PsFuture: A Pseudo-Future-based Zero-Shot Adaptive Policy for Simultaneous Machine Translation

Libo Zhao[1,2], **Jing Li**[2,3*], **Ziqian Zeng**[1*]
[1]Shien-Ming Wu School of Intelligent Engineering, South China University of Technology
[2]Department of Computing, Hong Kong Polytechnic University
[3]Research Centre for Data Science & Artificial Intelligence, Hong Kong Polytechnic University
wilbzhao@mail.scut.edu.cn, jing-amelia.li@polyu.edu.hk, zqzeng@scut.edu.cn

Abstract

Simultaneous Machine Translation (SiMT) requires target tokens to be generated in real-time as streaming source tokens are consumed. Traditional approaches to SiMT typically require sophisticated architectures and extensive parameter configurations for training adaptive read/write policies, which in turn demand considerable computational power and memory. We propose PsFuture, the first zero-shot adaptive read/write policy for SiMT, enabling the translation model to independently determine read/write actions without the necessity for additional training. Furthermore, we introduce a novel training strategy, Prefix-to-Full (P2F), specifically tailored to adjust offline translation models for SiMT applications, exploiting the advantages of the bidirectional attention mechanism inherent in offline models. Experiments across multiple benchmarks demonstrate that our zero-shot policy attains performance on par with strong baselines and the P2F method can further enhance performance, achieving an outstanding trade-off between translation quality and latency.[1]

1 Introduction

Simultaneous Machine Translation (SiMT) (Gu et al., 2017) is required to generate target tokens concurrently as it processes incoming source tokens. Differing from traditional machine translation (MT) methods (Bahdanau et al., 2015; Vaswani et al., 2017; Pang et al., 2024) that access the full source text, SiMT necessitates a read/write (R/W) policy to decide between emitting target tokens or awaiting more source input, coupled with the ability to translate from source prefixes to target prefixes (P2P) (Ma et al., 2018). Typically, the read/write policy is integrated with the translation mechanism: either employing a fixed wait-k policy alongside a corresponding translation model (Ma et al., 2018; Elbayad et al., 2020; Zhang et al., 2021b), or utilizing an adaptive policy (Gu et al., 2017; Dalvi et al., 2018; Zheng et al., 2019, 2020; Ma et al., 2020; Zhang and Feng, 2022b; Guo et al., 2023a; Zhao and Zeng, 2024; Chen et al., 2024) that dynamically adjusts read/write decisions based on the context, in conjunction with a model trained to translate policy-defined prefixes. This adaptive method has led to superior performance (Zhang and Feng, 2022a, 2023), yet it demands specialized architectural solutions and multitask learning frameworks for concurrent training of the closely linked adaptive policy and translation model, complicating component optimization and increasing computational demands.

On the other hand, DaP-SiMT (Zhao et al., 2023) introduces a novel framework that separates the adaptive read/write policies from the translation model, offering greater versatility in simultaneous translation. This approach demonstrates that translation models, when directed by an effective adaptive read/write policy, even if initially trained on fixed policies, can balance quality and latency well, achieving state-of-the-art (SOTA) outcomes. However, akin to other adaptive policies, it requires intricate designs and a significant parameter set for training the adaptive read/write policy, often demanding substantial computational resources and memory.

We introduce PsFuture, a zero-shot adaptive read/write policy based on pseudo-future information. This policy utilizes the inherent capabilities of the translation model itself to make read/write decisions without additional training. Similar to the policy in DaP-SiMT (Zhao et al., 2023), we draw inspiration from human simultaneous translation (Al-Khanji et al., 2000; Liu, 2008), where interpreters shift from listening to translating upon anticipating that further future words would not impact

* Corresponding author.
[1]The code is available at https://github.com/lbzhao970/PsFuture

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 1869–1881

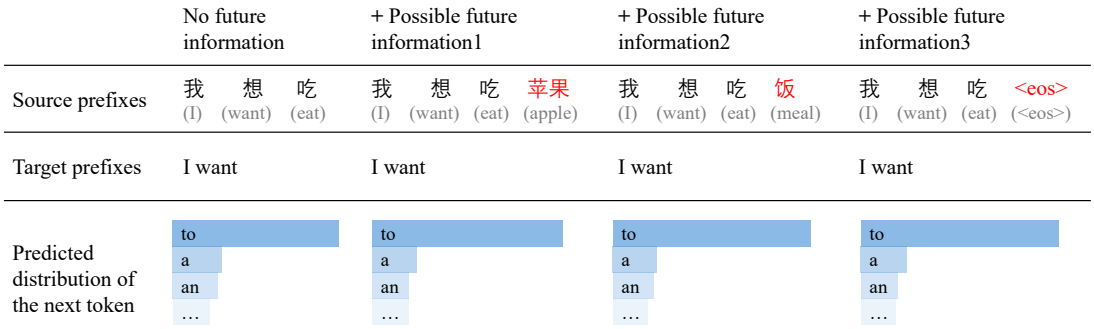


Figure 1: An Zh→En example demonstrating an ideal timing for predicting the next token "to". Even when provided with additional possible future information, the probability distribution of the predicted next token does not change significantly, remaining dominated by the token "to". Therefore, based on the current source prefix "我想吃" and the current target prefix "I want," a write operation can be executed to predict the next token as "to".

their current decisions. As illustrated in Figure 1, this behavior implies a minor divergence between translation predictions based on partial versus more complete source context. However, in simultaneous translation tasks, previewing future source information is not feasible. Our method, PsFuture, overcomes this by utilizing pseudo-future information, which is a token suffix in the source language that can be fixed or dynamically predicted by language models. By quantifying the divergence between the predicted next target token distributions with or without pseudo-future information, and comparing it to a predefined threshold, a read/write decision can be made.

