CRESiST: Self-Enhancing Explorative Policy Learning for Simultaneous Speech Translation

Chenyang Le[1*] **Pei Zhang**[2,3] **Yuxiang Chen**[4] **Bing Han**[1,2]
Derek F. Wong[3] **Baosong Yang**[2] **Yanmin Qian**[1†]

[1]Shanghai Jiao Tong University [2]Alibaba Token Hub, Alibaba Group
[3]University of Macau [4]University College London

nethermanpro@sjtu.edu.cn yanmin.qian@sjtu.edu.cn

Abstract

Simultaneous Speech Translation (SimulST) requires balancing translation quality and latency. While Large Language Models (LLMs) excel in offline translation, current LLM-based SimulST systems predominantly rely on prefix-incremental generation. This paradigm suffers from redundant Key-Value cache recomputation and necessitates external Voice Activity Detection to process unbounded streams. To overcome these bottlenecks, we propose an end-to-end interleaved read-write framework built upon an omni-modal LLM. By interleaving speech chunks and text tokens, our architecture achieves full KV cache reuse, natively supporting segmentation-free, infinite-length inference. Furthermore, to learn optimal streaming policies without rigid offline alignments, we introduce a novel Actor-Critic self-enhancement pipeline. An RNN-T Actor dynamically samples read-write trajectories, while an LLM-based Critic evaluates these paths via intrinsic semantic and latency rewards, distilling the optimal policy back to the Actor. Extensive experiments on sentence-level (FLEURS, CoVoST2) and document-level (MuST-C, ACL 60/60) benchmarks demonstrate that our model establishes new state-of-the-art quality-latency Pareto frontiers, substantially outperforming existing cascaded and static-alignment baselines.

1 Introduction

Simultaneous Speech Translation (SimulST) aims to incrementally translate a continuous audio stream while minimizing delivery latency. Although Large Language Models (LLMs) have revolutionized offline translation, deploying them for unbounded real-time speech reveals a fundamental architectural conflict. State-of-the-art SimulST approaches [1, 2, 3, 4, 5] predominantly rely on prefix-incremental generation [6], where newly arrived speech chunks are continuously appended to the historical prefix. This invalidates the Key-Value (KV) cache across incremental steps, causing a quadratic computational explosion and forcing systems to rely on external Segmentation, such as Voice Activity Detection (VAD). Such forced segmentation disrupts cross-sentence context, impairs coreference resolution, and fails to natively support infinite-length streams.

To bypass the KV cache bottleneck, recent studies explored interleaved read-write architectures [7, 8] that append acoustic and textual tokens into a unified sequence. However, these models heavily depend on static, heuristic-driven word alignments (e.g., MFA[9]) to construct supervised trajectories, failing to capture the dynamic, predictive nature of human interpreters. While some works adopt reinforcement learning [10, 11, 12, 13], they still require offline aligned data and struggle to scale

*Work done during an internship at Alibaba Group.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

multilingually. To address this, we propose **CRESiST** (**C**ritic-**R**efined **E**xploratory **Si**multaneous **S**peech **T**ranslation), an interleaved framework that discovers its streaming policy without any external alignment supervision. Built on the Qwen2.5-omni-7B backbone [14], CRESiST discards prefix-incremental generation and segmentation, achieving full KV cache reuse. Crucially, it operates at a granular 400ms micro-chunk size, enabling a highly fine-grained streaming policy.

To optimize this fine-grained policy without external alignments, we formulate the streaming decision process as a Markov Decision Process solved via a self-enhancing Actor-Critic pipeline. An auxiliary RNN-T acts as the Actor, actively exploring read-write trajectories. Simultaneously, the Speech LLM, equipped with an MoE Router and Refiner, serves as the Critic. The Critic evaluates trajectories using dynamic consistency and latency rewards, transferring its reward-maximizing preferences back to the Actor via reverse distillation. This iterative self-play allows CRESiST to autonomously discover adaptive, language-specific translation strategies (e.g., anticipating German verbs). Consequently, a single unified CRESiST model seamlessly translates Chinese and English streams into six diverse target languages.

Our main contributions are summarized as follows:

- **Unsupervised Streaming Policy in an Interleaved Framework:** While trained purely on offline parallel data, our framework autonomously transitions to an online interleaved paradigm, enabling full KV cache reuse for segmentation-free, infinite-length translation.
- **Fine-Grained Actor-Critic Policy Discovery:** We introduce a novel self-enhancement pipeline operating at a 400ms acoustic chunk size. Through reward-driven reverse distillation, the model autonomously learns optimal streaming policies without static forced alignments.
- **Multilingual Generalization:** A single unified CRESiST model supports Chinese/English inputs to six target languages, overcoming the limitations of language-pair-specific models.
- **State-of-the-Art Performance:** Extensive experiments across sentence-level and unbounded document-level benchmarks demonstrate that CRESiST outperforms strong baselines (e.g., InfiniSST), establishing new Quality-Latency Pareto frontiers.

2 Related Work

Read-Write Policy Learning Determining the optimal timing for translation is a core challenge in SimulST. A prevailing paradigm relies on static alignments from external tools (e.g., MFA) or probability mutations to construct supervised read-write trajectories [7, 15, 16, 8, 17, 18]. To circumvent the rigidity of forced alignments, fixed and heuristic approaches utilize continuous integrate-and-fire (CIF)[19], wait-k cascades, or attention boundary indicators [20, 21, 22, 23]. Recent unsupervised methods further explore dynamic boundary detection by leveraging offline-to-online inference divergence [4, 5], dual CIF distillation [24], or implicit learning via mixture-of-experts [3]. Furthermore, to optimize the latency-quality trade-off beyond static supervision, advanced reinforcement learning techniques (such as PPO and GRPO) have been increasingly adopted to autonomously refine streaming translation strategies [11, 13, 12].

Unbounded Audio Processing Processing continuous, infinite-length audio streams typically suffers from severe memory bottlenecks. Traditional systems heavily depend on external segmenters like VAD or acoustic heuristics [25], which inevitably disrupt cross-sentence contextual coherence. To mitigate this, recent studies have proposed history selection mechanisms that only retain fixed amounts of text tokens and their attention-covered acoustic frames [26]. Alternatively, formulating the continuous stream as multi-turn dialogues allows for the safe discarding of distant KV cache [7]; however, such formulations typically mandate tightly coupled supervised policies to function effectively.

3 CRESiST

3.1 Problem Formalization

Unlike previous RL approaches in SimulST that couple translation timing with content generation—creating an intractable policy space—we decouple timing by formalizing the read-write

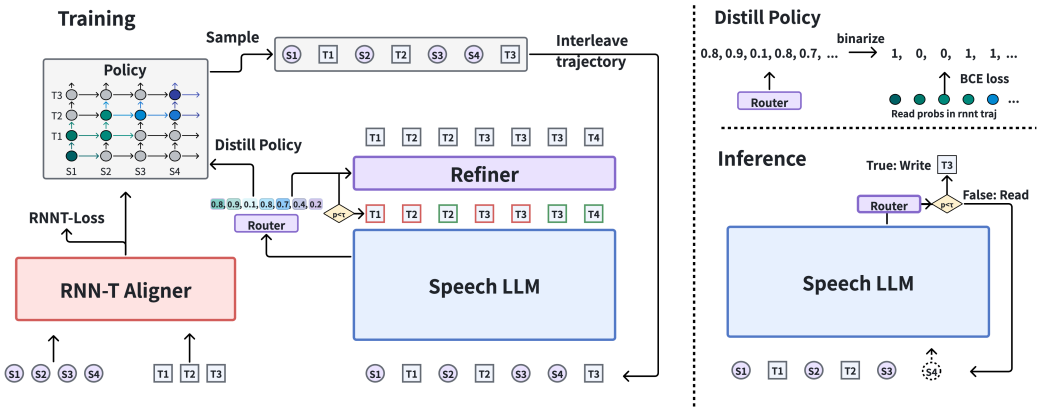


Figure 1: Training and inference diagram of CRESiST. In the interleaved sequence, red and green target tokens mark whether the Router score at that position passes the decision threshold, i.e., whether the LLM body loss $\mathcal{L}_{body}$ is computed there (tokens judged to be emitted too early are masked out). The Refiner loss $\mathcal{L}_{refiner}$ is computed at every text position, and the two losses are summed; the same ground-truth token is thus the target of two prediction heads over one sequence, not duplicated in it.

strategy as a Markov Decision Process (MDP). Given an offline speech-text pair $\{\mathbf{S}, \mathbf{T}\}$ with acoustic chunks $\mathbf{S} = (s_1, \ldots, s_{|\mathbf{S}|})$ and ground-truth text tokens $\mathbf{T} = (t_1, \ldots, t_{|\mathbf{T}|})$, the MDP is defined by the quadruple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. Specifically, the **state space ($\mathcal{S}$)** at step k is $s_k = (\mathbf{S}_{\leq u_k}, \mathbf{T}_{<v_k})$, representing the u_k speech chunks read and v_k target tokens written thus far. The **action space ($\mathcal{A}$)** is a binary space $\{\text{Read}, \text{Write}\}$ controlling the translation rhythm. The **transition dynamics ($\mathcal{P}$)** dictates the environment evolution $P(s_{k+1} \mid s_k, a_k)$: a *Read* action appends the next speech chunk yielding $s_{k+1} = (\mathbf{S}_{\leq u_k+1}, \mathbf{T}_{<v_k})$, whereas a *Write* action appends a text token to yield $s_{k+1} = (\mathbf{S}_{\leq u_k}, \mathbf{T}_{<v_k+1})$. Optimal write actions maximize congruence between **Data Dynamics** (ground-truth distribution $P_{data}(t_{v_k} \mid s_k)$) and **Model Dynamics** (LLM prediction $P_{model}(\cdot \mid s_k; \theta)$). Finally, the **reward function ($\mathcal{R}$)** provides a scalar $r_k(s_k, a_k)$ to maximize expected cumulative rewards, delicately balancing a *dynamic consistency evaluation* (aligning the Model and Data dynamics) against a *latency penalty* to ensure timely responses.

