SeqPO-SiMT: Sequential Policy Optimization for Simultaneous Machine Translation

Ting Xu[♠*], **Zhichao Huang**[♣†], **Jiankai Sun**[♢], **Shanbo Cheng**[♣†], **Wai Lam**[♠],
[♠]The Chinese University of Hong Kong, [♣]Bytedance, [♢]Stanford University,
xut0092@link.cuhk.edu.hk, jksun@stanford.edu,
{zhichao.huang, chengshanbo}@bytedance.com, wlam@se.cuhk.edu.hk

Abstract

We present Sequential Policy Optimization for Simultaneous Machine Translation (SeqPO-SiMT), a new policy optimization framework that defines the simultaneous machine translation (SiMT) task as a sequential decision making problem, incorporating a tailored reward to enhance translation quality while reducing latency. In contrast to popular Reinforcement Learning from Human Feedback (RLHF) methods, such as PPO and DPO, which are typically applied in single-step tasks, SeqPO-SiMT effectively tackles the multi-step SiMT task. This intuitive framework allows the SiMT LLMs to simulate and refine the SiMT process using a tailored reward. We conduct experiments on six datasets from diverse domains for En → Zh and Zh → En SiMT tasks, demonstrating that SeqPO-SiMT consistently achieves significantly higher translation quality with lower latency. In particular, SeqPO-SiMT outperforms the supervised fine-tuning (SFT) model by 1.13 [‡] points in COMET, while reducing the Average Lagging by 6.17 in the NEWSTEST2021 En → Zh dataset. While SiMT operates with far less context than offline translation, the SiMT results of SeqPO-SiMT on 7B LLM surprisingly rival the offline translation of high-performing LLMs, including Qwen-2.5-7B-Instruct and LLaMA-3-8B-Instruct.

1 Introduction

Simultaneous Machine Translation (SiMT) has made huge progress by leveraging large language models (LLMs) (Cheng et al., 2024; Koshkin et al., 2024). These approaches generally use partial translation data to finetune LLMs, enabling LLMs to translate partial source texts to target texts. However, the partial translation data are often generated using simple alignment tools, like heuristic methods (Ma et al., 2019) or attention mechanisms (Arivazhagan et al., 2019), which may introduce noise and degrade performance.

In parallel, Reinforcement Learning from Human Feedback (RLHF; Ouyang et al., 2022) has gained huge success in improving the performance of fine-tuned LLMs (DeepSeek-AI, 2025; Yang et al., 2024). RLHF is reward-driven and does not rely on partial translation data. Applying these techniques to SiMT appears to be a promising approach for improving its performance. However, we find that traditional RLHF methods like PPO (Schulman et al., 2017) and DPO (Rafailov et al., 2024) commonly work for a *single-step* process, while SiMT translates streaming inputs in a *multi-step* manner. The multi-step dependency between the source and target in SiMT is complex. First, the source texts of SiMT are provided step by step, and each step's source may have ambiguous meanings that require subsequent context for clarification. For example, the first step's source (*bark*) in Figure 1 is ambiguous, which requires subsequent source texts (*tree*) to clarify. Second, previous translation results can influence the overall translation quality. For example, in the second example of Figure 1, misinterpreting *"the bark of"* leads to errors in the overall translations. We claim that traditional RLHF methods commonly used in single-step reveal deficiencies in modeling the complex dependence relations in the multi-step SiMT setting.

To this end, we propose a new policy optimization method, **Seq**uential **P**olicy **O**ptimization (SeqPO), and apply it to the SiMT task. As shown in Figure 2, SeqPO-SiMT defines SiMT as a sequential decision-making process. To simulate the SiMT process, we segment a full sentence into multiple chunks and feed these chunks to the LLM sequentially. At each step, the model evaluates

[†]Corresponding authors.
[*]Work done when Ting Xu was interned at Bytedance.
[‡]Kocmi et al. (2024b) has indicated that an increase of 1 point in COMET represents a significant improvement.
[§]The work described in this paper is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200620).

Findings of the Association for Computational Linguistics: ACL 2025, pages 16107–16123
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

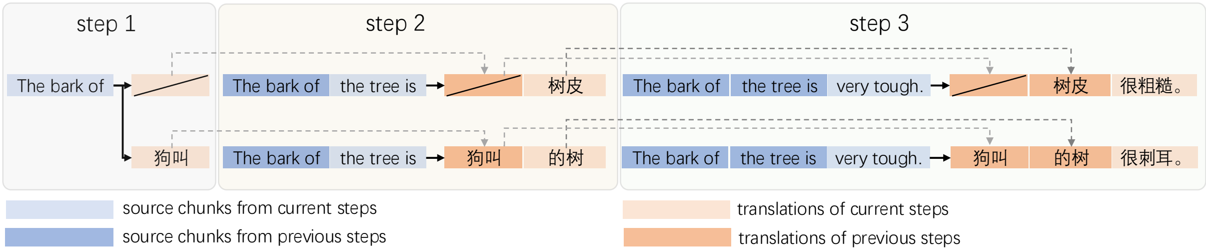


Figure 1: Two examples of SiMT, which translates streaming source texts into target texts. The source texts of SiMT are fed in a sequential manner. At each step, SiMT receives a new source text chunk and generates corresponding translations. Translations at each step can influence overall translations. A forward slash (/) indicates an empty translation, which means the model chooses not to translate and waits for subsequent texts.

the new chunk and translation history to decide whether to translate or wait for subsequent context. After completing all the steps, we construct a tailed reward to assess the entire SiMT process according to quality and latency. Finally, we optimize the SiMT process using policy gradient (Sutton and Barto, 2018). SeqPO-SiMT enables us to simulate the complex SiMT process as a multi-step decision problem and refine it based on quality and latency.

To validate the effectiveness of SeqPO-SiMT, we conduct experiments on extensive datasets for En $\rightarrow$ Zh and Zh $\rightarrow$ En SiMT tasks, including formal spoken language, informal spoken language, specialized knowledge domains, and news articles. Performance is measured using a range of comprehensive metrics. Extensive experimental results demonstrate that SeqPO-SiMT not only attains superior translation quality but also reduces latency. In low latency and high latency scenarios, the average COMET scores of SeqPO-SiMT are 1.3 and 1.25 points higher than those of the supervised fine-tuning (SFT) method, respectively. In particular, SeqPO-SiMT outperforms the SFT model by 1.13 points in COMET, while reducing the Average Lagging (AL) by 6.17 in NEWSTEST2021 En $\rightarrow$ Zh dataset. Because the SiMT task has only a limited amount of contexts while offline translation utilizes the full context, SiMT is more challenging than offline translation. Remarkably, the SiMT performance of SeqPO-SiMT is comparable to the offline translation performance of high-performing open-source models, such as Qwen-2.5-7B-Instruct (Yang et al., 2024) and LLaMA-3-8B-Instruct. These findings underscore the effectiveness of SeqPO-SiMT. We summarize our contribution as follows:

1. We define the RLHF process of SiMT LLMs as a sequential decision-making process to model the complex dependencies among steps in SiMT.
2. SeqPO-SiMT fuses both translation quality and latency into a reward. With a carefully designed fusion function, SeqPO-SiMT successfully improves the two metrics.
3. Extensive experiments demonstrate the superiority of SeqPO-SiMT, which not only enhances translation quality but also reduces the latency of SiMT. Furthermore, SeqPO-SiMT achieves SiMT translation quality comparable to the offline performance of strong LLMs like Qwen-2.5-7B-Instruct.

2 Sequential Policy Optimization

SeqPO-SiMT is a policy optimization framework that defines the SiMT task as a sequential decision making problem. It incorporates a tailored reward to enhance translation quality while reducing latency. This section first defines the basic components of SeqPO-SiMT: environment and policy. Then we describe the multi-step SiMT data sampling process. Finally, we describe how we optimize the policy model. The model architecture is illustrated in Figure 2, and the algorithm is described in Algorithm 1. For details on notations, please refer to Table 1.