The proposed PsFuture method can be directly applied to most existing simultaneous translation models, such as the multi-path wait-k model, which demonstrates superior performance when directed by effective adaptive policies (Zhao et al., 2023). Additionally, we investigate the application of the PsFuture method to offline translation models. Previous SiMT models (Elbayad et al., 2020; Zhang and Feng, 2022a) conventionally employ a unidirectional attention encoder with tailored masked-cross-attention for prefix-to-prefix training. This approach, while efficient, limits the model's ability to extract features, making it less adept in high-latency scenarios compared to offline models that utilize bidirectional attention mechanisms. To leverage the benefits of bidirectional attention in SiMT, we introduce a novel and effective training technique, Prefix-to-Full (P2F), designed to enhance the performance of offline translation models under diverse latency conditions. Our main contributions can be summarized as follows.

1. We propose the first zero-shot adaptive read/write policy in SiMT, PsFuture, which utilizes the inherent capabilities of the translation model to make read/write decisions without any additional training. To our knowledge, PsFuture is the only adaptive method in the current SiMT field that offers such flexibility.
2. We present an effective training technique, Prefix-to-Full (P2F) to enhance the performance of offline translation models under diverse latency conditions.
3. Experiments across multiple benchmarks demonstrate that our zero-shot policy attains performance on par with strong baselines and achieves an outstanding accuracy-latency balance.

2 Related Work

SiMT policies are broadly categorized into fixed and adaptive schemes. Fixed policies (Ma et al., 2018; Elbayad et al., 2020; Zhang et al., 2021b) execute read/write actions following predefined rules, such as the wait-k policy (Ma et al., 2018), which after reading k source tokens, alternates between reading and writing one token. Conversely, adaptive policies dynamically determine read/write actions based on the evolving source and target context, enhancing the balance between translation accuracy and latency.

Adaptive approaches employ methods like reinforcement learning within a Neural Machine Translation (NMT) framework (Gu et al., 2017), incremental decoding for variable target token output (Dalvi et al., 2018), and attention-based methods (Arivazhagan et al., 2019; Ma et al.,

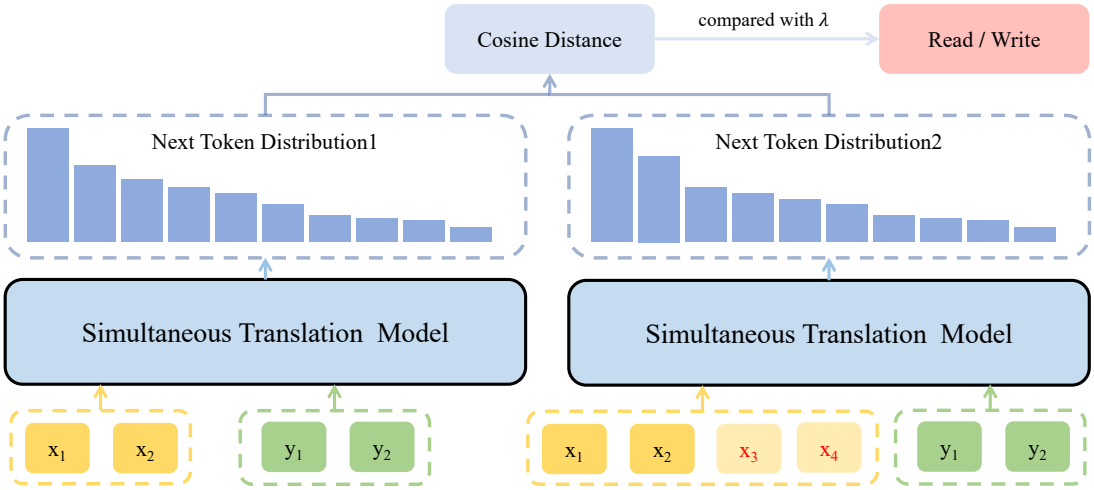


Figure 2: An overall schematic of the PsFuture policy. Based on the current source prefixes (x_1, x_2), target prefixes (y_1, y_2), and pseudo future information (x_3, x_4) (tokens highlighted in red), the simultaneous translation model can directly perform adaptive read/write decisions.

2020). Additionally, the wait-info policy (Zhang et al., 2022) and ITST (Zhang and Feng, 2022a) quantify the waiting latency and information weight respectively for adaptive policy formulation. HMT (Zhang and Feng, 2023) optimizes read/write decisions by enhancing the target sequence's marginal likelihood across various translation initiation points. Kim and Cho (2023) employs a word-level policy to enhance SiMT. Furthermore, Ma et al. (2023) introduces a non-autoregressive streaming Transformer (NAST) to mitigate the challenges of nonmonotonicity and source-information leakage present in conventional autoregressive SiMT frameworks. Guo et al. (2023b) propose to provide a tailored reference for the improvement of SiMT model training.

Sharing a similar inspiration with PsFuture, DaP-SiMT (Zhao et al., 2023) autonomously generates read/write supervisions by leveraging future information divergence for training a decision-making network. In contrast, our approach harnesses the model's inherent translation capability to attain an immediate, zero-shot read/write policy.

3 Preliminary

3.1 Full-sentence MT and SiMT

In full sentence translation tasks, an encoder-decoder architecture like the Transformer (Vaswani et al., 2017) transforms a translation pair $\mathbf{x} = (x_1, x_2, ..., x_N)$ and $\mathbf{y} = (y_1, y_2, ..., y_T)$ by encoding $\mathbf{x}$ into latent representations, followed by the autoregressive generation of target tokens from these representations. Generally, the model is optimized by minimizing the cross-entropy loss.

$$\mathcal{L}_{\text{mt}} = -\sum_{t=1}^{T} \log p\left(y_t \mid \mathbf{x}, \mathbf{y}_{<t}\right) \quad (1)$$

For Simultaneous Machine Translation (SiMT), where $g(t)$ denotes a monotonic non-decreasing function indicating the end timestamp of the source prefix required to produce the t-th target token, the objective function for SiMT can be adapted as follows,

$$\mathcal{L}_{\text{simt}} = -\sum_{t=1}^{T} \log p\left(y_t \mid \mathbf{x}_{\leq g(t)}, \mathbf{y}_{<t}\right). \quad (2)$$

3.2 Wait-k Policy and Multi-Path Wait-k

Wait-k policy (Ma et al., 2018), the most widely used fixed policy, starts by reading k source tokens and then alternates between WRITE and READ action. The function $g(t)$ for the wait-k policy can be formally calculated as,

$$g(t; k) = \min\{t + k - 1, N\}. \quad (3)$$

Multi-path Wait-k (Elbayad et al., 2020) is an efficient technique for wait-k training. It randomly samples different k values between batches during model optimization. By employing a unidirectional attention encoder with a tailored upper triangular masked cross-attention mechanism, the multi-path wait-k model achieves efficient prefix-to-prefix training. Zhao et al. (2023) demonstrates that the multi-path wait-k model can attain SOTA performance under the guidance of effective adaptive policies.