3.2 Actor-Critic Training Framework

To solve the MDP formulated in the previous section and transition toward an autonomous read-write interleaved paradigm, we propose a self-enhancing Actor-Critic training framework, outlined in Algorithm 1. Given an offline speech-text corpus $\mathcal{D} = \{(\mathbf{S}, \mathbf{T})\}$, our framework jointly optimizes translation fidelity and the simultaneous streaming policy. By decoupling the generation process into an exploration-evaluation loop, we effectively compute the MDP transition dynamics and the reward function $\mathcal{R}$ without relying on heuristic alignments.

The Actor for Trajectory Exploration. We introduce an auxiliary RNN-T[27] model parameterized by θ_{act} to serve as the Actor. Operating within the state space $\mathcal{S}$, the Actor explores the action space $\mathcal{A}$ by sampling a sequence of binary Read/Write decisions. For a given sample $(\mathbf{S}, \mathbf{T}) \sim \mathcal{D}$, it yields a trajectory of actions $\mathbf{a} \sim \pi_{act}(\cdot \mid \mathbf{S}, \mathbf{T}; \theta_{act})$, where $a_k \in \mathcal{V} \cup \{\phi\}$ (with ϕ denoting a "Read" action). This sampled trajectory simulates the transition dynamics $\mathcal{P}$, deterministically merging the source speech and target text into a unified, interleaved sequence $\mathbf{z} = \text{Interleave}(\mathbf{S}, \mathbf{T}, \mathbf{a})$. Through this exploration, the Actor constructs the streaming-style state transitions required by the downstream environment.

The Critic for Reward Evaluation and Validation. The Qwen-omni LLM backbone, equipped with a Router-Refiner module and parameterized by θ_{cri}, acts as the Critic. It ingests the state sequence $\mathbf{z}$ to evaluate the Actor's trajectory by computing the Reward Function ($\mathcal{R}$). The evaluation is twofold:

1) **Dynamic Consistency Reward ($\mathcal{R}_{consistency}$):** The Refiner and LLM body assess the congruence between Data Dynamics and Model Dynamics. By conditioning on local hidden states $\mathbf{H} = \text{LLM}(\mathbf{z}; \theta_{cri})$ and a globally compressed acoustic representation $\mathbf{h}_{global}$, it computes the token generation probabilities. We formally define the consistency reward as the negative log-likelihood of the generation targets:

$$\mathcal{R}_{consistency} = -(\mathcal{L}_{refiner} + \mathcal{L}_{body}) \tag{1}$$

where $\mathcal{L}_{refiner}$ and $\mathcal{L}_{body}$ are the cross-entropy generation losses of the Refiner and LLM body, respectively. High predictive accuracy yields a high consistency reward.

2) **Latency Reward ($\mathcal{R}_{latency}$):** To ensure timely responses without relying on heuristic constraints, the Router module assesses the sequence of hidden states to predict the action probability $p_k = \text{Router}(\mathbf{H}_{k-1}; \theta_{cri})$ at each step, where $p_k \to 1$ signifies a ***Read*** and $p_k \to 0$ signifies a ***Write***. We mathematically formalize the latency reward as the expected length of consecutive text generation in a given turn. Since the probability of executing at least m consecutive writes starting from step k is given by the cumulative product of write probabilities, the expected write length $\mathbb{E}[W_k]$ can be elegantly formulated as:

$$\mathcal{R}_{latency}^{(k)} = \mathbb{E}[W_k] = \sum_{m=1}^{M} \prod_{j=1}^{m} \left(1 - p_{k+j}\right) \tag{2}$$

where M denotes the maximum allowable length for a single conversation turn. By maximizing this expectation-based reward, the Critic intrinsically incentivizes the policy to emit longer textual chunks as early as possible, thereby minimizing the overall translation delay.

Essentially, the Critic judges whether the state-action path traversed by the Actor maximizes the overall reward $\mathcal{R} = \mathcal{R}_{consistency} + \gamma \mathcal{R}_{latency}$, facilitating fluent generation with minimal delay.

Self-Enhancement via Reward-Driven Distillation. The entire framework is optimized through an alternating closed-loop mechanism. First, the Critic is updated by minimizing the composite cost $\mathcal{J}_{critic}$, which integrates the negative rewards alongside an auxiliary load balancing loss ($\mathcal{L}_{balance}$) to stabilize the routing distribution:

$$\mathcal{J}_{critic} = -\mathcal{R}_{consistency} - \alpha \mathcal{R}_{latency} + \beta \mathcal{L}_{balance} \tag{3}$$

where α and β scale the latency reward objective and the balancing penalty, respectively.

Subsequently, the Actor is updated by minimizing its own composite objective $\mathcal{J}_{actor}$. To emphasize the reward-based policy feedback, we position a novel ***Reverse Distillation*** loss ($\mathcal{L}_{distill}$) as the primary optimization target, treating the standard RNN-T loss ($\mathcal{L}_{rnnt}$) as an auxiliary constraint to maintain basic acoustic-linguistic alignment:

$$\mathcal{J}_{actor} = \mathcal{L}_{distill} + \lambda \mathcal{L}_{rnnt} \tag{4}$$

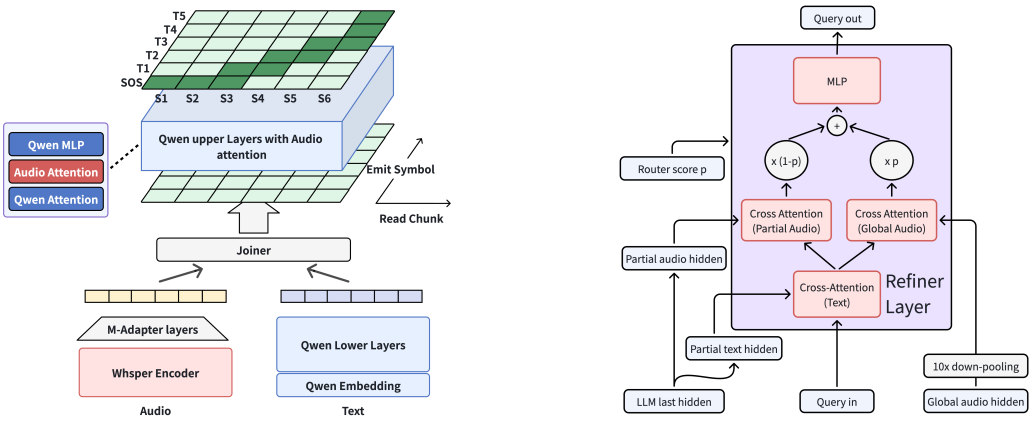
where λ acts as a down-scaling coefficient for the auxiliary loss. To compute the critical $\mathcal{L}_{distill}$, we use the reward-optimized Critic to binarize the Router's validated policy decisions via a predefined threshold τ, constructing an optimal target action sequence $\hat{a}_k = \mathbb{I}(p_k \geq \tau)$. A Binary Cross-Entropy (BCE) distillation loss then forces the Actor to align its action probabilities with these LLM-preferred, reward-maximizing decisions:

$$\mathcal{L}_{distill} = \sum_{k} \text{BCE}\Big(\hat{a}_k, \pi_{act}(a_k = \phi \mid \cdot; \theta_{act})\Big) \tag{5}$$

By continuously backpropagating gradients $\nabla_{\theta_{act}} \mathcal{J}_{actor}$, the RNN-T learns to autonomously sample trajectories that are intrinsically aligned with the LLM's optimal dynamics. We note that the pseudo-label $\hat{a}_k$ is treated with a stop-gradient on θ_{cri}, decoupling the Actor update from gradient flow through the Critic. We run K = 8 Critic steps followed by one Actor step for better convergence. Theoretical guarantees regarding the convergence of this joint optimization are provided in Appendix B.

3.3 Model Implementation

To instantiate the Actor-Critic framework, the CRESiST architecture is bifurcated into two structural components: an RNN-T-based Actor for alignment exploration and an LLM-based Critic for validation and generation.



(a) RNN-T Aligner (b) Refiner Layer

Figure 2: Structural details of the RNN-T aligner and the Refiner layer.

3.3.1 The Actor Model: Alignment Explorer

To avoid the computational inefficiency and "cold start" issues of rolling out interleaved sequences directly via the Speech LLM, the Actor is parameterized using an auxiliary Recurrent Neural Network Transducer (RNN-T).

As illustrated in Figure 2a, our Actor comprises an acoustic encoder, a text predictor, and a joint network, augmented with a 2D axial attention module. Each layer of this axial attention module consists of two causal self-attention mechanisms—operating sequentially along the text and speech axes—followed by an MLP, strengthening dual-axis modality interaction.

For the specific configuration, the acoustic encoder is initialized with the pre-trained Whisper-medium encoder[28], followed by two down-sampling M-Adapter layers to yield 2.5 Hz speech features (a 400ms window). A 400ms chunk-autoregressive mask is applied to strictly preserve causality. The text predictor and the axial attention module are initialized using the Qwen3-0.6B model, while the newly introduced audio-axis attention weights are initialized from scratch.

Read-Write Trajectory Construction. To construct the interleaved sequence $\mathbf{z}$ defined in Section 3.2, we traverse the $(T \times U)$ alignment lattice. At each node (t, u), the joint network outputs a blank ("Read") emission probability $p_y^{(t,u)}$. The routing action a_k is then directly sampled via:

$$a_k \sim \text{Bernoulli}\big(p_y^{(t_k,u_k)}\big) \tag{6}$$

The sampled path $\mathbf{a} = (a_1, \ldots, a_K)$ governs the lattice coordinate updates and dictates the exact merge boundaries for the LLM input space. Note that every sampled trajectory is a monotonic path on the lattice from $(0, 0)$ to (T, U): a blank advances only along the audio axis and a token emission only along the text axis, and once the last audio chunk is reached all remaining target tokens are emitted. Every trajectory therefore contains exactly U Write actions, so no target token is ever dropped from $\mathbf{z}$.

3.3.2 The Critic Model: Speech LLM and Validator

The Critic evaluates the Actor's trajectories and generates the final translation. It is built upon a Qwen-omni Large Language Model backbone, structurally augmented with Router and Refiner modules.

Speech LLM Backbone. Adapted from the "thinker" component of the Qwen2.5-omni-7B, the speech encoder is modified for streaming by applying a windowed chunk autoregressive mask (400ms chunk size, 20-chunk local attention window, yielding an 8-second receptive field). Cyclic sinusoidal positional encodings are applied every 2 seconds.

