMLR, 2017, pp. 2837–2846.

[8] X. Ma, J. Pino, J. Cross, L. Puzon, and J. Gu, "Monotonic multihead attention," *arXiv preprint arXiv:1909.12406*, 2019.

[9] S. Guo, S. Zhang, and Y. Feng, "Learning Optimal Policy for Simultaneous Machine Translation via Binary Search," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 2318–2333.

[10] A. Yin, T. Zhong, H. Li, S. Tang, and Z. Zhao, "Language Model is a Branch Predictor for Simultaneous Machine Translation," 2023.

[11] X. Chen, K. Fan, W. Luo, L. Zhang, L. Zhao, X. Liu, and Z. Huang, "Divergence-Guided Simultaneous Speech Translation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17 799–17 807, 2024.

[12] S. Zhang, Q. Fang, S. Guo, Z. Ma, M. Zhang, and Y. Feng, "StreamSpeech: Simultaneous Speech-to-Speech Translation with Multi-task Learning," 2024.

[13] L. Zhao, J. Li, and Z. Zeng, "PsFuture: A Pseudo-Future-based Zero-Shot Adaptive Policy for Simultaneous Machine Translation," 2024.

[14] S. Guo, S. Zhang, Z. Ma, and Y. Feng, "Large Language Models Are Read/Write Policy-Makers for Simultaneous Generation," 2025.

[15] S. Communication, L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, M. Duppenthaler, P.-A. Duquenne, B. Ellis, H. Elsahar, J. Haaheim, J. Hoffman, M.-J. Hwang, H. Inaguma, C. Klaiber, I. Kulikov, P. Li, D. Licht, J. Maillard, R. Mavlyutov, A. Rakotoarison, K. R. Sadagopan, A. Ramakrishnan, T. Tran, G. Wenzek, Y. Yang, E. Ye, I. Evtimov, P. Fernandez, C. Gao, P. Hansanti, E. Kalbassi, A. Kallet, A. Kozhevnikov, G. M. Gonzalez, R. S. Roman, C. Touret, C. Wong, C. Wood, B. Yu, P. Andrews, C. Balioglu, P.-J. Chen, M. R. Costa-jussà, M. Elbayad, H. Gong, F. Guzmán, K. Heffernan, S. Jain, J. Kao,

A. Lee, X. Ma, A. Mourachko, B. Peloquin, J. Pino, S. Popuri, C. Ropers, S. Saleem, H. Schwenk, A. Sun, P. Tomasello, C. Wang, J. Wang, S. Wang, and M. Williamson, "Seamless: Multilingual Expressive and Streaming Speech Translation," 2023.

[16] Z. Du, Y. Wang, Q. Chen, X. Shi, X. Lv, T. Zhao, Z. Gao, Y. Yang, C. Gao, H. Wang, F. Yu, H. Liu, Z. Sheng, Y. Gu, C. Deng, W. Wang, S. Zhang, Z. Yan, and J. Zhou, "CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models," 2024.

[17] X. Zeng, L. Li, and Q. Liu, "RealTranS: End-to-End Simultaneous Speech Translation with Convolutional Weighted-Shrinking Transformer," 2021.

[18] S. Zhang and Y. Feng, "Information-Transport-based Policy for Simultaneous Translation," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 992–1013.

[19] S. Papi, M. Negri, and M. Turchi, "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)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 13 340–13 356.

[20] S. Papi, M. Turchi, and M. Negri, "AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation," in *INTERSPEECH 2023*, 2023, pp. 3974–3978.

[21] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," 2006.

[22] N. Arivazhagan, C. Cherry, W. Macherey, C.-C. Chiu, S. Yavuz, R. Pang, W. Li, and C. Raffel, "Monotonic infinite lookback attention for simultaneous machine translation," *arXiv preprint arXiv:1906.05218*, 2019.

[23] X. Ma, A. Sun, S. Ouyang, H. Inaguma, and P. Tomasello, "Efficient Monotonic Multihead Attention," 2023.