4 Method

4.1 The Pseudo-Future-based Zero-Shot Adaptive Policy

In simultaneous translation, skilled human translators execute read/write decisions grounded in the evolving contexts of source and target texts. Conceptualizing a well-trained translation model as an intelligent agent like a human, our objective is to delineate a zero-shot adaptive read/write policy. This approach enables decision-making based purely on the model's inherent linguistic comprehension and translation proficiency, facilitating adaptive policies without necessitating further training.

Zooming in on the details of the read/write decision-making process, interpreters transition from listening to translating when they have acquired sufficient source context $\mathbf{x}_{\leq g(t)}$ to decide on extending the partial translation $\mathbf{y}_{<t}$ with the next target word y_t. This decision is based on the anticipation that additional source information will not alter their current translation choice, which implies a slight divergence $\mathbf{D}\left(\mathbf{p}_t^{\text{part}}, \mathbf{p}_t^{\text{more}}\right)$ between the interpreters' estimation of the translation distribution with partial source context $\mathbf{p}_t^{\text{part}}$, and the translation distribution considering the more complete source context $\mathbf{p}_t^{\text{more}}$. Interpreters opt to wait for more source words if this divergence becomes substantial.

$$\mathbf{p}_t^{\text{part}} = p(y_t = \cdot|\mathbf{x}_{\leq g(t)}, \mathbf{y}_{<t}) \tag{4}$$

$$\mathbf{p}_t^{\text{more}} = p(y_t = \cdot|\mathbf{x}_{\text{more}}, \mathbf{y}_{<t}), \tag{5}$$

where $\mathbf{x}_{\text{more}}$ represents the more complete source context by appending additional source tokens $(x_{g(t)+1}, x_{g(t)+2}, ...)$ to the current source texts $\mathbf{x}_{\leq g(t)}$ and the distributions can be computed by any SiMT translation models.

However, previewing future source information is not feasible during inferring in simultaneous translation. Our proposed PsFuture method, as the name implies, overcomes this by utilizing pseudo-future information $\mathbf{x}_{\text{ps-suffix}}$, which is a token suffix in the source language. It should be noted that pseudo-future information here does not merely refer to the predicted next few source tokens adhering to human natural language patterns, but rather a broader concept representing additional information beyond current source input. When such information minimally impacts the subsequent target token prediction of the translation agent, it indicates low ambiguity in the translating process, which suggests that the translation of the current source prefix remains incomplete, thereby signaling an appropriate moment for a WRITE operation. Conversely, it indicates an opportune moment for a READ operation. In this work, we explore various forms of pseudo-future information, including both predefined fixed suffixes and adaptive ones that are dynamically predicted by language models (detailed in Section 5.2).

As shown in Equation 6 and 7, we utilize cosine distance, which has been validated as effective in DaP-SiMT (Zhao et al., 2023), to quantify the divergence $\mathbf{D}\left(\mathbf{p}_t^{\text{part}}, \mathbf{p}_t^{\text{pseudo}}\right)$ between the predicted next target token distributions with or without pseudo-future information.

$$\mathbf{D}\left(\mathbf{p}_t^{\text{part}}, \mathbf{p}_t^{\text{pseudo}}\right) = 1 - \cos\left(\mathbf{p}_t^{\text{part}}, \mathbf{p}_t^{\text{pseudo}}\right) \tag{6}$$

$$\mathbf{p}_t^{\text{pseudo}} = p(y_t = \cdot|\mathbf{x}_{\text{pseudo}}, \mathbf{y}_{<t}), \tag{7}$$

where $\mathbf{x}_{\text{pseudo}}$ represents the fake complete source context by appending pseudo future information $\mathbf{x}_{\text{ps-suffix}}$ to the current source texts $\mathbf{x}_{\leq g(t)}$.

By comparing the divergence value to a predefined threshold λ, a read/write decision can be made as Equation 8. The overall schematic of the PsFuture policy is illustrated in Figure 2.

$$\textit{write} \text{ if } \mathbf{D}_{t,g(t)} < \lambda, \text{else } \textit{read} \tag{8}$$

Figure 3 shows an example divergence matrix based on the PsFuture method and a highlighted read/write path, in which we only employ a "<eos>" token as the pseudo-future suffix. It can be observed that comparing with a suitable threshold allows for the easy identification of a potential read/write path.

Following (Zhao et al., 2023), we also introduce another hyperparameter in the read/write decision-making process to limit the maximum number of continuous READ operations for certain languages, thereby enhancing their performance. The inference process is summarized in Algorithm 1.

4.2 The Prefix-to-Full Training Method Adapting Offline Models to SiMT

The PsFuture approach is versatile, compatible with most translation models, including the offline ones[2] (standard Transformer (Vaswani et al.,

[2]In this paper, we distinguish standard Transformer models, which employ the bidirectional attention mechanism, as offline translation models, to differentiate them easily from SiMT models that utilize the unidirectional attention mechanism.