The input is formatted using a multi-turn dialogue template. Task-level directives are embedded in the system prompt, and each audio chunk with its interleaved textual actions serves as a single conversational turn. The backbone processes this to yield local hidden states $\mathbf{H} = \text{LLM}(\mathbf{z}; \theta_{cri})$ and a globally compressed acoustic representation $\mathbf{h}_{global} = \text{AvePool}(\text{LLM}(\mathbf{S}; \theta_{cri}))$.

Router Module. Implemented as a lightweight 2-layer MLP with a Sigmoid activation, the Router projects the preceding hidden state $\mathbf{H}_{k-1}$ to a scalar probability $p_k \in [0, 1]$. While Section 3.2 details its role in the latency reward ($\mathcal{R}_{latency}$), structurally, p_k is thresholded by a predefined τ during inference to trigger either a "Read" ($p_k \geq \tau$) or "Write" ($p_k < \tau$) action.

Refiner Module. We adopt the Refiner architecture [3] for final generation. As shown in Figure 2b, the core structural integration involves repurposing the Router's probability output p as a dynamic routing scalar. The Refiner directly scales the global acoustic gain by p and the local acoustic gain by $(1 - p)$. This self-adaptive gating synchronizes the generation context with the streaming timing state before the refined features are projected to compute the consistency reward ($\mathcal{R}_{consistency}$). During inference, global acoustic information is absent. The Router can make policies without global information, and we do not use the Refiner at inference.

4 Experiment and Results

4.1 Experimental Setup

Training Data To support multilingual translation, we aggregate three open-source ASR datasets: Emilia[29], WenetSpeech[30], and GigaSpeech[31]. We utilize the Qwen3-235B-A22B[1] model to generate pseudo-labels for the transcribed texts, translating them into Chinese, English, and the other 4 European languages (DE, FR, ES, IT). (Chinese and English account for more than 50% of the pseudo-labels). We augment WenetSpeech and GigaSpeech by concatenating consecutive clips into long segments of a maximum of 30 seconds. All training stages dynamically sample from this unified corpus for training and validation.

Training Details Our training pipeline is implemented using PyTorch Lightning. The Speech LLM is parameter-efficiently fine-tuned using LoRA[32] with a rank of $r = 16$. When training the LLM for SimulST task, we also train offline ST and MT tasks in a multi-task way to preserve offline performance.

Following our proposed Actor-Critic paradigm, the training strictly proceeds in three stages:

1) **Actor Pre-training:** The auxiliary RNN-T Aligner is trained with the standard RNN-T loss to establish basic acoustic-linguistic alignments. Only the alignment task itself is new at this stage: the acoustic encoder is initialized from Whisper-medium, and the text-axis attention and MLP of each axial layer inherit the corresponding Qwen3-0.6B layer (the bottom 12 layers are frozen and receive no audio-axis attention; the upper 16 layers are augmented), while the audio-axis attention and the joint network are randomly initialized.
2) **Critic Pre-training:** Freezing the RNN-T parameters, we use its sampled trajectories to construct interleaved sequences. The Speech LLM (including the Router and Refiner) is optimized to adapt to this streaming input format.
3) **Joint Self-Enhancement:** All modules are unfrozen. The framework undergoes joint training via reverse distillation (as detailed in Section 3.2), continuously refining the Actor's sampling policy alongside the Critic's generation capability.

Benchmarks. We evaluate our model across diverse streaming scenarios. For **sentence-level evaluation**, we utilize the FLEURS[33] and CoVoST 2[34] datasets across the 10 supported directions. Notably, on FLEURS, we decouple the evaluation into Chinese-English bidirectional translations (to ensure a direct comparison with Seed-LiveInterpret2) and the remaining 8 directions. For **document-level continuous streaming**, we utilize the MuST-C and ACL6060 test sets (En-Zh, En-De). We use OmniSTEval for long-form evaluation.[2]

[1] `https://huggingface.co/Qwen/Qwen3-235B-A22B-Instruct-2507`

[2] `https://github.com/pe-trik/OmniSTEval/tree/main`

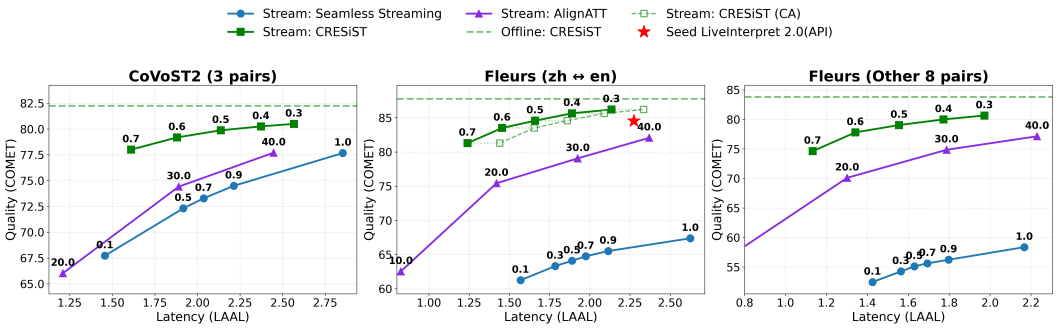


Figure 3: Sentence-level quality-latency trade-off curves on the CoVoST2 and FLEURS datasets. The proposed framework consistently establishes a superior Pareto frontier compared to Seamless Streaming and AlignATT across all sub-tasks.

Baselines. We benchmark our framework against strong state-of-the-art systems, including open-source foundations (**Seamless Streaming**), commercial SOTA **Seed-LiveInterpret2**, and academic simultaneous models (**AlignATT**, **StreamUni**, **InfiniSST**). More details about the baseline implementation can be found in Appendix C.3.

Metrics. Simultaneous translation requires balancing translation fidelity and streaming latency. For Translation Quality, we utilize **XCOMET-XXL**[3], a neural metric that aligns significantly better with human judgments compared to traditional n-gram metrics[35]. For **Streaming Latency**, we apply input-adaptive metrics to avoid heuristic bias: we report the Length-Adaptive Acoustic Latency (**LAAL**)[36] for sentence-level benchmarks, and adopt the recently proposed **LongYAAL** metric for document-level evaluations to accurately capture the cumulative delays in continuous long-form audio streams[37].

4.2 Main Results

Sentence-Level Quality-Latency Trade-offs. Figure 3 illustrates the quality-latency Pareto frontiers on the CoVoST2 and FLEURS benchmarks. By varying the decision threshold (e.g., $\tau \in [0.3, 0.7]$), we plot the translation quality against the latency metric. Our proposed framework consistently establishes a new state-of-the-art frontier, dominating both Seamless Streaming and AlignATT across all latency regimes. Notably, on the FLEURS (zh $\leftrightarrow$ en) benchmark, our streaming model strictly outperforms the commercial Seed LiveInterpret 2.0 API, achieving higher translation quality while maintaining a comparable or superior latency profile. Furthermore, the curve demonstrates that our model degrades gracefully at extremely low latency requirements (e.g., LAAL < 1.5), proving the effectiveness of the reverse-distilled streaming policy.

Document-Level Continuous Streaming. Evaluating on isolated sentences often obscures the compounding errors inherent in real-world, long-form translation. Table 1 presents our results on the MuST-C and ACL 60/60 benchmarks across three settings: *Offline w/ Oracle Segmentation*, *Streaming w/ Oracle Segmentation*, and the most rigorous *Streaming w/o Segmentation*. The contrast between the latter two settings effectively reveals a model's true capacity to process continuous, unbounded audio streams.

Notably, while InfiniSST is explicitly designed for infinite audio streams, it exhibits a paradoxical performance drop when evaluated at the sentence level with oracle boundaries. In contrast, CRESiST delivers consistently high performance across both streaming scenarios. Strikingly, our unsegmented streaming performance closely trails the oracle segmentation with a marginal degradation of only 1–2 COMET points, demonstrating that CRESiST can flawlessly manage infinite context without relying on explicit segmentation.

Furthermore, we observe that offline incremental methods such as StreamATT serve as highly competitive baselines on certain language pairs. However, their performance is notably inconsistent; on other language pairs, historical context selection leads to a catastrophic drop in translation

[3] `https://huggingface.co/Unbabel/XCOMET-XXL`

quality (e.g., on the ACL 60/60 en-de dataset, the COMET score plummets from 80.83 under oracle segmentation to 70.61) coupled with severe latency inflation. Ultimately, CRESiST successfully navigates infinite audio streams, avoiding the pitfalls of historical amnesia and alignment drift while maintaining state-of-the-art translation quality and minimal continuous delay. CRESiST is also complementary to RL-based policy optimization: GRPO post-training further improves it, whereas the same recipe cannot close the gap from a direct-SFT base (Appendix D.1).

Table 1: Document-level simultaneous translation performance on the MuST-C and ACL 60/60 benchmarks. The COMET is evaluated through XCOMET-XXL. Our proposed CRESiST model achieves state-of-the-art translation quality with minimal continuous delay, demonstrating exceptional robustness in the challenging unsegmented streaming scenario. *Oracle Segmentation* denotes sentence-level segmentation taken from the native sentence timestamps of the test sets, i.e., inference is run independently on each human-segmented sentence.

	MuST-C				ACL 60/60			
	en-zh		en-de		en-zh		en-de	
Method	COMET	LongYAAL	COMET	LongYAAL	COMET	LongYAAL	COMET	LongYAAL
Offline w/ Oracle Segmentation								
Qwen2.5-Omni	79.41	/	91.47	/	84.42	/	91.72	/
CRESiST	79.11	/	91.35	/	83.36	/	90.03	/
Streaming w/ Oracle Segmentation								
InfiniSST (m=3)	68.46	2.18	79.22	1.93	59.71	2.17	65.09	2.17
AlignATT (f=30)	73.45	1.39	84.10	1.10	76.94	1.74	80.83	1.57
CRESiST (τ=0.5)	**75.30**	**1.29**	**87.75**	**0.80**	**80.06**	**1.49**	**87.23**	**1.00**
Streaming w/o Segmentation								
StreamUni (k=3)	52.96	2.91	73.45	2.01	41.90	1.64	71.03	1.64
InfiniSST (m=3)	70.66	1.88	78.74	1.46	68.62	1.80	63.52	1.46
StreamATT (f=30)	70.23	2.05	74.35	2.42	76.63	2.15	70.61	1.89
CRESiST (τ=0.5)	**73.73**	**1.14**	**85.96**	**0.76**	**79.45**	**1.49**	**85.59**	**1.08**

5 Ablation Studies

5.1 Effect of Training Strategies

We first investigate the impact of our self-enhancing Actor-Critic training paradigm. Figure 4 presents the quality-latency trade-offs under different training strategies. For fairness, the training steps are kept identical. The baseline ("Stage 1 & Direct SFT"), which directly utilizes RNN-T aligned data for Supervised Fine-Tuning akin to the approach in InfiniSST, yields the poorest performance. It struggles to run at a granularity of 400 milliseconds (latency multiplier=1). The pipeline approach ("Stage 1 & 2"), which trains the modules sequentially without the final joint optimization, shows noticeable improvements over the direct SFT.