[24] S. Zhang and Y. Feng, "Hidden Markov Transformer for Simultaneous Machine Translation," in *The Eleventh International Conference on Learning Representations*, 2022.

[25] Z. Ma, Q. Fang, S. Zhang, S. Guo, Y. Feng, and M. Zhang, "A Non-autoregressive Generation Framework for End-to-End Simultaneous Speech-to-Any Translation," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 1557–1575.

[26] M. Łajszczak, G. Cámbara, Y. Li, F. Beyhan, A. van Korlaar, F. Yang, A. Joly, Á. Martín-Cortinas, A. Abbas, A. Michalski, A. Moinet, S. Karlapati, E. Muszyńska, H. Guo, B. Putrycz, S. L. Gambino, K. Yoo, E. Sokolova, and T. Drugman, "BASE TTS: Lessons from building a billion-parameter Text-to-Speech model on 100K hours of data," 2024.

[27] T. Dang, D. Aponte, D. Tran, and K. Koishida. (2024-06-10) LiveSpeech: Low-Latency Zero-shot Text-to-Speech via Autoregressive Modeling of Audio Discrete Codes. [Online]. Available: http://arxiv.org/abs/2406.02897

[28] A. Dekel, S. Shechtman, R. Fernandez, D. Haws, Z. Kons, and R. Hoory, "Speak While You Think: Streaming Speech Synthesis During Text Generation," 2023.

[29] Z. Sheng, Z. Du, S. Zhang, Z. Yan, Y. Yang, and Z. Ling, "SyncSpeech: Low-Latency and Efficient Dual-Stream Text-to-Speech based on Temporal Masked Transformer," 2025.

[30] T. Dang, D. Aponte, D. Tran, T. Chen, and K. Koishida, "Zero-Shot Text-to-Speech from Continuous Text Streams," 2024.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[32] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.

[33] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

[34] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[35] B. Zhang, H. Lv, P. Guo, Q. Shao, C. Yang, L. Xie, X. Xu, H. Bu, X. Chen, C. Zeng *et al.*, "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6182–6186.

[36] Z. Tang, D. Wang, Y. Xu, J. Sun, X. Lei, S. Zhao, C. Wen, X. Tan, C. Xie, S. Zhou *et al.*, "Kespeech: An open source speech dataset of mandarin and its eight subdialects," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[37] H. He, Z. Shang, C. Wang, X. Li, Y. Gu, H. Hua, L. Liu, C. Yang, J. Li, P. Shi, Y. Wang, K. Chen, P. Zhang, and Z. Wu, "Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation," in *Proc. of SLT*, 2024.

[38] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[39] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[41] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 186–191. [Online]. Available: https://www.aclweb.org/anthology/W18-6319

[42] X. Ma, M. J. Dousti, C. Wang, J. Gu, and J. Pino, "Simuleval: An evaluation toolkit for simultaneous translation," in *Proceedings of the EMNLP*, 2020.

[43] M. Ma, L. Huang, H. Xiong, R. Zheng, K. Liu, B. Zheng, C. Zhang, Z. He, H. Liu, X. Li, H. Wu, and H. Wang, "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*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, 2019, pp. 3025–3036.

[44] C. Wang, A. Wu, J. Gu, and J. Pino, "Covost 2 and massively multilingual speech translation." in *Interspeech*, 2021, pp. 2247–2251.

[45] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 798–805.

[46] A. Meister, M. Novikov, N. Karpov, E. Bakhturina, V. Lavrukhin, and B. Ginsburg, "Librispeech-pc: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end asr models," in *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 2023, pp. 1–7.