Notation	Meaning
$\mathbf{x}$	full source sentence
$\mathbf{y}$	full target sentence
x_t	source text chunk
m	# words in each source text chunk
T	# steps in a SiMT process
y_t	target text chunk
B	# sampling times in each step
π_θ	policy model
π_{ref}	reference model
f_q, f_l	quality and latency scorer function
$\hat{q}, q$	quality score, normalized quality score
$\hat{l}, l$	latency score, normalized latency score
r_T	reward

Table 1: Basic notations of this work.

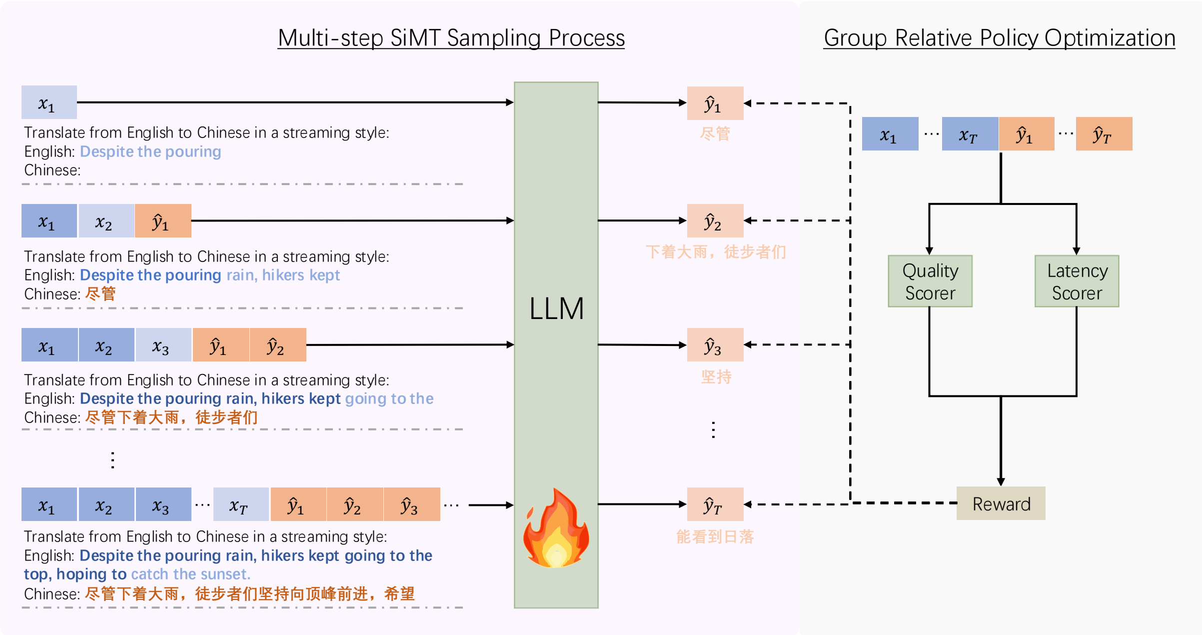


Figure 2: Model structure of SeqPO-SiMT. We first segment a full sentence in multiple chunks. At each step, we feed a new chunk to the LLM. Concatenating new source chunk and translation history, we query the LLM to generate new translations. At the end of the translation process, we evaluate the whole translation process by quality and latency. Light blue (Light orange) represents the new source chunk (new translation) for each step, and dark blue (dark orange) represents the source chunks (translations) from previous steps.

2.1 Basic Components

Environment. Consider a full source sentence $\mathbf{x} = (x_1, x_2, \cdots, x_T)$, where x_t is a source text chunk with m words, $T = \frac{|\mathbf{x}|}{m}$. At each time step, the environment emits a new text chunk in the full source sentence. After the environment emits the last text chunk, it reaches a terminal state and the SiMT process ends.

Policy. Following Cheng et al. (2024) and Koshkin et al. (2024), we employ an LLM as the policy model π_θ to generate translations. At time step t, the input to the LLM is based on existing source text chunks $x_{1:t} = (x_1, x_2, \cdots, x_t)$ and previous translation history $\hat{y}_{1:t-1} = (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{t-1})$. Concatenating all existing source text chunks and previous translations, the policy model produces translations as follows:

$$\hat{y}_t \sim \pi_\theta(\hat{y}_t | x_1, \cdots, x_t, \hat{y}_1, \cdots, \hat{y}_{t-1}). \quad (1)$$

Although every source chunk x_t is of pre-defined length m, the length of $\hat{y}_t$ is totally decided by the policy model. If the policy model decides not to translate and waits for more context, $\hat{y}_t$ will be empty, meaning its length is 0. If the policy model chooses to translate, $\hat{y}_t$ will consist of all the tokens generated by the policy model. This contrasts with rule-based methods. We allow the policy model to determine when to start translating and how much content to translate based on the context. This makes our policy more flexible than previous methods.

2.2 Multi-step SiMT Sampling Process

Traditional methods often fine-tune LLMs using partial translation data, which are derived from aligning full translation pairs. However, the alignment tools are often simple, such as heuristic methods or attention mechanisms, which may introduce noise and degrade performance. Instead of aligning the source and target sentences, we directly simulate and refine the SiMT process. We follow the multi-step nature of SiMT and conduct a multi-step SiMT sampling process as follows:

- The environment converts the source sentence $\mathbf{x}$ into T chunks: $\mathbf{x} = (x_1, x_2, \cdots, x_T)$, where T is the number of chunks. For example, we convert the source sentence in Figure 2 into five chunks: *("Despite the pouring", "rain, hikers kept", "going to the", "top, hoping to", "catch the sunset").*
- Initially, the environment emits the first text chunk x_1, and the policy produces B candi-

Algorithm 1 SeqPO-SiMT

Input: Translation quality metrics function f_q, Latency metrics function f_l, Source sentence $\mathbf{x}$, (Optional) Reference translation $\mathbf{y}$, Initial model π_{ref}, Policy model π_θ, Learning rate α

Output: Optimized model π_{θ^*}

for step $= 1$ to N **do**
  Sample $\mathbf{x}$ from training dataset and segment $\mathbf{x}$ into T chunks $\mathbf{x}_1, \cdots, \mathbf{x}_T$
  for $t = 1$ to T **do**
    Sample B translations from the policy π_θ: $\hat{y}_t^i \sim \pi_\theta(\hat{y}_t^i | x_1, \cdots, x_t; \hat{y}_1^i, , \cdots, \hat{y}_{t-1}^i)$, $i = 1, \cdots, B$.
  end for
  for $i = 1$ to B **do**
    Gather the SiMT process: $\hat{\mathbf{y}}^i = (\hat{y}_1^i, \cdots, \hat{y}_T^i)$
    Compute the quality and latency score: $\hat{q}^i = f_q(\mathbf{x}, \hat{\mathbf{y}}^i, \mathbf{y})$, $\hat{L}^i = f_l(\mathbf{x}, \hat{\mathbf{y}}^i, \mathbf{y})$
    Normalize $\hat{q}^i$, $\hat{L}^i$ to get q^i, L^i
    Calculate overall reward $r_T^i = \lambda q^i - L^i$ and KL divergence $D_t^i = \log \frac{\pi_\theta(\hat{y}_t^i | x_1^i, \cdots, x_t^i; \hat{y}_1^i, ...\hat{y}_{t-1}^i)}{\pi_{\text{ref}}(\hat{y}_t^i | x_1^i, ..., x_t^i; \hat{y}_1^i, ..., \hat{y}_{t-1}^i)}$
  end for
  Calculate policy gradient $\mathbf{g} = \frac{1}{B} \sum_{i=1}^{B} \sum_{t=1}^{T} (r_T^i - \beta D_t^i) \nabla_\theta \log \pi_\theta(\hat{y}_t^i | x_1, \cdots, x_t; \hat{y}_1^i, ...\hat{y}_{t-1}^i)$
  Update model $\theta = \theta + \alpha \mathbf{g}$
end for

date translations $\hat{y}_1^i$ based on x_1: $\hat{y}_1^i \sim \pi_\theta(x_1)$, for $i = 1, \cdots, B$, where B is the number of sampling times. As shown in Figure 2, the model translates the first chunk *"Despite the pouring"* into *"尽管"*.