In contrast, our full CRESiST framework with joint self-enhancement achieves the best overall Pareto frontier. The improvement is particularly pronounced on the Chinese-English (zh $\leftrightarrow$ en) pairs, where the overall latency is significantly reduced while maintaining high translation quality.

5.2 Sensitivity to the Latency Penalty Weight

Next, as a sensitivity analysis rather than an ablation, we examine the impact of the latency penalty hyperparameter α on the system, with all variants trained for an identical number of steps. As illustrated in Figure 5, α explicitly governs the trade-off between translation quality and streaming latency. As expected, when α is large, the model aggressively penalizes delays, resulting in a compressed, lower overall latency profile, but at the cost of relatively degraded translation quality. Conversely, an excessively small α fails to adequately constrain the streaming policy, leading to disproportionately high latency spikes at specific decision thresholds.

5.3 Actor Architecture: Axial Blocks vs. Conventional RNN-T

Finally, we evaluate the architectural design of our auxiliary RNN-T Actor by comparing our proposed model (incorporating 2D axial attention blocks) against a conventional RNN-T architecture. We

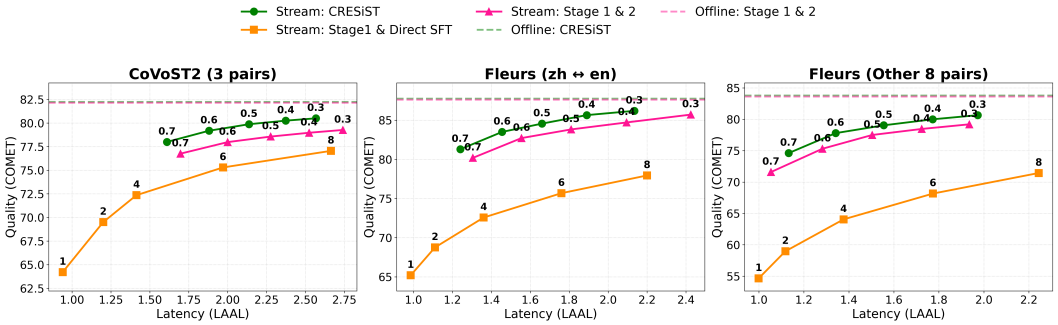


Figure 4: Ablation on training strategies. "Direct SFT" uses RNN-T aligned data directly (similar to InfiniSST). "Stage 1 & 2" represents pipeline training without the final joint optimization. "CRESiST" indicates our full joint training framework, which yields the most significant latency reduction, particularly on resource-rich directions (e.g., zh ↔ en).

assess the alignment quality on the validation set using two metrics: the perplexity (PPL, base e) of the emitted target text and the Average Lagging (AL, measured in seconds) of the generated interleaved sequence. [38]

As detailed in Table 2, both the two checkpoints are trained for 600k steps. The conventional RNN-T struggles significantly to produce optimal read-write paths, exhibiting abnormally high latency (AL = 8.03s) and poor text generation confidence (PPL = 22.3). The inclusion of axial blocks effectively strengthens the interaction between audio and text modalities, drastically improving alignment behavior. This design successfully drops the PPL to 6.03 and reduces the average lagging to 1.81s, providing a much higher-quality interleaved sequence for the downstream Speech LLM. We further verify in Appendix D.2 that the Actor is effective as a policy explorer but not as the final translator: reverse distillation transfers its latency behavior, while its translation quality stays far below the Speech LLM.

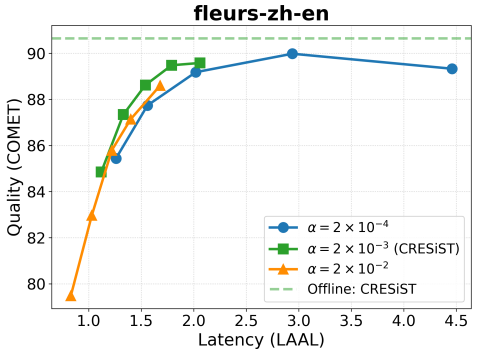


Figure 5: Impact of the latency penalty weight on the quality-latency trade-off. Larger weights aggressively compress latency at the expense of quality, while excessively small weights lead to prolonged delays at specific thresholds.

Table 2: Comparison of the Actor architecture using a conventional RNN-T versus our proposed model with axial blocks. Evaluated on the validation set by text perplexity (PPL) and Average Lagging (AL, in seconds). N_{pd} and N_{ax} are the number of layers in the predictor and the Axial attention module respectively

Model	N_{pd}	N_{ax}	PPL	AL
Conventional	28	0	22.3	8.03
Proposed	12	16	**6.03**	**1.81**

6 Conclusion and Limitations

Conclusion. In this work, we introduced a novel self-enhancing Actor-Critic framework designed to transition simultaneous speech translation from rigid, incremental processing to a dynamic, read-write interleaved paradigm. By decoupling policy exploration and reward evaluation, our method empowers an auxiliary RNN-T Actor to construct optimal streaming trajectories, while the Speech LLM Critic leverages its Router-Refiner module to ensure high-fidelity generation. Through a closed-

loop reverse distillation mechanism, the framework autonomously aligns the streaming policy with the LLM's generation rhythm. Extensive experiments demonstrate that our approach establishes new state-of-the-art quality-latency trade-offs across both sentence-level and challenging document-level benchmarks.

Limitations and Future Work. Despite the promising results, our framework exhibits a few limitations that warrant future investigation:

- **Data and Domain Bottlenecks:** As this work primarily focuses on streaming policy optimization, we didn't perform rigorous data engineering. The absolute translation quality is fundamentally bottlenecked by the backbone capacity and offline SFT data, which sub-optimally affect overall performance. For example, we observed an offline performance drop on the ACL60/60 dev set compared to the vanilla Qwen2.5-omni model, indicating domain distribution discrepancies and catastrophic forgetting in specialized academic domains.
- **Document-level Latency Control:** While adjusting the decision threshold (τ) effectively modulates the sentence-level quality-latency trade-off, it yields marginal tuning effects during document-level continuous streaming. Regulating the compounding dynamics of infinite audio streams likely requires dynamic penalty mechanisms, such as the context-aware latency multiplier utilized in InfiniSST, rather than relying solely on a static probability threshold.

Acknowledgments and Disclosure of Funding

This work was supported in part by China NSFC project under Grants No. 62631016, No. 62601524 and No. U25A20409; in part by SJTU Med-X (Medicine & Engineering) Translational Research Grant (YG2025LC09); in part by the UM and UMDF (Grant Nos. MYRG-GRG2024-00165-FST-UMDF, MYRG-GRG2025-00236-FST, MYRG-GRG2026-00018-FIC, MYRG-CRG2026-00007-FIC); and in part by the Alibaba Innovative Research Program.

References

[1] X. Ma, A. Sun, S. Ouyang, H. Inaguma, and P. Tomasello, "Efficient Monotonic Multihead Attention," *arXiv preprint arXiv:2312.04515*, 2023.

[2] Seamless 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.

[3] C. Le, B. Han, J. Li, C. Songyong, and Y. Qian, "Simulmega: Moe routers are advanced policy makers for simultaneous speech translation," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[4] 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.

[5] N. Hirschkind, J. Liu, X. Yu, and M. K. Nandwana, "Reina: Regularized entropy information-based loss for efficient simultaneous speech translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 37, 2026, pp. 31 005–31 013.

[6] D. Liu, G. Spanakis, and J. Niehues, "Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection," *arXiv preprint arXiv:2005.11185*, 2020.

[7] S. Ouyang, X. Xu, and L. Li, "Infinisst: Simultaneous translation of unbounded speech with large language model," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 3032–3046.

[8] S. Cheng, Z. Huang, T. Ko, H. Li, N. Peng, L. Xu, and Q. Zhang, "Towards achieving human parity on end-to-end simultaneous speech translation via llm agent," *arXiv preprint arXiv:2407.21646*, 2024.

[9] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association (Interspeech 2017)*, 2017, pp. 498–502.

[10] T. Xu, Z. Huang, J. Sun, S. Cheng, and W. Lam, "Seqpo-simt: Sequential policy optimization for simultaneous machine translation," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 16 107–16 123.

[11] S. Cheng, Y. Bao, Z. Huang, Y. Lu, N. Peng, L. Xu, R. Yu, R. Cao, Y. Du, T. Han, Y. Hu, Z. Li, S. Liu, S. Ma, S. Pan, J. Xiao, N. Xu, M. Yang, R. Ye, Y. Yu, J. Zhang, R. Zhang, W. Zhang, W. Zhu, L. Zou, L. Lu, Y. Wang, and Y. Wu, "Seed LiveInterpret 2.0: End-to-end simultaneous speech-to-speech translation with your voice," *arXiv preprint arXiv:2507.17527*, 2025, Seed-LiveInterpret 2.0 Technical Report, ByteDance Seed.

[12] T. Labiausse, R. Fabre, Y. Estève, A. Défossez, and N. Zeghidour, "Simultaneous speech-to-speech translation without aligned data," *arXiv preprint arXiv:2602.11072*, 2026.

[13] S. Ouyang, S. Ding, O. Hrinchuk, V. Lavrukhin, B. Yan, B. Ginsburg, and L. Li, "Hierarchical policy optimization for simultaneous translation of unbounded speech," *arXiv preprint arXiv:2604.21045*, 2026.