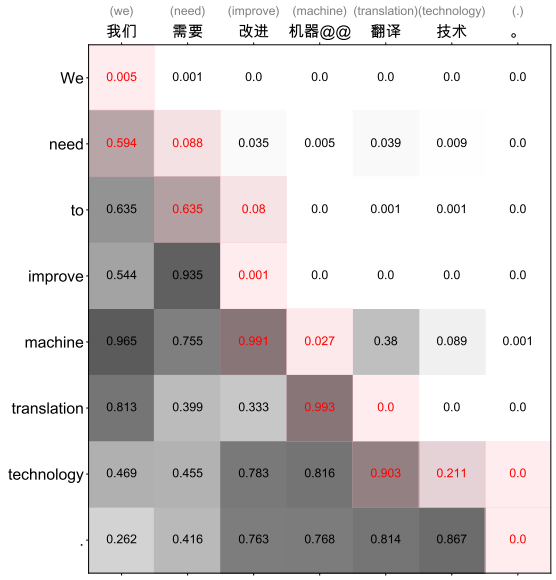


Figure 3: Example of a Zh→En divergence matrix $\mathbf{D}$, where $\mathbf{D}_{t,g(t)} = \mathbf{D}\left(\mathbf{p}_t^{\text{part}}, \mathbf{p}_t^{\text{pseudo}}\right)$. The red elements in the matrix denote a potential read/write path, determined by a predefined threshold λ (0.2 in this case).

2017)). Offline translation models have demonstrated substantial potential for simultaneous translation, as evidenced by their efficacy in speech translation (Papi et al., 2022). However, the lack of Prefix-to-Prefix (P2P) training in offline models leads to lower translation quality under low-latency conditions compared to SiMT models (Ma et al., 2018). On the other hand, the bidirectional attention mechanism of offline models significantly enhances feature extraction, surpassing the unidirectional attention mechanism typically used in SiMT models to facilitate P2P training. Thus, in high-latency scenarios, offline models usually achieve better translation quality, as shown in Table 1.

To harness the benefits of the bidirectional attention mechanism in real-time contexts, we introduces a simple yet effective training strategy for offline translation models named Prefix-to-Full (P2F). This method aims to preserve the model's superior performance in high-latency scenarios while improving its effectiveness in mid-to-low latency situations. The training regimen not only utilizes the conventional translation loss as Equation 1, but also integrates an innovative loss function, Prefix-to-Full (P2F) loss. P2F loss is designed to translate a source prefix into a complete sentence, with the prefix length l being uniformly distributed and randomly chosen. The overall loss is computed as follows.

$$\mathcal{L}_{\text{total}} = (1-\alpha)\mathcal{L}_{\text{mt}} + \alpha\mathcal{L}_{\text{p2f}} \quad (9)$$

$$\mathcal{L}_{\text{p2f}} = -\sum_{t=1}^{T} \log p\left(y_t \mid \mathbf{x}_{\leq l}, \mathbf{y}_{<t}\right) \quad (10)$$

$$l \sim \text{Uniform}(L) \quad (11)$$

$$\alpha \sim \text{Bernoulli}(r), \quad (12)$$

where r is a hyperparamete to control the proportion of the P2F loss. L is the candidate set of the prefix length l, or more specifically, $L = \{1, 2, ..., |\mathbf{x}|\}$.

The P2F loss endows offline translation models with the capability to translate prefixes. Although translating prefixes into full target sentences increases the risk of hallucinations during the simultaneous translation process, we posit that an effective read/write policy can mitigate such occurrences. For a detailed analysis of experiments on this, please refer to Section 6.1.

	Zh→En	De→En
Standard Transformer	20.32	32.99
Multi-path Wait-k	19.45	31.81
ITST	19.15	31.26

Table 1: Comparison of case-insensitive BLEU in offline scenario among the standard Transformer model(Vaswani et al., 2017), multi-path wait-k model(Elbayad et al., 2020) and ITST(Zhang and Feng, 2022a).

5 Experiments

5.1 Datasets

WMT2022 Zh→En[3]. We use a subset with 25M sentence pairs for training[4], from which 1500 unique sentence pairs are extracted as the validation set. We first tokenize the Chinese and English data using the Jieba Chinese Segmentation Tool[5] and Moses[6], respectively, and then apply BPE with 32000 merge operations. We employ the dev set of 956 sentence pairs from BSTC (Zhang et al., 2021a) as the test set.

WMT15 De→En[7]. All 4.5M sentence pairs from this dataset are used for training, and are tokenized using 32K BPE merge operations. We use newstest2013 (3000 sentence pairs) for validation and report results on newstest2015 (2169 sentence pairs).

[3] www.statmt.org/wmt22

[4] The data sources include casia2015, casict2011, casict2015, datum2015, datum2017, neu2017, News Commentary V16, ParaCrawl V9.

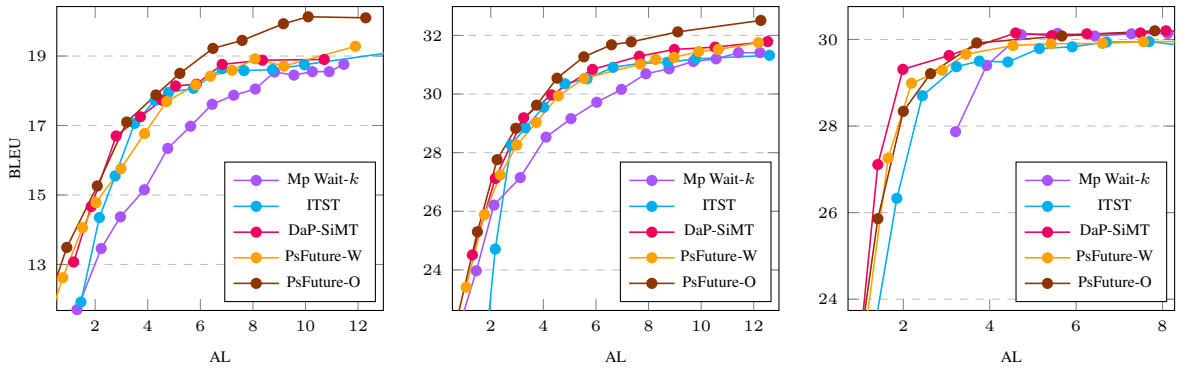
[5] https://github.com/fxsjy/jieba

[6] https://github.com/moses-smt

[7] www.statmt.org/wmt15

	Zh→En	De→En	En→Vi
Fixed suffix1	“<eos>”	“<eos>”	“<eos>”
Fixed suffix2	“<unk> <eos>”	“<unk> <eos>”	“<unk> <eos>”
Fixed suffix3	“... <eos>”	“... <eos>”	“... <eos>”
Fixed suffix4	“...信息到此中断。 <eos>”	“... Die Informationen enden hier. <eos>”	“... Information interrupted here. <eos>”
Random suffix	(Sampled Randomly Each Time A Read/Write Decision Occurs)		
Adaptive suffix	(Dynamically Generated by Language Models)		

Table 2: The pseudo-future suffixes across various language pairs utilized in this paper.