- At each subsequent time step t, the environment emits a new text chunk, x_t. The policy model then produces translations based on Equation (1), considering both the new text chunk and the translation history. For example, as illustrated in step 2 of Figure 2, concatenating the previous text chunk ("*Despite the pouring*") with the current text chunk ("*rain hikers kept*") yields the source text ("*Despite the pouring rain, hikers kept*"). This is then concatenated with the previous translation ("尽管"). We fill in the source texts and translation into the template, construct a prompt for the model to generate translation, and obtain the result (*"下着大雨，徒步者们"*).

- Keep running the last step until all source texts are fully translated, i.e., $t > T$.

In the end, by aggregating all the translation steps, we have $\hat{\mathbf{y}}^i = (\hat{y}_1^i, \hat{y}_2^i, \cdots, \hat{y}_T^i)$. For example in Figure 2, the final translation is *"尽管下着大雨，徒步者们坚持向顶峰前进，希望能看到日落"*.

2.3 Reward

After the multi-step sampling process, we evaluate the entire SiMT process and further refine it. Concretely, an accurate final reward is provided to policy model π_θ at the last step T as r_T, and the policy model can adopt r_T to optimize its sequential decision from step 1 to step T. Specifically, we assess SiMT's performance from two perspectives: quality and latency. Translation quality can be evaluated using existing metrics for machine translation, such as COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020), or the RLHF reward model (Ouyang et al., 2022). Latency can be measured using metrics like Average Lagging (AL; Ma et al., 2019) and Length-Adaptive Average Lagging (LAAL; Papi et al., 2022).

Our primary objective is to identify a translation process that achieves not only high quality but also maintains low latency. However, there is an inevitable trade-off between translation quality and latency. On one hand, a conservative policy that waits longer may have higher translation quality. On the other hand, a radical policy with small latency may be short of translation quality. In order to balance these two metrics and unify them into the same scale, we propose to incorporate reward normalization and latency truncation into a fused reward to measure both quality and latency. Specifically, we define the reward as follows:

$$\begin{aligned}
\hat{q}^i &= f_q(\mathbf{x}^i, \hat{\mathbf{y}}^i, \mathbf{y}), \quad \hat{L}^i = f_l(\mathbf{x}^i, \hat{\mathbf{y}}^i, \mathbf{y}), \\
q^i &= \frac{\hat{q}^i - \text{mean}(\{\hat{q}^1, \cdots, \hat{q}^B\})}{\text{std}(\{\hat{q}^1, \cdots, \hat{q}^B\})}, \\
L^i &= \max(m, \frac{\hat{L}^i - \text{mean}(\{\hat{L}^1, \cdots, \hat{L}^B\})}{\text{std}(\{\hat{L}^1, \cdots, \hat{L}^B\})}), \\
r_T^i &= \lambda q^i - L^i,
\end{aligned} \tag{2}$$

where f_q and f_l are quality and latency scorer functions, $\mathbf{y}$ is the gold translation, λ is a hyper-parameter that decides the trade-off between quality and latency, the superscript i means the i-th sample from the mini-batch. In the above equation, we normalize the values of quality and latency because they have different scales. By nomalization, we can convert these two metrics to the same scale, thereby facilitating their fusion. And we truncate the L by the chunk size of $\mathbf{x}$ to avoid overfitting to the latency score. In addition to the reward, we also add commonly used KL constraint (Schulman et al., 2017) to keep the policy model stable in the training process. Thus, the final objective function of SeqPO-SiMT is

$$\begin{aligned}
J(\pi_\theta) =& \mathbb{E}_{\hat{y}_1, \cdots, \hat{y}_T \sim \pi_\theta(\cdot), \mathbf{x} \sim p_{\text{data}}} \sum_{t=1}^{T} [r_T \\
&- \beta \log \frac{\pi_\theta(\hat{y}_t | x_1, \cdots, x_t; \hat{y}_1, \cdots, \hat{y}_{t-1})}{\pi_{\text{ref}}(\hat{y}_t | x_1, \cdots, x_t; \hat{y}_1, \cdots, \hat{y}_{t-1})}],
\end{aligned} \tag{3}$$

where p_{data} is the training dataset.

2.4 Optimization

As for the optimization method, our approach can be applied to various policy gradient optimization methods. In this paper, Group Relative Policy Optimization (GRPO; Shao et al., 2024), which has demonstrated its effectiveness and efficiency in various large language models, such as DeepSeekMath (Shao et al., 2024) and DeepSeek-R1 (DeepSeek-AI, 2025), is selected as our optimization method. Specifically, we sample B trajectories for each to calculate the baseline reward. In SeqPO-SiMT, we choose GRPO over the popular PPO (Schulman et al., 2017) because of the following reasons:

1. **Resources Efficiency**: GRPO utilizes a grouped average as its baseline, whereas PPO incorporates a new critic model for baseline computation, which increases memory and computational overhead. In our context, translation quality and latency require two separate critic models, which makes the memory requirements unaffordable.

2. **Accurate Metric**: In SiMT, latency is a rule-based metric. PPO uses a neural reward model may introduce more noise and complicate the training pipeline (DeepSeek-AI, 2025).

3 Experimental Setup

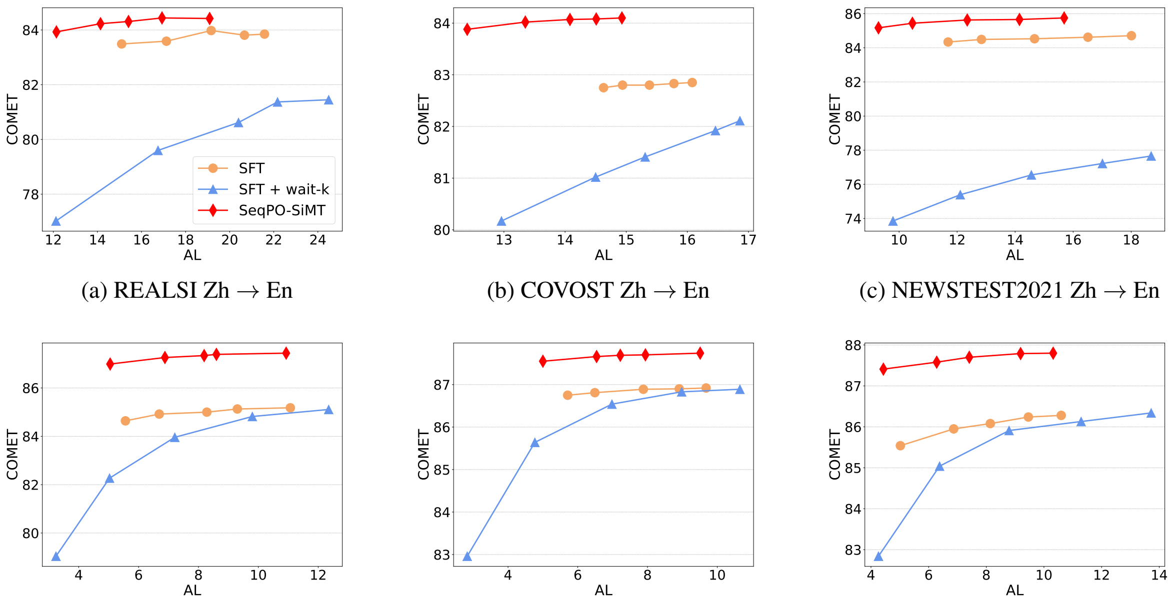
Models and Training Data. We focus on Chinese and English, a language pair with significant structural differences. For the Zh $\rightarrow$ En setup, we utilize WeNet (Zhang et al., 2022) and WMT19 (Barrault et al., 2019) as our training data. For the En $\rightarrow$ Zh setup, we employ GigaST (Chen et al., 2021; Ye et al., 2022) as the training data. We use Qwen-2.5-7B (Yang et al., 2024) as our backbone.