[47] P. Anastassiou, J. Chen, J. Chen, Y. Chen, Z. Chen, Z. Chen, J. Cong, L. Deng, C. Ding, L. Gao, M. Gong, P. Huang, Q. Huang, Z. Huang, Y. Huo, D. Jia, C. Li, F. Li, H. Li, J. Li, X. Li, X. Li, L. Liu, S. Liu, S. Liu, X. Liu, Y. Liu, Z. Liu, L. Lu, J. Pan, X. Wang, Y. Wang, Y. Wang, Z. Wei, J. Wu, C. Yao, Y. Yang, Y. Yi, J. Zhang, Q. Zhang, S. Zhang, W. Zhang, Y. Zhang, Z. Zhao, D. Zhong, and X. Zhuang, "Seed-tts: A family of high-quality versatile speech generation models," 2024. [Online]. Available: https://arxiv.org/abs/2406.02430

[48] Z. Gao, S. Zhang, I. McLoughlin, and Z. Yan, "Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition," *arXiv preprint arXiv:2206.08317*, 2022.

[49] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

A SimulMEGA-TTS Implementation Details

Figure 6 shows how to implement SimulMEGA on top of a uni-directional decoder-only backbone. We take CosyVoice 2 as the base model and fine-tune it into SimulMEGA-TTS. We take the last layer hidden states output of text tokens as the h^{prefix} and the hidden state of the text EOS token as H^{global}.

In S2TT, normally, when the input reaches the end of the stream, we no longer turn to the read/write strategy and instead continue the generation until the EOS token appears. However, in the TTS task, the end of speech is relatively ambiguous. Therefore, if we employ the same strategy as S2TT, the hallucination problem may occur at the end of the generation. Therefore, we employ a mixed strategy to tell the end of the generation:

$$\text{End of Generation} = \begin{cases} \text{True} & \text{if } E_{\text{t}} \in T_{\text{text}} \text{ and } (E_s \in T_{\text{speech}} \text{ or } p_{\cdot,i} > \lambda_{\text{end}}) \\ \text{False} & \text{otherwise} \end{cases} \tag{9}$$

Where $E_{\text{t/s}}$ denotes text/speech EOS token. $T_{\text{text/speech}}$ denotes text or speech tokens in the current sequence. $p_{\cdot,i}$ is the router output at the current position after all input streams are read. $\lambda_{\text{end}} = 0.9$.

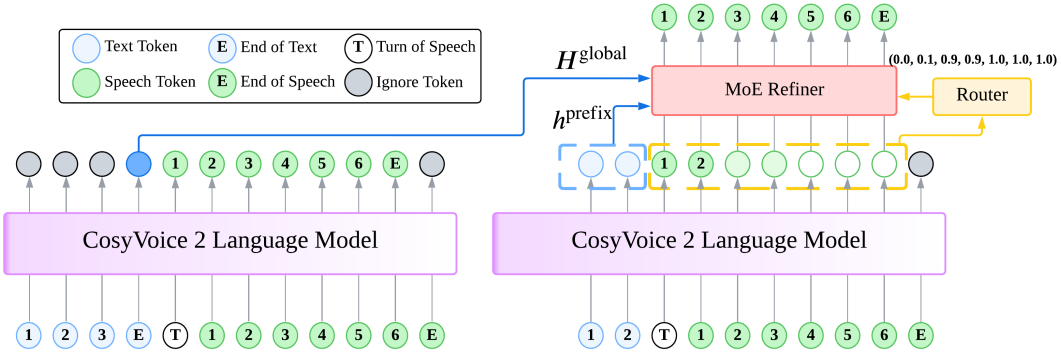


Figure 6: Illustration of SimulMEGA-TTS. The left uses offline text input while the right uses prefix text input for policy learning.

B Experiment Details

B.1 S2TT Baselines

1. **Meta Seamless series:** Meta Seamless series contains a set of models including SeamlessM4T-large-v2 for offline translation, SeamlessStreaming for simultaneous translation and SeamlessExpressive for voice cloning. They also combines the capability of SeamlessStreaming with SeamlessExpressive into a single model named Seamless. In our experiment we use SeamlessM4T-large-v2 for offline evaluation, SeamlessStreaming for simultaneous S2TT evaluation and Seamless for simultaneous S2ST evaluation.
2. **Wait-K:** For each input test sample, We use a VAD to determine the starting time. Then we wait for two chunks (1.28s) and start generation at a fixed ratio, ranging from 0.5 to 2.5 tokens per chunk.
3. **DiG-SST:** Following the original paper, we freeze the base model and add three learnable policy layers on top of the text decoder to predict the divergence between prefix and global input. It was trained for 100k steps on the same training set of SimulMEGA.
4. **EDATT:** We follow the paper and official code[5] for implementation. We re-tuned the thresholds and hyperparameter and set attention layer to be the 8th layer and set attention frame to be 4.
5. **AlignATT:** We follow the updated official code[6]. A frame-wise normalization is applied to the attention score to mitigate maximum attention fixation.