[14] J. Xu, Z. Guo, J. He, H. Hu, T. He, S. Bai, K. Chen, J. Wang, Y. Fan, K. Dang, B. Zhang, X. Wang, Y. Chu, and J. Lin, "Qwen2.5-omni technical report," 2025. [Online]. Available: https://arxiv.org/abs/2503.20215

[15] P. Zhang, Y. Wang, J. Tang, B. Yang, R. Wang, D. F. Wong, and F. Huang, "Direct simultaneous translation activation for large audio-language models," in *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026, pp. 18 367–18 371.

[16] Z. Yang, L. Wei, R. Koshkin, X. Chen, and S. Nakamura, "Sasst: Leveraging syntax-aware chunking and llms for simultaneous speech translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 40, 2026, pp. 34 358–34 367.

[17] B. Fu, D. Yu, M. Liao, C. Li, X. Chen, Y. Chen, K. Fan, and X. Shi, "Efficient and adaptive simultaneous speech translation with fully unidirectional architecture," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 36, 2026, pp. 30 735–30 743.

[18] T. Labiausse, L. Mazaré, E. Grave, A. Défossez, and N. Zeghidour, "High-fidelity simultaneous speech-to-speech translation," in *International Conference on Machine Learning*. PMLR, 2025, pp. 32 116–32 129.

[19] L. Dong and B. Xu, "Cif: Continuous integrate-and-fire for end-to-end speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6079–6083.

[20] K. Deng, W. Chen, X. Chen, and P. Woodland, "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)*, 2025, pp. 16 718–16 734.

[21] S. Guo, X. Li, M. Liu, W. Chen, and Y. Feng, "Streamuni: Achieving streaming speech translation with a unified large speech-language model," *arXiv preprint arXiv:2507.07803*, 2025.

[22] 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.

[23] 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.

[24] H. Tan, H. Ouchi, and S. Sakti, "Simulsense: Sense-driven interpreting for efficient simultaneous speech translation," in *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026, pp. 19 437–19 441.

[25] I. Tsiamas, G. I. Gállego, J. A. Fonollosa, and M. R. Costa-jussà, "Shas: Approaching optimal segmentation for end-to-end speech translation," *arXiv preprint arXiv:2202.04774*, 2022.

[26] S. Papi, M. Gaido, M. Negri, and L. Bentivogli, "Streamatt: Direct streaming speech-to-text translation with attention-based audio history selection," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 3692–3707.

[27] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[29] 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.

[30] 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.

[31] G. Chen, S. Chai, G.-B. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *Interspeech 2021*, 2021.

[32] 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," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nSbk_W2q56v

[33] 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.

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

[35] N. M. Guerreiro, R. Rei, D. v. Stigt, L. Coheur, P. Colombo, and A. F. T. Martins, "xCOMET: Transparent machine translation evaluation through fine-grained error detection," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 979–995, 2024. [Online]. Available: https://aclanthology.org/2024.tacl-1.54/

[36] S. Papi, M. Gaido, M. Negri, and M. Turchi, "Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation," in *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, J. Ive and R. Zhang, Eds. Online: Association for Computational Linguistics, Jul. 2022, pp. 12–17. [Online]. Available: https://aclanthology.org/2022.autosimtrans-1.2/

[37] P. Polák, S. Papi, L. Bentivogli, and O. Bojar, "Better late than never: Meta-evaluation of latency metrics for simultaneous speech-to-text translation," *arXiv preprint arXiv:2509.17349*, 2025.

[38] 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, Jul. 2019, pp. 3025–3036. [Online]. Available: https://aclanthology.org/P19-1289/

[39] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3505–3506.

[40] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[41] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "Must-c: a multilingual speech translation corpus," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2012–2017.

[42] E. Salesky, K. Darwish, M. Al-Badrashiny, M. Diab, and J. Niehues, "Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology," in *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, E. Salesky, M. Federico, and M. Carpuat, Eds. Toronto, Canada (in-person and online): Association for Computational Linguistics, Jul. 2023, pp. 62–78. [Online]. Available: https://aclanthology.org/2023.iwslt-1.2/

[43] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2505.09388

A Overall Training Algorithm

Algorithm 1 summarizes the complete end-to-end training procedure of our proposed Actor-Critic framework for simultaneous translation.

Algorithm 1 Actor-Critic Training for Simultaneous Translation

Input: Dataset $\mathcal{D} = \{(\mathbf{S}, \mathbf{T})\}$, Actor parameters θ_{act}, Critic parameters θ_{cri}, Learning rate η, Decision threshold τ, Loss weights λ, α, β, critic step num K.

while not converged **do**
  Sample a batch of speech-text pairs: $(\mathbf{S}, \mathbf{T}) \sim \mathcal{D}$
  // Step 1: Actor Exploration & Transition Dynamics ***($\mathcal{S}, \mathcal{A}, \mathcal{P}$)***
  Sample alignment path ($a_k \in \mathcal{V} \cup \{\phi\}$, ϕ denotes Read): $\mathbf{a} \sim \pi_{act}(\cdot \mid \mathbf{S}, \mathbf{T}; \theta_{act})$
  Construct interleaved state sequence: $\mathbf{z} = \text{Interleave}(\mathbf{S}, \mathbf{T}, \mathbf{a})$
  // Step 2: Critic Forward Pass & State Evaluation
  Extract main body hidden states: $\mathbf{H} = \text{LLM}(\mathbf{z}; \theta_{cri})$
  Predict router probabilities ($p_k \in [0, 1]$): $p_k = \text{Router}(\mathbf{H}_{k-1}; \theta_{cri})$
  Extract global acoustic state: $\mathbf{h}_{global} \leftarrow \text{AvePool}(\text{LLM}(\mathbf{S}; \theta_{cri}))$
  Predict refiner distributions: $P_{refiner}(t_i \mid \mathbf{z}_{\leq \text{idx}(i)}, \mathbf{H}, \mathbf{h}_{global}; \theta_{cri})$
  // Step 3: Compute Reward Function $\mathcal{R}$ & Critic Loss
  Compute Consistency Reward:
  $\mathcal{R}_{consistency} \leftarrow \sum_i \log P_{refiner}(t_i \mid \cdot) + \sum_{\{k|p_k<\tau\}} \log P_{llm}(y_k \mid \mathbf{H}_k)$
  Compute Latency Reward: $\mathcal{R}_{latency} \leftarrow \sum_k \sum_{m=1}^{M} \prod_{j=1}^{m} \left(1 - p_{k+j}\right)$
  Compute Load Balancing constraint: $\mathcal{L}_{balance}(p_k)$
  Aggregate Critic cost: $\mathcal{J}_{critic} = -\mathcal{R}_{consistency} - \alpha \mathcal{R}_{latency} + \beta \mathcal{L}_{balance}$
  // Step 4: Policy Feedback & Actor Optimization
  Compute base Actor loss: $\mathcal{L}_{rnnt} = -\log \pi_{act}(\mathbf{T} \mid \mathbf{S}; \theta_{act})$
  Derive optimal action targets from Critic: $\hat{a}_k = \mathbb{I}(p_k \geq \tau)$
  Compute policy distillation loss: $\mathcal{L}_{distill} = \sum_k \text{BCE}\Big(\hat{a}_k, \pi_{act}(a_k = \phi \mid \cdot; \theta_{act})\Big)$
  // Step 5: Parameter Updates
  if $t \bmod (K + 1) = 0$ **then**
    Update Actor parameters: $\theta_{act} \leftarrow \theta_{act} - \eta \nabla_{\theta_{act}} \big(\mathcal{L}_{distill} + \lambda \mathcal{L}_{rnnt}\big)$
  else
    Update Critic parameters: $\theta_{cri} \leftarrow \theta_{cri} - \eta \nabla_{\theta_{cri}} \mathcal{J}_{critic}$
  end if
end while

B Convergence Analysis of the Reverse Distillation Loop

This appendix provides a local convergence analysis for the Actor–Critic reverse distillation loop in Section 3.2, formalizing the intuition that the Actor's quality is controlled by the Critic's quality despite the self-referential pseudo-labels. The analysis is local; we do not claim global convergence on a non-convex landscape.

We re-parameterize the Actor and Critic as θ (θ_{act}) and ϕ (θ_{cri}) for analytical convenience. The two-timescale update at outer step t runs $K = 8$ Critic SGD steps followed by one Actor step; the pseudo-label $\hat{a}_k = \mathbb{I}(\text{sg}[p_k(\phi)] \geq \tau)$ uses a stop-gradient on ϕ. Throughout, we assume standard regularity in a local neighborhood $\mathcal{N} \times \mathcal{M}$ of an equilibrium (θ^*, ϕ^*): local smoothness of J_a and J_c in their first arguments, a quadratic-growth condition for the Critic around $\phi^*(\theta) := \arg\min_\phi J_c(\phi; \theta)$, cross-Lipschitzness of $\nabla_\phi J_c$ in θ, a Tsybakov-style margin condition $\mathbb{P}(|p(s; \phi) - \tau| \leq \delta) \leq \kappa\delta$ on the Router output, a Polyak–Łojasiewicz condition for J_a, unbiased minibatch gradients with bounded variance σ^2, and bounded gradients on $\mathcal{N} \times \mathcal{M}$. These conditions are weaker than those typically invoked in the analysis of two-timescale stochastic approximation (Borkar 2008; Karimi et al. 2016) and are widely satisfied by overparameterized neural networks. We refer the reader to standard references for their formulation.

Theorem 1 (Local linear convergence to a noise-controlled neighborhood)**.** Let $e_t := \phi_{t+1} - \phi^*(\theta_t)$ denote the Critic tracking error and $V_t := J_a(\theta_t; \phi^*(\theta_t)) - \min_\theta J_a(\theta; \phi^*(\theta_t))$ the Actor optimality gap. Under the regularity conditions above, there exist constants $\eta_a^0, \eta_c^0, K_0, c, C_\sigma > 0$ and a contraction rate $\rho^* \in (0, 1)$ such

that for any

$$\eta_c \le \eta_c^0, \qquad \eta_a \le \eta_a^0, \qquad K \ge K_0,$$

and any initialization $(\theta_0, \phi_0) \in \mathcal{N} \times \mathcal{M}$, the Lyapunov function $\Phi_t := V_t + c\,\mathbb{E}\|e_t\|^2$ satisfies

$$\Phi_t \;\le\; (\rho^*)^t\,\Phi_0 \;+\; \frac{C_\sigma\,(\eta_a+\eta_c)\,\sigma^2}{1-\rho^*}. \tag{7}$$

A valid concrete choice is $c = 4\eta_a L_{a,\phi}^2/\mu_a$ and $K_0 = \lceil \log(8L_\phi^2\eta_a^0 G^2/\mu_a)/\log(1/\rho)\rceil$, yielding $\rho^* = \max(1-\mu_a\eta_a/2,\ 3/4)$, where $L_{a,\phi} = O(\kappa L_p)$ controls the Actor's sensitivity to Critic perturbations through the margin condition.