Because the base LLM lacks SiMT capabilities, we initially construct a dataset for warm-up, the process is similar to the previous works (Koshkin et al., 2024; Cheng et al., 2024). Specifically, we randomly select 40,000 data samples from the training datasets and construct partial translation data by prompting LLMs. Details about the SFT data construction process is shown in Appendix A.2.

Evaluation Benchmarks. To comprehensively validate the effectiveness of SeqPO-SiMT, we conduct experiments on datasets from diverse domains. For the Zh $\rightarrow$ En setup, we evaluate SeqPO-SiMT on COVOST (Wang et al., 2020), REALSI (Cheng et al., 2024), and NEWSTEST2021 on Zh $\rightarrow$ En, including specialized knowledge, informal spoken language, and news articles. For the En $\rightarrow$ Zh setup, we evaluate SeqPO-SiMT on REALSI, MUSTC (Di Gangi et al., 2019), and NEWSTEST2021, including informal spoken language, formal spoken language, and news articles.

Implementation Details. We set the hyperparameter as follows: For λ parameter, we sample 50 SiMT data and score them with a range of λ values, then we manually evaluate the scoring performance for each λ value and select $\lambda = 2$ because it can balance between quality and latency. For other hyper-parameters, we set $B = 5$, $\beta = 0.02$ for En $\rightarrow$ Zh, $\beta = 0.1$ for Zh $\rightarrow$ En. COMET and AL are chosen as the translation quality reward and latency reward, respective. Due to space limitations, we put other implementation details in Appendix A.1.

Evaluation Metrics. We measure the performance through comprehensive metrics of translation quality and latency. For translation quality, we employ COMET (Rei et al., 2020), BLEURT



(a) REALSI Zh → En (b) COVOST Zh → En (c) NEWSTEST2021 Zh → En

(d) REALSI En → Zh (e) MUSTC En → Zh (f) NEWSTEST2021 En → Zh

Figure 3: COMET v.s. AL on Zh → En and En → Zh SiMT tasks.

Dataset	Method	Low latency					High latency				
		BLEURT ↑	COMET↑	GPT-4 ↑	AL ↓	LAAL ↓	BLEURT ↑	COMET ↑	GPT-4 ↑	AL ↓	LAAL ↓
REALSI	SFT	64.14	83.49	83.24	15.1	15.87	64.8	83.77	84.07	**18.27**	**18.94**
	SFT+wait-k	59.37	79.6	78.9	16.75	16.97	61.2	80.97	79.87	22.17	22.37
	SeqPO-SiMT	**65.93**	**84.23**	**85.49**	**14.14**	**14.59**	**66.24**	**84.42**	**85.92**	19.09	19.44
COVOST	SFT	60.17	82.75	75.47	14.63	14.72	60.33	82.85	75.86	16.08	16.13
	SFT+wait-k	57.06	80.17	71.47	12.96	13.06	59.16	81.92	73.93	16.46	16.49
	SeqPO-SiMT	**63.01**	**83.95**	**79.86**	**12.91**	**13.01**	**63.28**	**84.1**	**80.13**	**14.93**	**14.99**
NEWS	SFT	65.01	84.34	86.02	10.18	12.94	65.69	84.7	86.54	17.18	18.33
	SFT+wait-k	50.53	73.85	72.99	9.79	9.93	55.03	77.66	78.62	18.69	18.91
	SeqPO-SiMT	**66.67**	**85.17**	**87.67**	**9.29**	**9.92**	**67.94**	**85.75**	**89.17**	**15.69**	**16.34**

Table 2: Detailed results in low and high latency levels on Zh → En SiMT tasks. NEWS is an abbreviation for NEWSTEST2021. The best results are highlighted in bold.

(Sellam et al., 2020), and GPT-4 as metrics. Detailed prompt for GPT-4 is described in Appendix A.1. For latency, we use Average Lagging (AL; Ma et al., 2019) and Length-Adaptive Average Lagging (LAAL; Papi et al., 2022) as metrics.

Baselines. To demonstrate the effectiveness of our method, we compare the results with the SFT method and wait-k (Ma et al., 2019), a commonly used method in SiMT.

- **SFT** trains exclusively on partial translation SFT data, which is the same as the SFT data of SeqPO-SiMT.
- **SFT + wait-k** uses the same model as SFT. During inference, it uses the wait-k policy (Ma et al., 2019). When receiving the first k tokens, it waits and does not generate any tokens. After the first k tokens, every time it receives a new token, it will produce a new token for translation.

4 Experimental Results

In this section, we present the main results of our experiments, emphasizing the consistent and significant superior performance of SeqPO-SiMT across various benchmarks and metrics. We compare with offline translations to emphasize the high translation quality of SeqPO-SiMT. Then we provide an in-depth understanding of how SeqPO-SiMT concurrently improves quality and latency.

Dataset	Method	Low latency					High latency				
		BLEURT ↑	COMET↑	GPT-4 ↑	AL ↓	LAAL ↓	BLEURT ↑	COMET ↑	GPT-4 ↑	AL ↓	LAAL ↓
REALSI	SFT	61.84	84.64	87.45	5.56	5.77	63.19	85.18	88.11	10.34	11.22
	SFT + wait-k	58.92	82.27	84.06	**5.03**	**5.14**	62.46	84.82	88.33	**9.8**	**9.96**
	SeqPO-SiMT	**64.84**	**86.99**	**87.53**	5.05	**5.14**	**66.41**	**87.44**	**89.07**	10.93	11.04
MUSTC	SFT	65.84	86.75	91.84	5.71	5.95	66.12	86.92	92.45	9.68	9.84
	SFT + wait-k	63.94	85.64	89.15	**4.77**	**4.95**	66.04	86.83	92.24	**8.98**	**9.16**
	SeqPO-SiMT	**66.76**	**87.55**	**92.1**	5	5.15	**67.46**	**87.72**	**93.18**	9.51	9.62
NEWS	SFT	61.12	85.54	90.99	5.02	5.9	62.28	86.28	92.28	10.6	11.29
	SFT + wait-k	57.1	82.84	83.58	**4.26**	**4.85**	62.25	86.13	92.2	11.29	11.9
	SeqPO-SiMT	**63.37**	**87.41**	**91.63**	4.43	5.02	**64.62**	**87.78**	**93.31**	**10.32**	**10.83**

Table 3: Detailed results in different latency level on En $\rightarrow$ Zh SiMT tasks. NEWS is an abbreviation for NEWSTEST2021. The best results are highlighted in bold.

4.1 Main Results

SeqPO-SiMT consistently and significantly outperforms other methods in both quality and latency. The COMET and AL performance for different methods are demonstrated in Figure 3. The results show that SeqPO-SiMT consistently achieves a superior translation quality across all latency levels and all datasets, particularly in the low latency level. Other figures about COMET v.s. LAAL, BLEURT v.s. AL, and BLEURT v.s. LAAL are available in appendix B.1.