(a) Zh→En, Transformer-Big (b) De→En, Transformer-Base (c) En→Vi, Transformer-Small

Figure 4: Comparison of BLUE vs. AL curves between multi-path (abbreviated as Mp) wait-k, ITST, DaP-SiMT, and our proposed PsFuture approach on three language pairs. PsFuture-W and PsFuture-O denote the multi-path wait-k model based PsFuture method and the offline model (P2F-enhanced) based PsFuture method, respectively.

IWSLT15 En→Vi[8]. All 133K sentence pairs from this dataset (Luong and Manning, 2015) are used for training. We use TED tst2012 (1553 sentence pairs) for validation and TED tst2013 (1268 sentence pairs) as the test set. Following the settings in (Ma et al., 2020), we adopt word-level tokenization and replace rare tokens (frequency < 5) with <unk>. The vocabulary sizes are 17K for English and 7.7K for Vietnamese, respectively.

5.2 Settings

The Pseudo-Future Suffix . In this study, we investigate various pseudo-future suffixes, denoted as $\mathbf{x}_{\text{ps-suffix}}$, as detailed in Table 2. These suffixes can be divided into two categories: fixed and adaptive. A primary criterion for selecting a fixed suffix is its richness in information. For instance, the “<eos>” token, often encountered in training translation models, effectively indicates sentence termination. Consequently, all chosen suffixes conclude with “<eos>” to guarantee an essential increment of information.

Specifically, the fixed suffixes range from the basic “<eos>” token (suffix 1) to more complex structures involving special tokens (“<unk> <eos>”, suffix 2) and natural sentence extensions (suffixes 3 and 4), which simulate ellipsis and ellipsis with signals of information discontinuity. We also conduct an experiment with random suffixes to investigate the sensitivity of the PsFuture method to suffix content. These random suffixes consist of four tokens, each randomly selected from the top 200 most frequent tokens in the vocabulary, ensuring adequate information. Furthermore, the suffix is resampled randomly each time a read/write decision occurs.

The adaptive suffix is dynamically generated by large language models, based on the current source prefix for pseudo-future information prediction. For Zh→En, we employ the Chinese-Llama-2-7b model[9], while the Llama-2-7b-chat (Touvron et al., 2023) is used for De→En and En→Vi experiments. In the main results (5.3), we empirically determine the optimal suffix through performance evaluation. Section 6.1 delves into the effects of various suffixes on the experimental outcomes, providing a thorough assessment.

[8] nlp.stanford.edu/projects/nmt

[9] https://github.com/LinkSoul-AI/Chinese-Llama-2-7b

The Prefix-to-Full Loss Ratio. The hyperparameter P2F ratio r is employed to control the proportion of the P2F loss. The most effective configurations are identified as 0.5, 0.8, and 0.5 for Zh→En, De→En, and En→Vi, respectively. Detailed information on the ablation studies concerning hyperparameter r is referred to Section 6.1.

Other Settings. The proposed PsFuture policy undergoes empirical experiments based on the multi-path wait-k model and the P2F-enhanced offline model as mentioned in Section 4.2, comparing its performance with two leading models in the SiMT domain, ITST (Zhang and Feng, 2022a) and DaP-SiMT (Zhao et al., 2023). All our implementations are based on the Transformer (Vaswani et al., 2017) architecture and adapted from the Fairseq Library (Ott et al., 2019). For the Zh→En experiments, we utilize the transformer big architecture, while the base and small architectures are used for De→En and En→Vi experiments respectively.

For evaluation, following ITST and DaP-SiMT, we report case-insensitive BLEU (Papineni et al., 2002) scores to assess translation quality and Average Lagging (AL/token) (Ma et al., 2018) to measure latency. Regarding the maximum number of continuous read actions in our method, we empirically select the best-performing configurations, which are no constraint, 4, no constraint for Zh→En, De→En, En→Vi respectively. Furthermore, to achieve more robust inference results, the initial length of the source prefix during the real-time translation process is set to 2.

5.3 Main Results

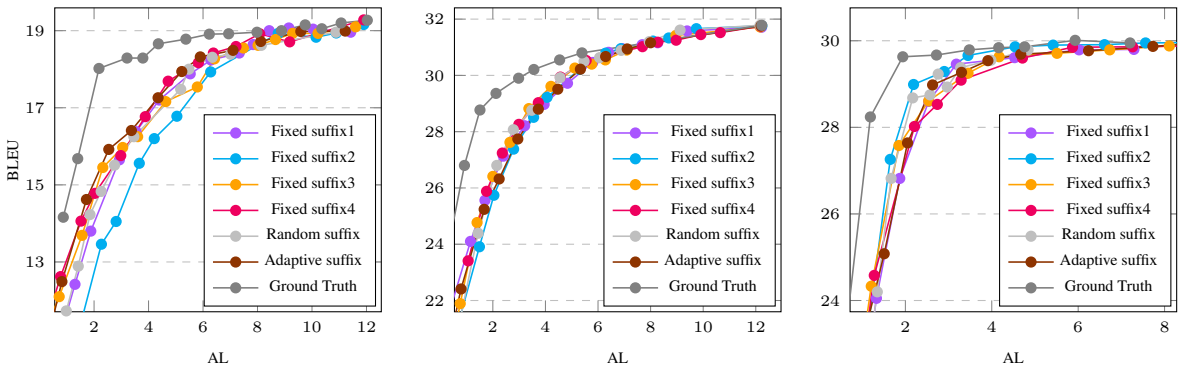
We compare the proposed PsFuture method against previous approaches for three language pairs in Figure 4. PsFuture-W and PsFuture-O refer to the multi-path wait-k model-based PsFuture approach and the offline model (P2F enhanced) based PsFuture method, respectively.