Proof. The proof has three steps: the Critic tracking error contracts geometrically, the Actor optimality gap descends modulo Critic error, and a weighted Lyapunov function combines the two.

Critic tracking error. The map $\theta \mapsto \phi^*(\theta)$ is L_ϕ-Lipschitz with $L_\phi \le L_{c,\theta}/\mu_c$, by a standard implicit-function argument under quadratic growth (Karimi et al. 2016, Thm. 5). Since $\phi_{t,0} = \phi_t = \phi^*(\theta_{t-1}) + e_{t-1}$, we obtain

$$\|\phi_{t,0} - \phi^*(\theta_t)\| \;\le\; \|e_{t-1}\| + L_\phi\|\theta_t - \theta_{t-1}\| \;\le\; \|e_{t-1}\| + L_\phi\eta_a G,$$

where the second inequality uses the bounded-gradient assumption. The standard SGD analysis under quadratic growth yields, for the K-step inner loop with $\eta_c \le 1/L_c$,

$$\mathbb{E}\|\phi_{t,K} - \phi^*(\theta_t)\|^2 \;\le\; \rho^K\,\mathbb{E}\|\phi_{t,0} - \phi^*(\theta_t)\|^2 + \frac{\eta_c\sigma^2}{\mu_c}, \qquad \rho := 1-\eta_c\mu_c.$$

Squaring and combining the two displays via $(a+b)^2 \le 2a^2 + 2b^2$,

$$\mathbb{E}\|e_t\|^2 \;\le\; 2\rho^K\,\mathbb{E}\|e_{t-1}\|^2 + 2\rho^K L_\phi^2\eta_a^2 G^2 + \frac{\eta_c\sigma^2}{\mu_c}. \tag{8}$$

Choosing K_0 so that $2\rho^{K_0} \le 1/2$ (this is the K_0 stated in the theorem),

$$\mathbb{E}\|e_t\|^2 \;\le\; \tfrac{1}{2}\mathbb{E}\|e_{t-1}\|^2 + \underbrace{\rho^K L_\phi^2\eta_a^2 G^2 + \frac{\eta_c\sigma^2}{\mu_c}}_{=:\,\varepsilon_c}. \tag{9}$$

Actor descent. Let $g_t^* := \nabla_\theta J_a(\theta_t; \phi^*(\theta_t))$ and $g_t := \nabla_\theta J_a(\theta_t; \phi_{t+1})$. The pseudo-label $\hat{a}_k$ flips between ϕ_1 and ϕ_2 only on the event $\{|p(s;\phi_1) - \tau| \le L_p\|\phi_1 - \phi_2\|\}$, which by the margin condition has probability at most $\kappa L_p\|\phi_1 - \phi_2\|$. Combined with the Lipschitzness of BCE in its target argument, this yields

$$\|g_t - g_t^*\| \;\le\; L_{a,\phi}\,\|e_t\|, \qquad L_{a,\phi} = O(\kappa L_p). \tag{10}$$

By smoothness of J_a in θ (with the second argument held at $\phi^*(\theta_t)$) and unbiasedness of the stochastic gradient,

$$\mathbb{E}\,J_a(\theta_{t+1}; \phi^*(\theta_t)) \;\le\; J_a(\theta_t; \phi^*(\theta_t)) - \eta_a\langle g_t^*, g_t\rangle + L_a\eta_a^2\|g_t\|^2 + L_a\eta_a^2\sigma^2.$$

Substituting $g_t = g_t^* + (g_t - g_t^*)$, using Cauchy–Schwarz and Young's inequality together with (10), and applying the PL condition $\|g_t^*\|^2 \ge 2\mu_a V_t$:

$$\mathbb{E}\,J_a(\theta_{t+1}; \phi^*(\theta_t)) \;\le\; J_a(\theta_t; \phi^*(\theta_t)) - \mu_a\eta_a V_t + 2\eta_a L_{a,\phi}^2\mathbb{E}\|e_t\|^2 + L_a\eta_a^2\sigma^2,$$

provided $\eta_a \le 1/(4L_a)$. Shifting the second argument from $\phi^*(\theta_t)$ to $\phi^*(\theta_{t+1})$ contributes only $O(\eta_a^2)$ by the implicit-function bound and a standard envelope argument; folding this into the noise term gives

$$\mathbb{E}V_{t+1} \;\le\; (1-\mu_a\eta_a)\,V_t + 2\eta_a L_{a,\phi}^2\,\mathbb{E}\|e_t\|^2 + C_1\eta_a^2\sigma^2. \tag{11}$$

Lyapunov function. Set $\Phi_t := V_t + c\,\mathbb{E}\|e_t\|^2$ with $c = 4\eta_a L_{a,\phi}^2/\mu_a$. Combining (9) and (11),

$$\mathbb{E}\Phi_{t+1} \;\le\; (1-\mu_a\eta_a)V_t + \big(2\eta_a L_{a,\phi}^2 + \tfrac{c}{2}\big)\,\mathbb{E}\|e_t\|^2 + C_\sigma(\eta_a+\eta_c)\sigma^2.$$

With our choice of c, the coefficient on $\mathbb{E}\|e_t\|^2$ equals $\frac{c(1+\mu_a)}{2} \le \frac{3c}{4}$ (after a harmless rescaling so that $\mu_a \le 1/2$). Hence

$$\mathbb{E}\Phi_{t+1} \;\le\; \rho^*\,\mathbb{E}\Phi_t + C_\sigma(\eta_a+\eta_c)\sigma^2, \qquad \rho^* := \max\big(1-\tfrac{\mu_a\eta_a}{2},\ \tfrac{3}{4}\big),$$

and iterating yields (7). □

Interpretation. The bound decomposes joint convergence into two coupled contractions: the Critic tracks $\phi^*(\theta_t)$ at geometric rate ρ^K per outer step, and the Actor descends in J_a at rate $1 - \mu_a\eta_a$ modulo a perturbation proportional to the Critic's tracking error. The two-timescale schedule $K = 8$ ensures ρ^K is small enough that the Critic does not bottleneck the Actor. The margin condition enters only through the constant $L_{a,\phi} = O(\kappa L_p)$: a sharper Router output (smaller κ) means small Critic perturbations rarely flip the binarized pseudo-label, so the Actor is robust to a noisy or undertrained Critic. At stationarity,

$$\limsup_t \mathbb{E}V_t \;\le\; \frac{2L_{a,\phi}^2}{\mu_a} \cdot \limsup_t \mathbb{E}\|e_t\|^2 + O(\eta_a\sigma^2),$$

which formalizes the qualitative claim that *the Actor is no worse than the Critic, up to a constant determined by the margin and PL parameters*. We do not claim global convergence, nor that the equilibrium matches a true oracle policy—the latter would require J_c being a faithful surrogate for translation quality.

C Extended Experimental Details

In this section, we provide comprehensive details regarding the training configurations, inference setup, and baseline implementations to facilitate reproducibility. Detailed hyperparameter configurations will be made publicly available alongside our open-sourced code.

C.1 Load Balancing Loss

To prevent routing collapse in our two-expert Refiner architecture and maintain numerical stability, we adopt a lightweight, combined load-balancing loss. We first define a squared soft margin penalty, $\mathcal{M}(v; a, b)$, which continuously penalizes a variable v when it deviates from a target interval $[a, b]$:

$$\mathcal{M}(v; a, b) = \begin{cases} (a - v)^2, & \text{if } v < a \\ (v - b)^2, & \text{if } v > b \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

This squared formulation ensures smooth, differentiable gradients at the boundaries.

Given a batch of N tokens, let x_i denote the raw routing logit for the i-th token, and $p_i = \sigma(x_i)$ be its probability of being routed to the first expert. Let $\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i$ represent the batch-averaged routing probability. Our overall balancing loss, L_{bal}, simultaneously regularizes the micro-level logits to prevent Sigmoid saturation and constrains the macro-level routing probabilities to ensure equitable expert utilization:

$$L_{balance} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{M}(x_i; -5, 5) + \mathcal{M}(\bar{p}; 0.3, 0.7) \tag{13}$$

By softly bounding individual logits within $[-5, 5]$ and the average routing probability within $[0.3, 0.7]$, the model mathematically guarantees statistical load balancing while preserving token-level routing flexibility.

C.2 Training and Inference Details

Hardware and Environment All models were trained and evaluated on a single computation node equipped with eight NVIDIA A100 (80GB) GPUs. Our implementation is built upon the PyTorch framework and PyTorch Lightning. To optimize the memory footprint during training, we integrated DeepSpeed ZeRO Stage 2[39] alongside gradient checkpointing.

Optimization and Scheduling Across all training stages, we employed the AdamW[40] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and a weight decay of 0.001. The learning rate was governed by a linear decay schedule following a 500-step linear warmup. During the parameter-efficient fine-tuning of the Speech LLM backbone, LoRA was applied to all linear modules (excluding word embeddings and the final projection layer) with a rank of $r = 16$ and an alpha parameter of $\alpha = 32$. The peak learning rate was set to 1.0×10^{-4} for Stages 1 and 2, and 2.0×10^{-5} for Stage 3. Additionally, the training batch size per GPU was dynamically bounded to a maximum of 30 seconds of audio. Training of each stage takes about 8 hours.

Inference Setup For sentence-level inference, we utilized Voice Activity Detection (VAD) exclusively to truncate initial silence at the beginning of audio clips. For document-level inference, VAD was applied solely to filter out excessively long non-speech pauses (e.g., > 2 seconds) to prevent severe distribution mismatches with the training data. Acoustic features were continuously extracted using a 400ms window. The KV cache for both the acoustic prefix and the generated text tokens was dynamically maintained and incrementally updated. To minimize inference latency, we adopted a greedy decoding strategy coupled with a windowed 4-gram no-repeat logits processor to prevent degenerative repetitions. All reported latency metrics—for both our proposed method and the baselines—were computed at the word level (or character level for CJK languages).