We provide detailed numerical results in Table 2 and Table 3. Specifically, we first divide the latency into two groups, low latency and high latency. To avoid overfitting the model to a specific metric, we provide many metrics for quality and latency. In both high-latency and low-latency scenarios, the translation quality of SeqPO-SiMT is significantly higher than that of other methods, achieving consistent improvements in BLEURT, COMET, and GPT-4. On average, The COMET scores of SeqPO-SiMT are 1.3 and 1.25 higher than those of the supervised finetuning (SFT) method in low latency and high latency scenarios, respectively. Kocmi et al. (2024a) shows that an increase of 1 point in COMET represents a significant improvement, so the progress we made is impressive. In particular, SeqPO-SiMT outperforms the SFT model by 1.13 points in COMET, while decreasing the AL by 6.17 in NEWSTEST2021 En $\rightarrow$ Zh. SeqPO-SiMT also achieves superior performance on BLEURT and GPT-4's evaluation, indicating that SeqPO-SiMT genuinely improves translation quality rather than overfitting to the COMET metric.

SeqPO-SiMT achieves more stable COMET with varied latency. As shown in Figure 3, as the latency increases, SeqPO-SiMT exhibits more stable performance compared to the wait-k strategy in terms of COMET scores. Particularly in low-latency scenarios, the translation quality of our method is significantly higher than that of the wait-k strategy.

4.2 Comparison with Offline Models

SeqPO-SiMT outperforms offline SFT model and LLaMA-3-8B-Instruct, achieving comparable results to the offline Qwen-2.5-7B-Instruct. As most of the previous SiMT algorithms (Koshkin et al., 2024; Yu et al., 2025, 2024; Guo et al., 2024) use different benchmarks with different base models, it is hard to fairly compare previous methods in our setting. To further show that SeqPO-SiMT achieves SoTA performance in 7B LLMs, we compare the SiMT results of SeqPO-SiMT with the offline results of the high-performing open-source model, i.e., LLaMA3-8B-Instruct and Qwen2.5-7B-Instruct. For the SiMT translation results, we use results with low latency. The results are shown in Table 4. We can see that: SeqPO-SiMT even surpasses the translation quality of offline SFT model, Qwen2.5-7B-Instruct and LLaMa3-8B-Instruct, indicating that our method significantly boosts translation quality and achieving the SoTA performance in 7B model size.

4.3 In-depth Analysis of Quality and Latency

Change of COMET and AL during training. We further study the changes in COMET and AL during the training process, as shown in Figure 4a. During training, AL first decreases rapidly and then slowly increases, while COMET initially rises and then stabilizes. We believe this is because AL is easy to fit (if the model outputs many meaningless tokens each time it receives input, AL will also decrease). Therefore, during the training process,

	BLEURT	COMET	GPT-4
	MUSTC En → Zh		
SFT	65.84	86.75	92.27
offline SFT	<u>66.28</u>	<u>86.94</u>	<u>92.61</u>
offline LLaMa3	60.43	83.78	86.98
offline Qwen2.5	65.47	86.49	91.97
SeqPO-SiMT	**66.76**	**87.55**	**92.7**
	REALSI Zh → En		
SFT	64.14	83.49	83.74
offline SFT	<u>65.06</u>	<u>83.92</u>	83.72
offline LLaMa3	63.27	81.19	85.66
offline Qwen2.5	64.08	82.14	**85.79**
SeqPO-SiMT	**65.93**	**84.23**	<u>85.59</u>

Table 4: Comparison of translation quality between SeqPO-SiMT's SiMT results and other LLMs' offline translation results. The best results and the second-best are highlighted by bold and underscore respectively.

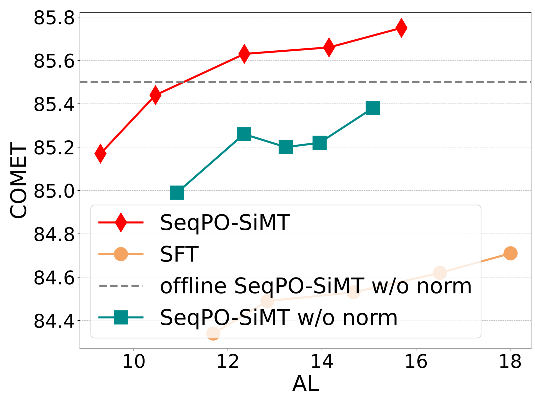
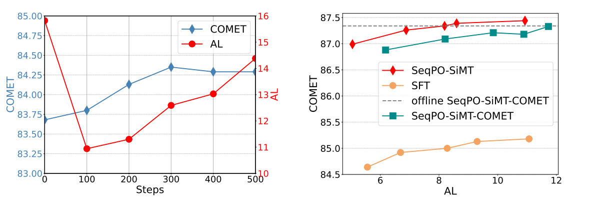


Figure 5: Ablation study on the normalization module.

the model first reduces latency. Then to increase the overall reward, the model further refines its translation quality.

SeqPO-SiMT boosts both quality and latency without compromise. We explore the maximum translation quality achievable when we only optimize translation quality, i.e., only optimize the COMET score (SeqPO-SiMT-COMET). The results, as shown in Figure 4b, indicate that SeqPO-SiMT has lower latency than SeqPO-SiMT-COMET at the same translation quality. Notably, different from previous methods (Koshkin et al., 2024; Zhang, 2024) which sacrifice quality or latency, SeqPO-SiMT can achieve comparable translation quality with offline SeqPO-SiMT-COMET, demonstrating that our method effectively boosts quality and latency without compromise.



(a) Change of COMET and AL during Training.

(b) Comparison with only optimizing COMET.

Figure 4: In-depth analysis of quality and latency.

4.4 Ablation Study on Reward

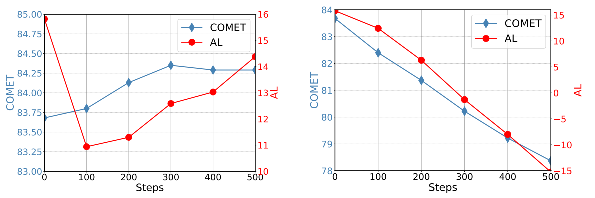
Simple truncation effectively avoids overfitting to latency. Our reward design incorporates a carefully designed truncation module. Figure 6 compares the training dynamics with and without this module. As shown in Figure 6b, the absence of the truncation mechanism leads to a sharp decline in AL during model training, even reaching negative values, accompanied by a persistent decrease in the COMET score. Upon inspecting the model's translation output, we observe a tendency to produce numerous meaningless tokens. We hypothesize that this occurs because the model can easily reduce latency by simply outputting such tokens, thereby artificially increasing the overall reward and leading to overfitting to the latency metric. However, as Figure 6a illustrates, the model with the truncation module effectively avoids AL overfitting while balancing latency and translation quality. These results demonstrate that truncation, despite its inherent simplicity, effectively mitigates overfitting to latency.

Normalization can better balance quality and latency. Our method incorporates normalization for both quality and latency scores. Figure 5 presents the results of the ablation study on the normalization module. As illustrated in the figure, the removal of the normalization module results in a degradation of quality at the same latency levels. These findings demonstrate that normalization, despite its inherent simplicity, effectively balances quality and latency.

Additional Results. Due to space limitations, we have included additional experimental results in the appendix. These results encompass offline translation outcomes, a comparison with traditional encoder-decoder models, an analysis under low-latency conditions, a human evaluation of SiMT results, and a case study.

	BLEURT	COMET	GPT-4
	MUSTC En $\to$ Zh		
offline SFT	66.28	86.94	92.61
offline LLaMa3	60.43	83.78	86.98
offline Qwen2.5	65.47	86.49	91.97
offline SeqPO-SiMT	**67.59**	**87.74**	**93.33**
	REALSI Zh $\to$ En		
offline SFT	65.06	83.92	83.72
offline LLaMa3	63.27	81.19	85.66
offline Qwen2.5	64.08	82.14	85.79
offline SeqPO-SiMT	**66.82**	**84.62**	**86.79**

Table 5: Comparison of translation quality between SeqPO-SiMT and other LLMs' offline translation results. The best results and the second-best are highlighted by bold and underscore respectively.



(a) Change of COMET and AL with truncation module during Training.