In our experiment we observed that both DiG-SST and AlignATT behave poorly on all X2ZH pairs, which hinders the overall performance. We hypothesis that this might be caused by the UTF-8 byte-level BPE tokenizer. It encodes each CJK character into multiple tokens, potentially leading to unexpected output distribution and attention pattern.

[5] https://github.com/hlt-mt/FBK-fairseq/blob/master/examples/speech_to_text/simultaneous_translation/agents/v1_1/simul_offline_edatt.py

[6] https://github.com/hlt-mt/FBK-fairseq/blob/master/examples/speech_to_text/simultaneous_translation/agents/v1_1/simul_alignatt_seamlessm4t.py

B.2 Quality-latency variable

Table 3 shows the choice of the variable that controls the quality-latency tradeoff in the main experiment. From left to right the quality and latency gradually increase.

Table 3: Quality-latency variables in the main experiment.

Method	variable	S2TT	S2ST
SimulMEGA	λ	[0.9, 0.7, 0.5, 0.3, 0.1]	[0.8, 0.7,... , 0.2]
Seamless	t_{EMMA}	[0.3, 0.5, 0.7, 0.9, 1]	[0.3, 0.5, 0.7, 0.9]
DiG-SST	λ	[0.1, 0.07, 0.05, 0.03, 0.01]	-
EDATT	α	[0.05, 0.03, 0.02, 0.015, 0.01]	-
AlignATT	f	[32, 48, 64, 80, 96]	-
Wait-K	ratio	[2.5, 2.0, 1.5, 1.0, 0.5]	-

C S2TT computation overhead.

For evaluating the computation overhead, we plot the computation-aware(CL) quality-latency trade-off curve in the CoVoST2 FR-EN testset. Computation-aware latency considers the actual inference time of the model. The evaluation is performed on a fully idle machine with a single Nvidia-H100 GPU. As shown in Figure 7, for SimulMEGA, the extra AL due to computation is around 50 ms, whereas the number for Seamless is around 200 ms.

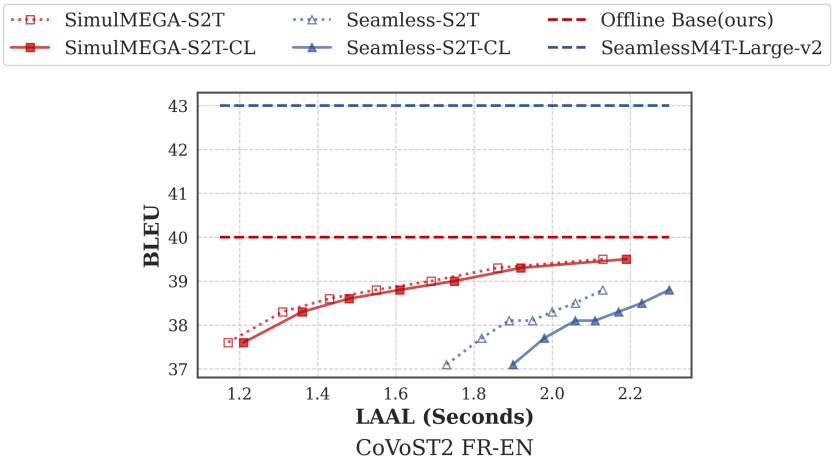


Figure 7: Computation aware quality-latency curve on CoVoST2 FR-EN. CL denotes Computation-Aware.

D S2ST System Deployment & Latency.