C.3 Baseline Implementation Details

To ensure a fair and rigorous comparison, we carefully configured all baselines according to their official implementations or API specifications. Both the proposed method and the baselines were evaluated within a unified framework, closely mirroring the official SimulEval protocol. Greedy decoding was uniformly applied across all baseline models.

- **Seamless Streaming [2]:** We utilized the official `seamless-streaming` checkpoint. Its read-write strategy is governed by the native EMMA policy, and we generated the quality-latency Pareto frontier by adjusting the EMMA decision threshold. The audio input stride was set to 400ms, which is identical to our method.
- **AlignATT [23] & StreamATT [26]:** We followed the implementations of AlignATT and StreamATT provided in the InfiniSST repository[4], adapting them to operate on top of the original Qwen2.5-Omni Model. We varied the frame boundary to trace the quality-latency trade-off. We have compared extracting attention weights from the middle layer, the final layer, and averaging over layers, showing that an extract from the middle layer yields slightly better results. For StreamATT, the context window was restricted to 40 text tokens, with the retained acoustic history bounded between 10 and 30 seconds. Infinite repetition is hard to avoid in StreamATT, even with the no-repeats logits processor. So we set repetition_penalty=1.2 in StreamATT.
- **StreamUni [21]:** We evaluated this baseline using its official checkpoints and code. We use chunk size=640ms and wait-k=3.
- **InfiniSST [7]:** We evaluated this baseline using its official checkpoints[5]. Latency was modulated by varying the input chunk size from 960ms to 4800ms. Since we observed that these checkpoints exhibited degenerate performance during sentence-level evaluations, we only report their document-level results.
- **Seed LiveInterpret 2.0:** This model was evaluated via the commercial Volcengine WebSocket API.[6]. To minimize network-induced variance, we ensured the network Round Trip Time (RTT) remained strictly below 20ms during all queries. Due to budget constraints, this API was evaluated exclusively on the FLEURS test set.

D Additional Experiments

D.1 Compatibility with RL-based Policy Optimization

CRESiST is an SFT-time policy-discovery framework: translation is still learned by supervised maximum likelihood on the offline corpus, while the read-write policy is discovered by the Actor-Critic loop. It is therefore complementary to, rather than a substitute for, RL-based policy optimization, and the resulting model is a valid starting point for subsequent RL. To verify this, we apply GRPO on top of CRESiST following the reward design of HPO [13]. For reference, we apply the same recipe to an InfiniSST-style SFT baseline trained on the same Qwen2.5-Omni backbone with RNN-T aligned data (the "Direct SFT" setting of Section 5). CRESiST+GRPO is trained for 700 steps on the multilingual corpus, and SFT+GRPO for 1000 steps on En-Zh. Table 3 reports document-level En-Zh results in the streaming w/o segmentation setting.

Table 3: Document-level En-Zh results (streaming w/o segmentation, XCOMET-XXL ↑ / LongYAAL ↓) with and without GRPO post-training. SFT denotes an InfiniSST-style supervised baseline on the same backbone.

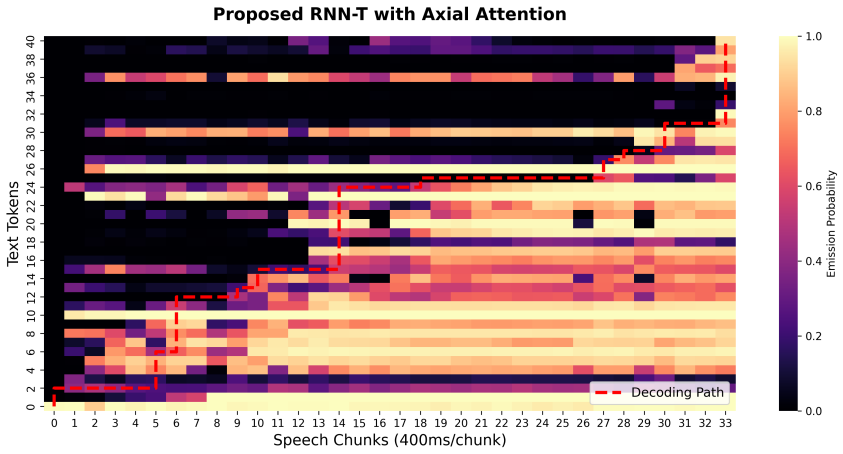
	ACL 60/60		MuST-C	
System	COMET	LongYAAL	COMET	LongYAAL
SFT (InfiniSST-style)	68.43	2.27	66.16	2.34
SFT + GRPO (HPO recipe)	73.64	2.15	69.71	2.15
CRESiST	79.45	1.49	73.73	1.14
CRESiST + GRPO	**80.04**	**1.37**	**78.47**	1.23

Three observations follow. (1) CRESiST is fully compatible with RL: GRPO further improves it on both benchmarks (e.g., +4.7 COMET on MuST-C) without hurting latency. (2) RL does not compensate for a weak SFT base: CRESiST without any RL already outperforms SFT+GRPO by a clear margin in both quality and latency, so SFT-time policy learning remains meaningful even when RL post-training is available. (3) Our GRPO

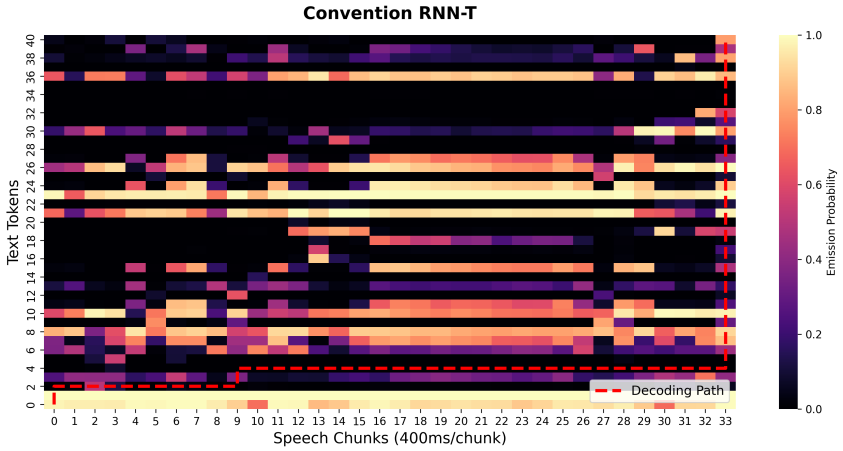
[4] `https://github.com/LeiLiLab/InfiniSST/blob/release/agents/alignatt.py`

[5] `https://github.com/LeiLiLab/InfiniSST`

[6] `https://www.volcengine.com/docs/6561/1756902`



(a) Proposed RNN-T



(b) Conventional RNN-T

Figure 6: Comparison of emission probabilities and decoding paths. The proposed RNN-T with axial attention (a) successfully maintains an incremental, monotonic alignment for simultaneous translation. In contrast, the conventional RNN-T (b) fails to emit tokens timely during the speech stream, causing its decoding path to delay and degenerate into an offline translation behavior at the end of the utterance.

gains are smaller than those reported in the HPO paper, which we attribute to the stronger starting point. We also note that after GRPO post-training the model streams raw audio stably without the VAD pause filtering described in Appendix C.

D.2 The RNN-T Actor as an Inference Model

A natural question is whether the self-enhancement loop improves the RNN-T Actor enough for it to serve as the final inference model on its own, which would also quantify how much of the final performance is contributed by the Speech LLM. Our Actor is itself a complete simultaneous translation model trained on the same data (Whisper-medium encoder, Qwen3-0.6B predictor with axial attention, and a joint network), so we decode it directly with streaming greedy decoding (blank = Read, non-blank = Write), both before (Stage 1) and after (Stage 3) reverse distillation. Table 4 reports sentence-level results on FLEURS.

The results confirm that the RNN-T is effective as an Actor but unsuitable as the final translator. Reverse distillation transfers the *policy*, not the quality: the Actor's COMET is essentially unchanged, while its latency drops markedly (Zh-En LAAL 2.29s $\rightarrow$ 1.52s, close to that of the Speech LLM), showing that its read/write behavior has been aligned with the Critic. Its translation quality, however, remains 33–37 XCOMET points below the Speech LLM; the RNN-T decides reliably *when* to translate, but its 0.6B predictor lacks the linguistic

Table 4: Using the RNN-T Actor directly as the inference model on FLEURS (XCOMET-XXL ↑ / LAAL ↓), before and after reverse distillation, compared with the full CRESiST Speech LLM.

	En-Zh		Zh-En	
Inference model	COMET	LAAL	COMET	LAAL
RNN-T Actor, before distillation	43.7	3.20	55.5	2.29
RNN-T Actor, after distillation	44.1	3.02	55.1	**1.52**
CRESiST (Speech LLM)	**80.5**	**1.78**	**88.6**	1.54

and world knowledge needed for *what* to generate, with named-entity hallucinations being the dominant failure mode. The RNN-T is also an inefficient inference model: each step evaluates the joint network along an entire row or column of the $(T \times U)$ lattice, so its decoding cost grows with the product of the audio and text lengths rather than with the sequence length, and scaling the transducer toward the 7B class would multiply this lattice cost while still lacking pretrained LLM knowledge. These results cleanly separate the two roles in our framework, where the Actor supplies the policy and the Speech LLM supplies the quality, and justify keeping a small RNN-T as the policy explorer paired with a pretrained Speech LLM as the generator.

E Case Visualization

In this section, we present case visualizations to demonstrate the effectiveness of our proposed framework from two distinct perspectives: the internal alignment quality of our speech-text aligner, and the overall end-to-end translation latency.

E.1 Analysis of RNN-T Alignment Quality

First, we examine the internal behavior of our proposed RNN-T aligner. Figure 6 visualizes the emission probabilities and the resulting decoding paths for both the proposed RNN-T (with axial attention) and a conventional RNN-T on a specific translation case.

As explicitly shown in Figure 6b, the conventional RNN-T struggles to confidently predict the necessary text tokens during the ongoing speech stream. Due to this failure in timely token emission, it cannot establish a proper incremental alignment. Consequently, the decoding path is forced to delay the majority of its predictions until the very end of the speech input (around chunk 33), effectively degenerating the system into an offline translation paradigm.