(b) Change of COMET and AL without truncation module during Training.

Figure 6: Ablation study on the truncation module.

5 Related Work

5.1 Simultaneous Machine Translation

Existing SiMT methods can be divided into three categories: rule-based, alignment-based, and reinforcement learning-based. **Rule-based** methods rely on heuristics. For example, Ma et al. (2019) proposes a wait-k policy to wait for k tokens before translating. Cho and Esipova (2016) wait until the source input provides more information. **Alignment-based** methods adapt the full sentence translation model to SiMT by aligning the source sentence and target translation at the word level. For example, Zheng et al. (2019) convert full sentence translation pairs to partial translation pairs through some heuristic alignment rules, then use the partial data to train the SiMT model. Arivazhagan et al. (2019), Communication et al. (2023), and Zhang et al. (2024) integrate an alignment module into the model, like monotonic attention and Connectionist Temporal Classification (Graves et al., 2006) to align the source and target sentence. However, we claim that the alignment process is highly noisy. **Reinforcement learning-based** methods build upon an existing offline translation model by adding a new read/write policy (Grissom II et al., 2014; Satija, 2016; Gu et al., 2017; Alinejad et al., 2018; Ive et al., 2021). These methods typically treat the translation model as the environment and focus exclusively on optimizing the read/write policy. However, the translation model is only trained on full-sequence translation pairs. They sacrifice a significant amount of translation quality to improve latency, resulting in poor experimental outcomes.

5.2 Reinforcement Learning from Human Feedback

RLHF is a technique that aligns LLMs with human preferences. There are many RLHF methods shown to be effective, such as PPO (Schulman et al., 2017)), DPO (Rafailov et al., 2024), and GRPO (Shao et al., 2024). RLHF has been successfully applied to various applications, such as ensuring safety (Dai et al., 2023) and mitigating toxicity (Korbak et al., 2023). However, they mainly tackle single-step generation processes while SiMT is a multi-step decision making process.

6 Conclusion

In this work, we introduce Sequential Policy Optimization for Simultaneous Machine Translation (SeqPO-SiMT), a novel framework that defines the SiMT task as a sequential decision-making problem. Specifically, we conduct multi-step SiMT data sampling processes and optimize according to quality and latency. This intuitive framework allows the SiMT model to effectively refine the translation quality and reduce latency. We conduct extensive experiments on six datasets from the diverse domains for En $\to$ Zh and Zh $\to$ En SiMT tasks. Experimental results demonstrate that SeqPO-SiMT consistently achieves significantly higher translation quality with lower latency. Moreover, SeqPO-SiMT achieves comparable translation quality as high-performing offline translation models, such as Qwen-2.5-7B-Instruct and LLaMA-3-8B-Instruct.

7 Limitations

While this work achieves good performance on 7B LLMs, we cannot ensure that this method can scale to extremely large LLMs, like Deepseek-V3-671B. Future works can scale to larger language models. The current method is still bilingual, and as the number of languages increases, balancing quality and latency across multiple languages presents significant challenges. Future research could expand to multilingual SiMT.

References

Ashkan Alinejad, Maryam Siahbani, and Anoop Sarkar. 2018. Prediction improves simultaneous neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3022–3027, Brussels, Belgium. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain ASR corpus with 10, 000 hours of transcribed audio. In *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021, Brno, Czechia, August 30 - September 3, 2021*, pages 3670–3674. ISCA.

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

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *CoRR*, abs/1606.02012.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation. *Preprint*, arXiv:2312.05187.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 369–376. ACM.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1352, Doha, Qatar. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Shoutao Guo, Shaolei Zhang, and Yang Feng. 2024. Decoder-only streaming transformer for simultaneous translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8851–8864, Bangkok, Thailand. Association for Computational Linguistics.

Julia Ive, Andy Mingren Li, Yishu Miao, Ozan Caglayan, Pranava Madhyastha, and Lucia Specia.

2021. Exploiting multimodal reinforcement learning for simultaneous machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3222–3233, Online. Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024a. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Christian Federmann, and Matt Post. 2024b. Navigating the metrics maze: Reconciling score magnitudes and accuracies. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1999–2014, Bangkok, Thailand. Association for Computational Linguistics.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17506–17533. PMLR.

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

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

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. *CoRR*, abs/2009.09025.

Harsh Satija. 2016. Simultaneous machine translation using deep reinforcement learning. In *ICML 2016 Workshop on Abstraction in Reinforcement Learning*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Richard S Sutton and Andrew G Barto. 2018. Reinforcement learning: an introduction, 2nd edn. adaptive computation and machine learning.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020. CoVoST: A diverse multilingual speech-to-text translation corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Rong Ye, Chengqi Zhao, Tom Ko, Chutong Meng, Tao Wang, Mingxuan Wang, and Jun Cao. 2022. Gigast: A 10,000-hour pseudo speech translation corpus. *arXiv preprint arXiv:2204.03939*.

Donglei Yu, Xiaomian Kang, Yuchen Liu, Yu Zhou, and Chengqing Zong. 2024. Self-modifying state modeling for simultaneous machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9781–9795, Bangkok, Thailand. Association for Computational Linguistics.

Donglei Yu, Yang Zhao, Jie Zhu, Yangyifan Xu, Yu Zhou, and Chengqing Zong. 2025. SimulPL: Aligning human preferences in simultaneous machine translation. In *The Thirteenth International Conference on Learning Representations*.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, Di Wu, and Zhendong Peng. 2022. WENETSPEECH: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 6182–6186. IEEE.

Jiarui Zhang. 2024. Guided profile generation improves personalization with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4005–4016, Miami, Florida, USA. Association for Computational Linguistics.

Shaolei Zhang, Qingkai Fang, Shoutao Guo, Zhengrui Ma, Min Zhang, and Yang Feng. 2024. StreamSpeech: Simultaneous speech-to-speech translation with multi-task learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8964–8986, Bangkok, Thailand. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022. Information-transport-based policy for simultaneous translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 992–1013, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2023. Hidden markov transformer for simultaneous machine translation. *Preprint*, arXiv:2303.00257.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

A Experimental Settings

In this section, we offer additional implementation details regarding our experiments, as well as a comprehensive overview of the SFT data construction process involved in this work.

A.1 Implementation Details

We use the pretrained model provided by HuggingFace and run all the experiments on NVIDIA A100 GPU with pytorch. And we use the code provided in trl * to train the SFT model. The hyper-parameters of the training and generation used in our experiments are shown in Table 6. During multi-step SiMT sampling, we randomly sample five translations with greedy search and temperature $= 1.0$. Figure 9 illustrates the template used to score translations from different models. Translation results were evaluated using gpt-4o-2024-08-06.

Transformer Hyper-parameters	
optimizer	AdamW
adam-β	(0.9, 0.999)
gradient clipping	1.0
gradient accumulation steps	8
learning rate	1×10^{-6}
precision	bf16
batch size	32
Generation Hyper-parameters	
temperature	1.0
max new tokens	60
top_k	0
top_p	1.0
do_sample	True

Table 6: Hyper-parameters of the SiMT model.

A.2 SFT Data Construction

For our SFT data, we randomly sample 40,000 instances from the training datasets. Specifically, we extract 40,000 Zh $\rightarrow$ En translation examples from WMT19 and WENET, and another 40,000 En $\rightarrow$ Zh translation examples from GigaST. We use these full sentence translation pairs to construct partial translation pairs. The process is illustrated in Figure 8. It begins with aligning the source and target sentences. We prompt LLMs to segment both sentence and target sentences into chunks and pair semantically equivalent text chunks. As shown in Figure 8, we segment the source and target sentences into four chunks. Finally, we concatenate the aligned text chunks to generate partial translation data.

*https://github.com/huggingface/trl

B Experimental Results

In this section, we provide detailed main results, offline SeqPO-SiMT's results, ablation study on the reward function, and a qualitative case study.