We deploy our S2ST system on an Ubuntu server equipped with NVIDIA H100 GPUs. For the frontend and backend, we utilize FastRTC[7] and FastAPI[8], respectively. To enhance inference efficiency, vLLM[9] is used to accelerate the language model in TTS, and TensorRT[10] is used to accelerate the flow model.

We evaluate the system latency with streaming inputs, using four language pairs (ZH-EN, EN-ZH, FR-EN, FR-ZH). For each direction, we test two samples, each 20–30 seconds in length, and record event timestamps for latency calculation. Virtual Audio Cable[11] software is employed to eliminate echo interference and ensure

[7] https://github.com/gradio-app/fastrtc
[8] https://github.com/fastapi/fastapi
[9] https://github.com/vllm-project/vllm
[10] https://github.com/NVIDIA/TensorRT
[11] https://vac.muzychenko.net/en/

consistent audio input across tests. The average latency results are presented in Table 4. Here, the S2T/S2S Start Offset indicates the time required to generate the first text or speech chunk, while the S2T/S2S End Offset measures the delay of the final text chunk or speech frame relative to the end of the source speech.

It should be noted that these results provide only a rough estimation of system latency, as actual values may vary due to server or network conditions, output content length, speech rate, or other factors.

Table 4: Simultaneous S2ST system latency statistics (in seconds). N denotes using non-stream TTS and S denotes stream TTS

		Offline	SimulMEGA		Seamless
		N	N	S	-
S2TT	Start Offset	13.56	3.12	2.95	3.55
	End Offset	0.78	0.44	0.58	0.61
S2ST	Start Offset	15.98	7.43	3.87	4.33
	End Offset	14.16	7.65	7.54	6.42

E Numerical results

Numerical results of simulMEGA S2TT on CoVoST 2 testset is shown in Table 5.

Table 5: Numerical results of simulMEGA S2TT on CoVoST 2 testset.

Threshold		**0.9**	**0.7**	**0.5**	**0.3**	**0.1**	**0**
zh-en	BLEU	21.06	25.37	26.01	26.40	26.63	26.89
	LAAL	1.395	2.249	2.539	2.807	3.180	-
de-en	BLEU	32.90	36.08	36.94	37.15	37.41	37.44
	LAAL	1.217	1.862	2.149	2.420	2.856	-
es-en	BLEU	37.21	40.27	40.94	41.28	41.46	41.51
	LAAL	1.047	1.521	1.760	2.001	2.405	-
fr-en	BLEU	36.50	38.76	39.33	39.60	39.76	40.02
	LAAL	0.952	1.391	1.590	1.812	2.195	-
it-en	BLEU	34.64	37.42	38.12	38.57	38.89	38.90
	LAAL	1.101	1.621	1.873	2.125	2.512	-
en-zh	BLEU	39.93	42.77	43.11	43.17	43.42	43.57
	LAAL	1.539	2.305	2.680	2.976	3.381	-
en-de	BLEU	29.65	32.89	33.16	33.61	33.66	33.75
	LAAL	1.266	1.813	2.121	2.404	2.808	-

F S2ST Case Study

We visualize two S2ST example of SimulMEGA in FR-EN and ZH-EN, respectively. The second row is the isochronic text label of source speech and the forth row indicates the timestamps when each target text chunk are generated.