Firstly, the PsFuture-W experiment significantly surpasses traditional multi-path wait-k models, benefiting from the proposed PsFuture policy over the fixed wait-k policy. Notably, the performance of PsFuture-W often matches or exceeds the SiMT leading model ITST, which is specifically trained with a complicated adaptive read/write policy. This highlights the capability of SiMT translation models to make adaptive decisions themselves. Although trained with a fixed strategy, the multi-path wait-k model, when coupled with the zero-shot PsFuture policy, significantly outperforms its counterparts and rivals strong SiMT baselines.

Secondly, the performance of PsFuture-O demonstrates improvements over PsFuture-W to varying extents across all language pairs, especially in the Zh→En experiment where it outdoes the former SiMT SOTA method, DaP-SiMT. As anticipated, the offline translation model, endowed with superior feature extraction capabilities, achieves better performance at moderate to high latencies, while the introduction of the Prefix-to-Full loss ensures the model maintains comparable effectiveness at lower latencies.

6 Analysis

In this scetion, we aim to provide a detailed examination of the proposed method. Unless otherwise noted, the PsFuture-related experiments are based on the multi-path wait-k model, and the results stem from the Zh→En Transformer-Big model.



(a) Zh→En, Transformer-Big (b) De→En, Transformer-Base (c) En→Vi, Transformer-Samll

Figure 5: Effect of the pseudo-future suffix

6.1 Effect of the pseudo-future suffix

This part investigates the influence of various pseudo-future suffixes (Table 2) on the experiment results. As shown in Figure 5, the majority of suffixes tested can achieve a desirable equilibrium between translation quality and latency, which showcases the tolerance of the proposed method to the choice of suffixes. Through the comparison of various experimental results, it is also feasible to identify specific suffixes for particular language pairs to optimize performance. Adaptive suffixes, generated by large language models, consistently perform well across various corpora. However, due to a lack of extensive experimentation with different adaptive suffixes, their effectiveness does not surpass that of the best fixed suffixes. We believe that a large-scale exploration of adaptive suffix experiments could potentially yield superior outcomes.

Additionally, it is surprising that the random suffix experiment exhibits unexpectedly strong performance. Although there are fluctuations in specific areas, the overall result is comparable to that of other meticulously crafted suffixes. This indicates that PsFuture's effectiveness is not significantly affected by suffix content. This finding indicates that the proposed method possesses a substantial lower bound, emphasizing its robustness and straightforward applicability. These qualities align with the method's key features: simple yet effective.

Furthermore, experiments with ground truth suffixes are conducted to ascertain the upper bound of the PsFuture method. The results indicate that there remains potential for enhancement. Future efforts will focus on incrementally approaching this upper limit by exploring and refining suffixes.

6.2 Effect of the P2F loss

Figure 6 illustrates the impact of different Prefix-to-Full (P2F) loss ratios on the performance of our experiments. Setting the P2F ratio r to 0 corresponds to conventional offline translation model training. This configuration, when applied directly to SiMT tasks, yields less than ideal results, especially at lower to medium latencies. Incorporating any level of P2F loss markedly improves performance, effectively tailoring the offline model for SiMT applications. Moreover, the experimental results reveal a noticeable sensitivity to the P2F ratio r, indicating that an optimal r can enhance the balance between translation accuracy and latency.

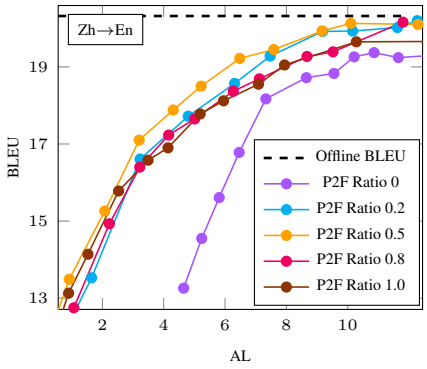


Figure 6: BLEU vs. AL curves comparing among PsFuture-O experiments with varying P2F ratios.

6.3 Concerns on Hallucination

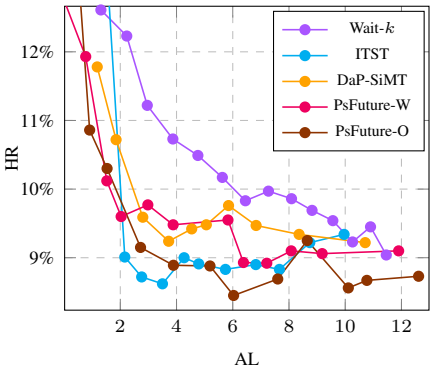


Figure 7: Hallucination Rate (HR) vs. Average Lagging (AL) curves comparing PsFuture-O with other methods.

In the PsFuture-O experiment, the additional introduction of the Prefix-to-Full (P2F) loss aims to enhance the model's capability to translate a source prefix into a full target sentence, thereby adapting it for SiMT tasks. However, this approach may increase the risk of hallucinations during the translation process. A hallucination is defined as a generated token that cannot be aligned with any source word. To illustrate this potential issue, we compare the hallucination rate (Chen et al., 2021) of hypotheses generated by PsFuture-O with those produced by other methods. The comparative results are depicted in Figure 7.

It is evident that, overall, the PsFuture-O experiment achieves the lowest hallucination rate, surpassing not only the DaP-SiMT and PsFuture-W methods, which rely on the multi-path wait-k model, but also outperforming the meticulously trained ITST model. This indicates that the proposed PsFuture policy effectively mitigates the occurrence of hallucinations during the simultaneous translation inference process.