B.1 Detailed Main Results

To comprehensively present the changes in translation quality with respect to translation delay, we have provided the variation graphs of BLEURT v.s. AL, BLEURT v.s. LAAL, and COMET v.s. AL in Figure 10, Figure 11, and Figure 12, respectively. The results show that our proposed SeqPO-SiMT achieves a higher translation quality across all latency levels on all datasets, particularly in the low latency level.

B.2 Offline SeqPO-SiMT

We perform offline translation (full sentence translation) using SeqPO-SiMT as base model and compare the results with those of high-performing LLMs in Table 5. The results demonstrate state-of-the-art (SOTA) performance in offline translation.

B.3 Comparison with Traditional Encoder-Decoder Models

We compare the performance of SeqPO-SiMT with traditional encoder-decoder models, including HMT (Zhang and Feng, 2023), ITST (Zhang and Feng, 2022), and SM^2 (Yu et al., 2024), on NIST 2003-2006 datasets in Table 7. It is evident that SeqPO-SiMT significantly outperforms other methods across various latency levels. We believe this is due to two main reasons. First, SeqPO-SiMT possesses strong foundational capabilities. Second, SeqPO-SiMT uses both quality and latency as rewards, guiding its reinforcement learning process in a reward-oriented manner, which directly enhances translation quality and reduces translation latency.

B.4 Analysis under very low latency in En $\rightarrow$ Zh Setting

Simply using SeqPO-SiMT in the En $\rightarrow$ Zh setting does not achieve low latency; however, our method can be combined with traditional read/write policies like wait-k to enable SiMT under low-latency conditions. The results are shown in Table 8,

	Low Latency		High Latency	
	COMET	AL	COMET	AL
HMT	78.73	**6.11**	79.85	**11.35**
ITST	79.27	6.94	79.93	11.40
SM^2	79.91	6.19	80.45	11.61
SeqPO-SiMT	**82.80**	6.32	**83.37**	11.69

Table 7: Comparison results with traditional encoder-decoder methods.

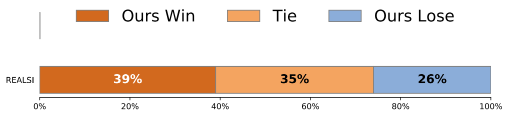


Figure 7: Human Evaluation between SeqPO-SiMT and the SFT model.

demonstrating that SeqPO-SiMT significantly outperforms SFT in translation quality at low latency, highlighting the effectiveness of SeqPO-SiMT.

B.5 Human Evaluation

To verify that SeqPO-SiMTaligns with human preference, we randomly sample 100 sentences from the REALSI Zh $\rightarrow$ En dataset and conduct a manual evaluation by professional simultaneous interpreters. The results in Figure 7 show that SeqPO-SiMT achieves a higher win rate than the SFT model, indicating stronger alignment with human preference.

B.6 Case Study

We analyze the translation results of SFT and SeqPO-SiMT presented in Table 9. Our findings indicate that, given the same source inputs, SeqPO-SiMT exhibits lower latency compared to SFT, allowing it to deliver translations promptly after receiving semantically complete source texts. Furthermore, the SFT model fails to incorporate the information from 百科辞典 in the Zh $\rightarrow$ En translation and redundantly repeats the phrase *who have been sentenced to do these very harsh sentences* in the En $\rightarrow$ Zh translation. In contrast, our method accurately translates the source sentence. This case study demonstrates that SeqPO-SiMT can achieves higher translation quality alongside lower latency.

Dataset	Method	BLEURT	COMET	AL	LAAL
REALSI	SFT + wait-k	45.73	63.35	4.08	4.11
	SeqPO-SiMT + wait-k	**55.67**	**75.98**	**4.06**	**4.08**
COVOST	SFT + wait-k	45.03	65.87	3.88	3.95
	SeqPO-SiMT + wait-k	**55.97**	**77.69**	**3.87**	**3.91**
NEWS	SFT + wait-k	43.52	66.63	4.83	4.88
	SeqPO-SiMT + wait-k	**53.53**	**76.33**	**4.51**	**4.59**

Table 8: Analysis under very low latency in Zh $\rightarrow$ En.

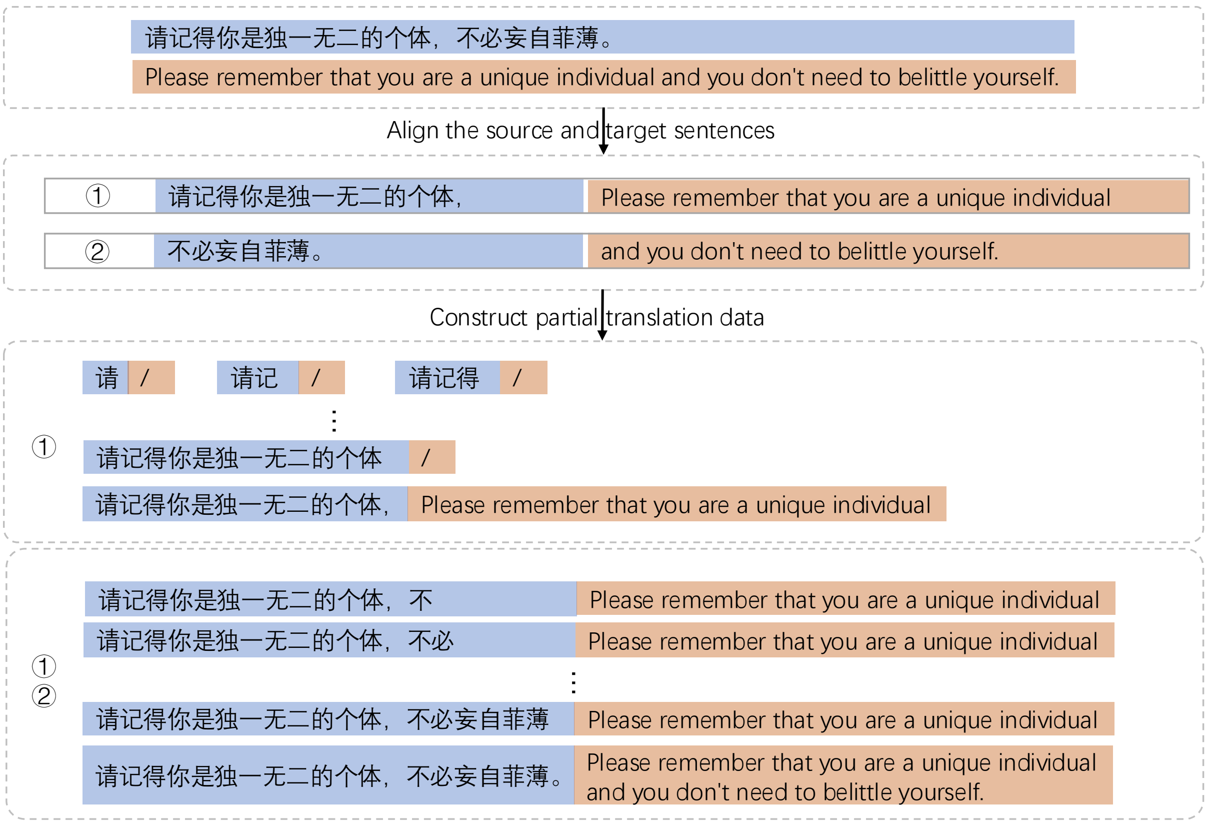
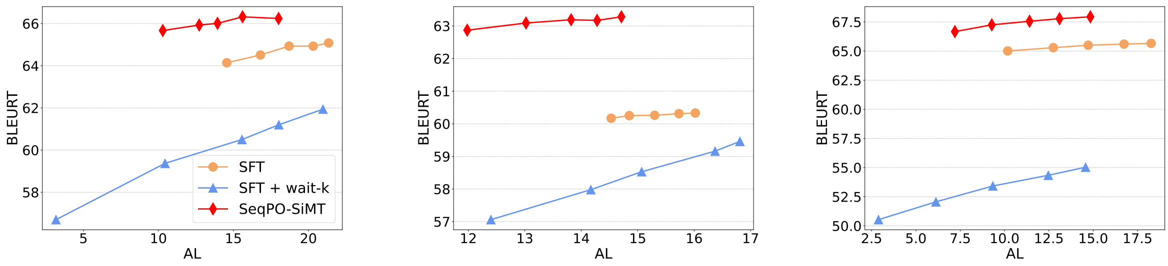