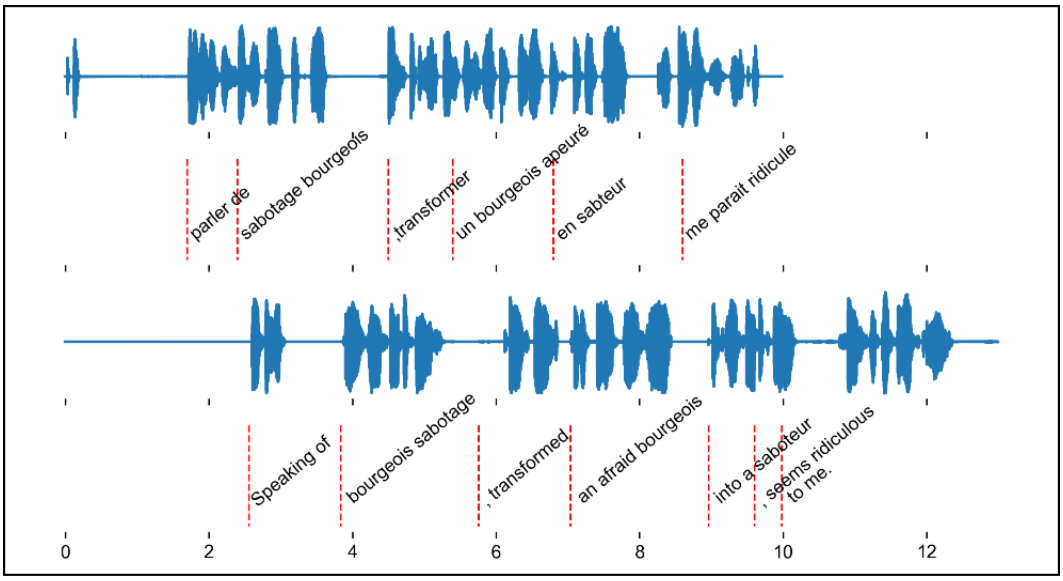


Figure 8: (a) FR-EN

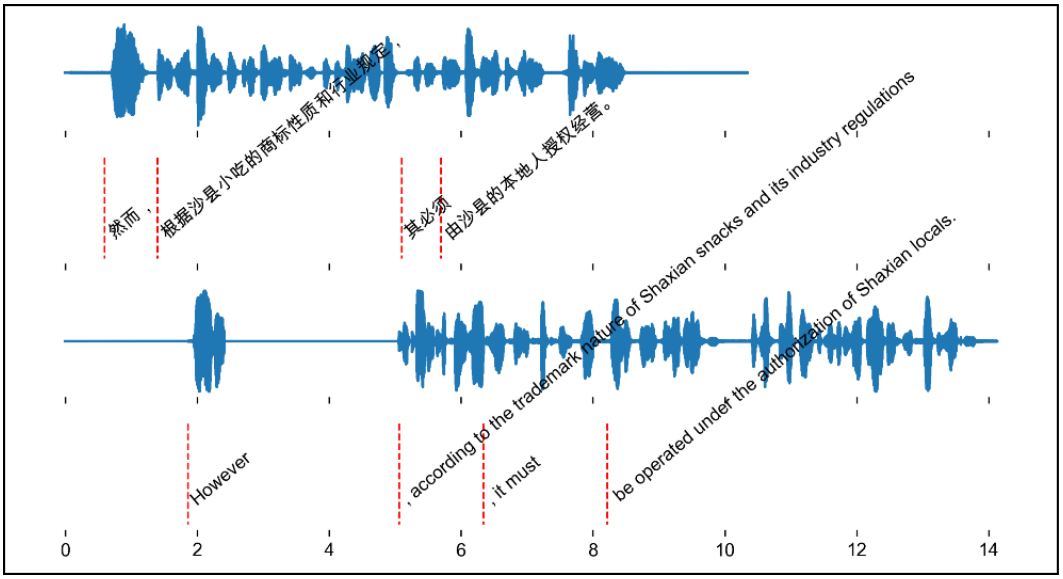


Figure 8: (b) ZH-EN

.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are clearly stated.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have included the discussion on the limitation in the last section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have disclosed full details to reproduce the result in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All ASR data we use is open-sourced, and there are various ways to pseudo label it(Perhaps with LLMs). The code will soon be opensourced on Github.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have included most of the details. The remaining can be found in the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Reporting statistical significance is not a common practice for our setting due the heavy training and evaluation overhead. However, we have tested our method on two tasks, two backbones, and more than 40 language pairs, which to some extent statistically proves the effectiveness of our method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We reports the device and execution time in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm we follow every respect of the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have reported that the voice cloning part may subject to misuse.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The risk of misuse is not high. Normally, people will turn to other dedicated TTS models for faking a voice.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit all assets we use and respect all licenses of these assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Currently, we haven't published any new assets. And in the future, we will document it.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The work does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.