7 Conclusion

In this paper, we propose the first zero-shot adaptive read/write policy for SiMT, PsFuture. It empowers the translation model to autonomously decide on read/write actions without requiring additional training and can attain effectiveness on par with previously meticulously trained adaptive policies. Moreover, we introduce a novel training strategy, Prefix-to-Full (P2F), specifically tailored to adjust offline translation models for SiMT applications, exploiting the benefits of the bidirectional attention mechanism inherent in offline models.

Limitations

In this work, the proposed PsFuture policy conducts two forward computations for each read/write decision-making, which may increase the total computational load when inferring. However, it's important to note that while other adaptive policy methods may require only one forward computation for each decision, they also necessitate additional computations, which are also not negligible when compared to single forward computing. Overall, despite the increased computational requirement for inference, the PsFuture method eliminates the need for additional learnable parameters and training to obtain a read/write decision maker, which also significantly reduces computational demands during training.

Ethics Statement

After careful review, to the best of our knowledge, we have not violated the ACL Ethics Policy.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62406114), the Guangzhou Basic and Applied Basic Research Foundation (No. 2023A04J1687), the Fundamental Research Funds for the Central Universities (No. 2024ZYGXZR074), the NSFC Young Scientists Fund (No. 62006203), the Research Grants Council of the Hong Kong Special Administrative Region (No. PolyU/25200821), the Innovation and Technology Fund (No. PRP/047/22FX), PolyU Research Centre on Data Science and Artificial Intelligence (No. 1-CE1E) and a gift fund from Microware (No. N-ZDG2).

References

Raja Al-Khanji, Said El-Shiyab, and Riyadh Hussein. 2000. On the use of compensatory strategies in simultaneous interpretation. *Journal des Traducteurs 45(3):548–577*.

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. *arXiv preprint arXiv:1906.05218*.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Junkun Chen, Renjie Zheng, Atsuhito Kita, Mingbo Ma, and Liang Huang. 2021. Improving simultaneous translation by incorporating pseudo-references with fewer reorderings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5857–5864, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinjie Chen, Kai Fan, Wei Luo, Linlin Zhang, Libo Zhao, Xinggao Liu, and Zhongqiang Huang. 2024. Divergence-guided simultaneous speech translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17799–17807.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient wait-k models for simultaneous machine translation. *arXiv preprint arXiv:2005.08595*.

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. 2023a. Learning optimal policy for simultaneous machine translation via binary search. *arXiv preprint arXiv:2305.12774*.

Shoutao Guo, Shaolei Zhang, and Yang Feng. 2023b. Simultaneous machine translation with tailored reference. *arXiv preprint arXiv:2310.13588*.

Kang Kim and Hankyu Cho. 2023. Enhanced simultaneous machine translation with word-level policies. *arXiv preprint arXiv:2310.16417*.

Minhua Liu. 2008. How do experts interpret. *Implications from research in interpreting studies and cognitive*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, et al. 2018. Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. *arXiv preprint arXiv:1810.08398*.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020. Monotonic multihead attention. In *International Conference on Learning Representations*.

Zhengrui Ma, Shaolei Zhang, Shoutao Guo, Chenze Shao, Min Zhang, and Yang Feng. 2023. Non-autoregressive streaming transformer for simultaneous translation. *arXiv preprint arXiv:2310.14883*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianhui Pang, Baosong Yang*, Derek Fai Wong*, Yu Wan, Dayiheng Liu, Lidia Sam Chao, and Jun Xie. 2024. Rethinking the exploitation of monolingual data for low-resource neural machine translation. *Computational Linguistics*, 50(1):25–47.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Does simultaneous speech translation need simultaneous models? *arXiv preprint arXiv:2204.03783*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. ACL '02, page 311–318, USA. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ruiqing Zhang, Xiyang Wang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Zhi Li, Haifeng Wang, Ying Chen, and Qinfei Li. 2021a. Bstc: A large-scale chinese-english speech translation dataset. *arXiv preprint arXiv:2104.03575*.

Shaolei Zhang and Yang Feng. 2022a. Information-transport-based policy for simultaneous translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Online and Abu Dhabi. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022b. Reducing position bias in simultaneous machine translation with length-aware framework. *arXiv preprint arXiv:2203.09053*.

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

Shaolei Zhang, Yang Feng, and Liangyou Li. 2021b. Future-Guided Incremental Transformer for Simultaneous Translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14428–14436.

Shaolei Zhang, Shoutao Guo, and Yang Feng. 2022. Wait-info Policy: Balancing Source and Target at Information Level for Simultaneous Machine Translation.

Libo Zhao, Kai Fan, Wei Luo, Wu Jing, Shushu Wang, Ziqian Zeng, and Zhongqiang Huang. 2023. Adaptive policy with wait-k model for simultaneous translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4832, Singapore. Association for Computational Linguistics.

Libo Zhao and Ziqian Zeng. 2024. Dap-simt: divergence-based adaptive policy for simultaneous machine translation. *International Journal of Machine Learning and Cybernetics*, pages 1–20.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. *arXiv preprint arXiv:2004.13169*.

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 Effect of The Max Continuous READ Constraint

Following DaP-SiMT (Zhao et al., 2023), we set a constraint on the maximum consecutive reads allowed during inference, necessitating a write action once this limit is reached. Figure 8 demonstrates the influence of this hyperparameter on various language pairings. Consistent with DaP-SiMT, we note that this parameter exerts little or even negative impact on the Zh→En and En→Vi experiments, yet it proves substantially advantageous for the De→En pair. Thus, it is advisable to identify the optimal maximum number of continuous reads on the validation set before the practical implementation of this approach.

B Case Study

Here, we present specific cases to demonstrate the effectiveness of the proposed method, as illustrated in Figure 9 and Figure 10. It is evident that the PsFuture policy can effectively align the source and target tokens. Even in instances where there is a significant difference in word order between source and target, the PsFuture method can still make correct decisions, waiting for more source information to proceed with the accurate translation.