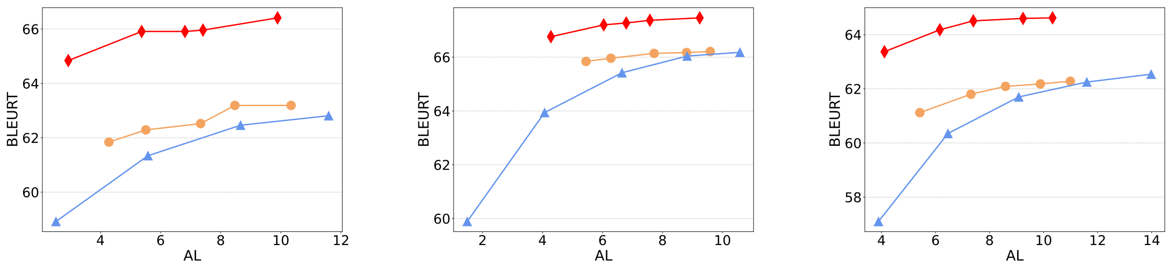
Figure 8: Illustration of SFT data construction process.

Score the following translation from <SOURCE_LANG>to <TARGET_LANG> on a continuous scale from 0 to 100, where score of zero means no meaning preserved and score of one hundred means perfect meaning and grammar .
<SOURCE_LANG> source: <SOURCE_SENTENCE>
<TARGET_LANG> translation: <TARGET_SENTENCE>
Please only give the answer in the form of a number, and do not give any other information.

Figure 9: Prompt template while scoring translation results from different models.

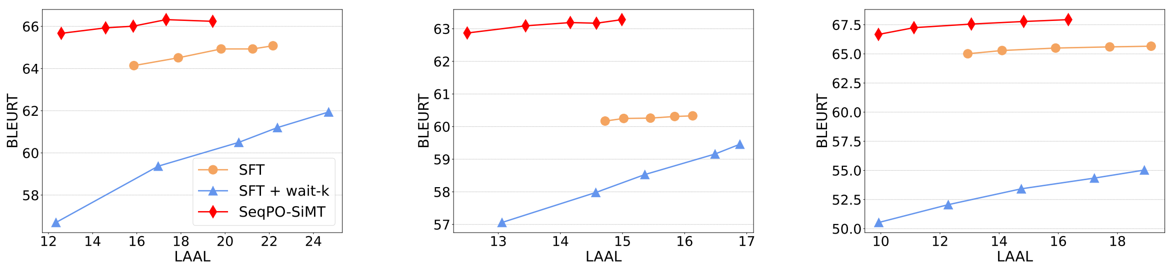


(a) REALSI Zh → En (b) COVOST Zh → En (c) Newstest2021 Zh → En

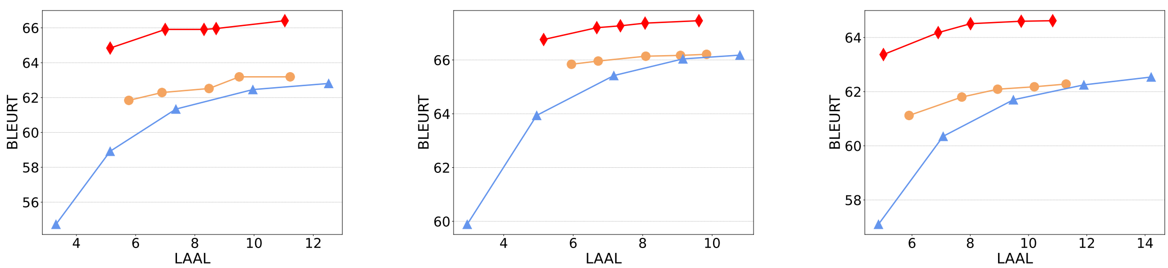


(d) REALSI En → Zh (e) MUSTC En → Zh (f) Newstest2021 En → Zh

Figure 10: BLEURT against AL on Zh → En and En → Zh SiMT tasks.

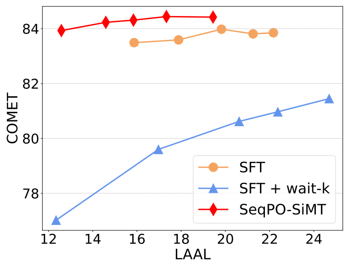


(a) REALSI Zh → En (b) COVOST Zh → En (c) Newstest2021 Zh → En

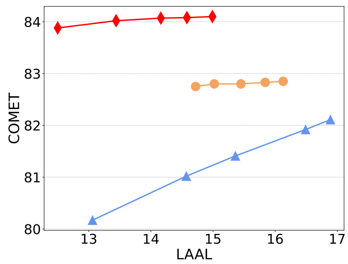


(d) REALSI En → Zh (e) MUSTC En → Zh (f) Newstest2021 En → Zh

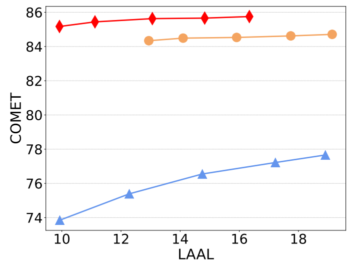
Figure 11: BLEURT against LAAL on Zh → En and En → Zh SiMT tasks.



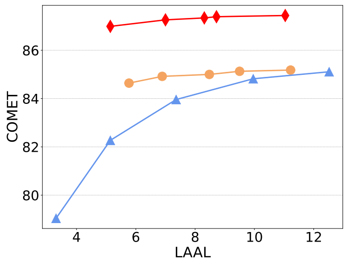
(a) REALSI Zh → En



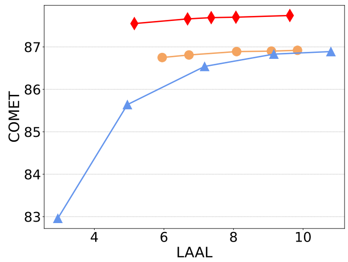
(b) COVOST Zh → En



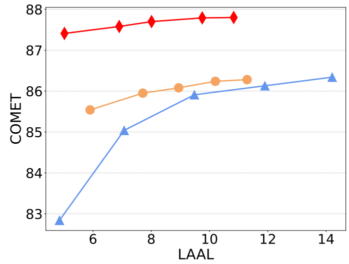
(c) Newstest2021 Zh → En



(d) REALSI En → Zh



(e) MUSTC En → Zh



(f) Newstest2021 En → Zh

Figure 12: COMET against LAAL on Zh → En and En → Zh SiMT tasks.

Language pair	Source texts	SFT output	SeqPO-SiMT output
Zh → En	该百科辞典	/	/
	有电子版和	/	/
	免费的网络	/	This encyclopedia has both a digital version
	版。	there is also a digital version and a free online version.	and a free online version.
En → Zh	And I've been	/	/
	representing these kids	/	/
	who have been	我一直在代表那些	我一直在代表这些孩子,
	sentenced to do	被判处	/
	these very harsh	/	/
	sentences.	非常严厉刑罚的孩子们。他们被判处了这些非常严厉的刑罚。	他们被判处了非常严厉的刑罚。

Table 9: Case Study of SFT and SeqPO-SiMT. A forward slash (/) indicates an empty output.