In contrast, our proposed RNN-T (Figure 6a) overcomes this critical limitation. It yields significantly sharper and more concentrated emission probabilities, successfully predicting tokens as the speech stream unfolds. This distinct, high-quality monotonic decoding path demonstrates that our approach effectively maintains the incremental alignment required for true simultaneous translation, avoiding the offline degradation seen in the conventional baseline.

E.2 End-to-End Translation Timing Comparison

Furthermore, we visualize the end-to-end translation timing of a specific case from the FLEURS EN-ZH test set to compare our approach directly with the Seed LiveInterpret 2.0 API.

As illustrated in Figure 7, Seed LiveInterpret 2.0 exhibits a strong preference for coarse-grained, clause-level translation, often waiting for larger chunks of speech context to accumulate before generating an output. Conversely, our proposed model performs fine-grained, adaptive translation at the word level. This allows our model to execute more immediate and flexible read/write decisions, thereby effectively reducing translation latency while maintaining high generation quality.

F Assets and Licenses

Table 5 details the existing assets (including datasets, models, and codebases) utilized in this work, alongside their licenses and reference information.

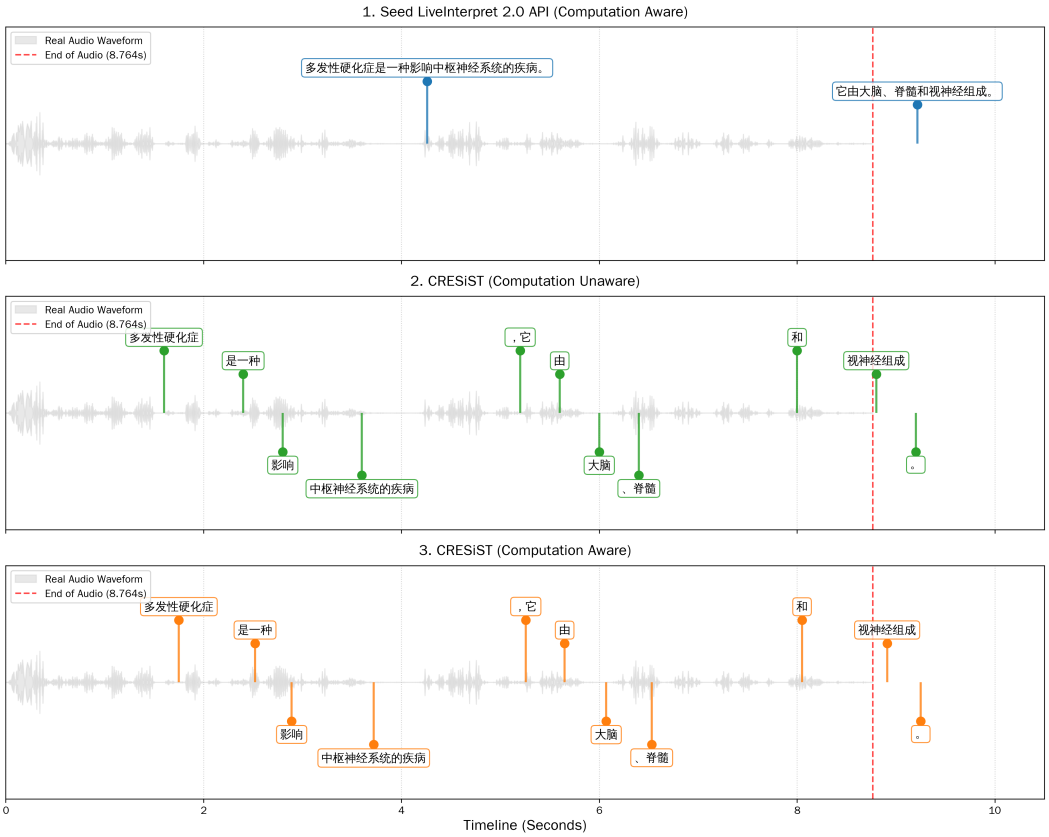


Figure 7: Translation timing visualization for a sample from the FLEURS EN-ZH test set. The timeline illustrates the latency advantage of our model, which performs fine-grained, word-level adaptive translation, in contrast to the coarse-grained, clause-level buffering strategy employed by the Seed LiveInterpret 2.0 API.

Table 5: Summary of existing datasets, models, and codebases used in this paper.

Asset Name	Type	License	Reference	URL
		Datasets		
Emilia	Dataset	CC BY 4.0	[29]	https://huggingface.co/datasets/amphion/Emilia-Dataset
WenetSpeech	Dataset	CC BY 4.0	[30]	https://www.openslr.org/121/
GigaSpeech	Dataset	Apache 2.0	[31]	https://github.com/SpeechColab/GigaSpeech
FLEURS	Dataset	CC BY 4.0	[33]	https://huggingface.co/datasets/google/fleurs
CoVoST 2	Dataset	CC BY-NC 4.0	[34]	https://huggingface.co/datasets/facebook/covost2
MuST-C	Dataset	CC BY-NC-ND 4.0	[41]	https://opendatalab.com/OpenDataLab/MuST-C
ACL 60/60	Dataset	CC BY 4.0	[42]	https://aclanthology.org/attachments/2023.iwslt-1.2.dataset.zip
		Models		
Qwen2.5-omni-7B	Model	Apache 2.0	[14]	https://huggingface.co/Qwen/Qwen2.5-Omni-7B
Qwen3-0.6B	Model	Apache 2.0	[43]	https://huggingface.co/Qwen/Qwen3-0.6B
Qwen3-235B-A22B-Instruct	Model	Apache 2.0	[43]	https://huggingface.co/Qwen/Qwen3-235B-A22B-Instruct-2507
Whisper-medium	Model	Apache 2.0	[28]	https://github.com/openai/whisper
Seamless Streaming	Model	CC BY-NC 4.0	[2]	https://huggingface.co/facebook/seamless-streaming
StreamUni	Model	Apache 2.0	[21]	https://huggingface.co/ICTNLP/StreamUni-Phi4
InfiniSST	Model	MIT	[7]	https://github.com/LeiLiLab/InfiniSST
		Codebases & Frameworks & Metrics		
PyTorch	Framework	BSD-3		https://pytorch.org/
PyTorch Lightning	Framework	Apache 2.0		https://github.com/Lightning-AI/pytorch-lightning
DeepSpeed	Codebase	Apache 2.0	[39]	https://github.com/deepspeedai/deepspeed
OmniSTEval	Codebase	Apache 2.0	[37]	https://github.com/pe-trik/OmniSTEval/tree/main
XCOMET-XXL	Metric	CC-BY-NC-SA 4.0	[35]	https://huggingface.co/Unbabel/XCOMET-XXL

G Broader Impacts

Our proposed CRESiST framework significantly advances simultaneous speech translation, which has the positive potential to break down cross-lingual barriers, facilitate global communication, and improve accessibility for non-native speakers or the hearing-impaired. However, we must also acknowledge potential negative societal impacts. First, like many LLM-based systems, our model may exhibit performance disparities across different demographics, potentially showing bias against non-standard accents or low-resource dialects. Relying on such systems in high-stakes scenarios (e.g., medical or legal settings) without human oversight could lead to critical misunderstandings. Furthermore, the extensive computational resources required for training such omni-modal LLMs contribute to a substantial carbon footprint. We encourage future research to address these fairness and efficiency challenges to ensure safe and equitable deployment.

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 main claims regarding the $O(1)$ KV cache reuse, the self-enhancing Actor-Critic pipeline, and state-of-the-art performance are explicitly stated in the Abstract and Introduction, and are rigorously supported by the theoretical formulation in Section 3 and empirical results in Section 4

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 explicitly discuss the limitations of our work, including data/domain bottlenecks and challenges with document-level static latency control, in Section 6

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The formal definition of the Markov Decision Process is provided in Section 3.1. Furthermore, the complete local convergence analysis of the reverse distillation loop, including all regularity assumptions and the proof of Theorem 1, is detailed in Appendix B.

   Guidelines:

- The answer [N/A] 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 provide detailed descriptions of the training pipeline, datasets, metrics, and baseline implementations in Section 4.1 and Appendix C. We also plan to open-source the code and detailed hyperparameter configurations to ensure full reproducibility.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: The code and checkpoints are not ready for open access. We will open-source it once the paper is accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Comprehensive experimental details, including dataset splits, baseline implementations, optimizer choices (AdamW), learning rate schedules, and LoRA configurations, are provided in Section 4.1 and Appendix C.

   Guidelines:

   - The answer [N/A] 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: Due to the immense computational cost associated with training and evaluating omni-modal Large Language Models, running multiple full training seeds to compute standard deviations or error bars was computationally prohibitive.

   Guidelines:

   - The answer [N/A] 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 explicitly disclose the computing hardware (a single node equipped with eight NVIDIA A100 80GB GPUs) and the environment details (PyTorch, DeepSpeed ZeRO Stage 2) used for all experiments in Appendix C.

Guidelines:

- The answer [N/A] 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: Our research conforms to the NeurIPS Code of Ethics. We solely utilize publicly available datasets and open-source models, and our discussion on broader impacts thoughtfully addresses potential ethical concerns such as fairness and bias.

Guidelines:

- The answer [N/A] 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 discuss both the positive impacts and potential negative impacts of our work in the Appendix section.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We have not released the model. And the risk of misuse is minimal. And we will add safeguards concerning the broader impacts when it is released.

Guidelines:

- The answer [N/A] 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 have cited the original papers and provided URLs for all existing assets (datasets, models, and code bases) used in our experiments. Detailed information, including versions and explicit licenses, is comprehensively summarized in Table 5 in Appendix F.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: The code and the checkpoints have not been released yet. And by the time it is released, it will follow the guidelines of Huggingface for documentation.

Guidelines:

- The answer [N/A] 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: [N/A]

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

Guidelines:

- The answer [N/A] 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: [N/A]

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

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, a declaration is not required.

Answer: [Yes]

Justification: The core methodology of this research inherently relies on Large Language Models. We explicitly and comprehensively describe our usage of LLMs in the paper, including using the Qwen2.5-omni-7B backbone as our Critic model (Section 3) and utilizing the Qwen3-235B model to generate pseudo-labels for training data construction (Section 4.1).

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.