C Discussion on The Extra Cost Caused by Bi-directional Encoders

During the decoding process, the use of a unidirectional encoder allows for incremental decoding, which reduces computational requirements. However, this is not feasible with bidirectional encoders. Compared to unidirectional encoders, predicting each target token necessitates the additional computation of $g(t) - 1$ encoder hidden states ($g(t)$ represents the current number of source tokens). While the extra computational load is affordable for shorter texts, it becomes considerably burdensome for longer texts, potentially imposing untenable cost. If users cannot accommodate the substantial computational demand, they can opt for a unidirectional encoder with the PsFuture method, akin to the PsFuture-W experiment mentioned in this paper which also demonstrates performance comparable to previous top non-zero-shot read/write policies.

D Algorithm

The inference process of PsFuture policy is summarized in Algorithm 1.

E Numerical Results

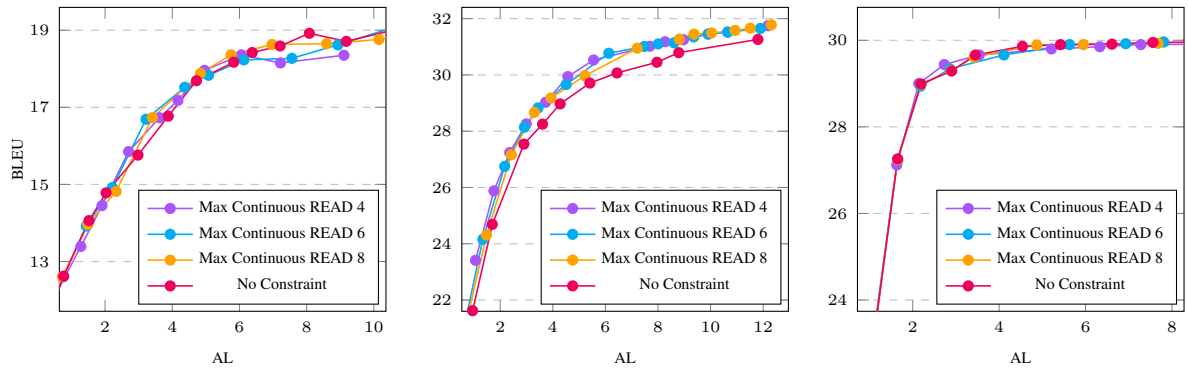
The numerical main results are presented in Table 3.

Algorithm 1: SiMT inference with PsFuture policy

Input: streaming source tokens: $\mathbf{X}_{\leq j}$,
threshold: δ,
target idx: $i \leftarrow 1$,
source idx: $j \leftarrow 2$,
max continuous READ constraint: r_{max},
current number of continuous READ: $r_c \leftarrow 1$

Output: target tokens: $\mathbf{Y} \leftarrow$ {<BOS>}

while $\mathbf{Y}_{i-1} \neq$ <EOS> **do**
  calculate R/W confidence (cosine distance) c with $\mathbf{Y}_{i-1}$ using the PsFuture policy mentioned in 4.1;
  if $c \leq \delta$ **or** $r_c \geq r_{max}$ **then**
    translate y_i with $\mathbf{X}_{\leq j}$,$\mathbf{Y}_{\leq i-1}$;
    if $y_i \neq$ <EOS> **or** $j \geq |\mathbf{X}|$ **then**
      // execute WRITE action
      $\mathbf{Y}$.Append(y_i);
      $r_c \leftarrow 0$;
      $i \leftarrow i + 1$;
    else
      // execute READ action
      $j \leftarrow j + 1$;
      $r_c \leftarrow r_c + 1$;
  else
    // execute READ action
    $j \leftarrow j + 1$;
    $r_c \leftarrow r_c + 1$;
return $\mathbf{Y}$



(a) Zh→En, Transformer-Big (b) De→En, Transformer-Base (c) En→Vi, Transformer-Small

Figure 8: Effect of the constraint on the maximum number of continuous read

Source	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个	正确	的	行动	。		
	(so)	(first)	(you)	(need)		(this)	(thing)	(have a)	(correct)	(understanding)				(have)	(this)	(correct)		(action)	(.)		
Step	**Streaming Inputs**																				**Outputs**
1	那么	首先																			So
2	那么	首先	你																		first
3	那么	首先	你																		you
4	那么	首先	你	需要																	need
5	那么	首先	你	需要	对																to
6	那么	首先	你	需要	对	这个	东西	有个													have
7	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识											a
8	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识											correct
9	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识											understanding
10	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识											of
11	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识											this
12	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.										thing
13	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才									before
14	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有							you
15	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有							can
16	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个						have
17	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个						this
18	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个	正确					correct
19	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个	正确	的	行动			action
20	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个	正确	的	行动	。		.
21	那么	首先	你	需要	对	这个	东西	有个	正确@@	认识	.	才	会	有	这个	正确	的	行动	。	<eos>	<eos>

Figure 9: Case No.226 in BSTC Zh→En test set, evaluated with $\lambda = 0.08$.

Source	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话	动作	。		
	(with)		(dialogue)	(state)		(we)		(can)	(go)	(based on)	(some)	(bool)	(rule)	(go)	(trigger)	(this)	(conversation)	(action)			
Step	**Streaming Inputs**																				**Outputs**
1	有	了																			With
2	有	了	对话	状态																	the
3	有	了	对话	状态																	dialogue
4	有	了	对话	状态	呢	我们															state
5	有	了	对话	状态	呢	我们															we
6	有	了	对话	状态	呢	我们	就	可以													can
7	有	了	对话	状态	呢	我们	就	可以	去	基于											go
8	有	了	对话	状态	呢	我们	就	可以	去	基于											to
9	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个					trigger
10	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个					this
11	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				conversation
12	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				based
13	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				on
14	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				some
15	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				Bo@@
16	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				ol@@
17	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				ean
18	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				rules
19	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话				.
20	有	了	对话	状态	呢	我们	就	可以	去	基于	一些	布尔	规则	去	触发	这个	对话	动作	。	<eos>	<eos>

Figure 10: Case No.85 in BSTC Zh→En test set, evaluated with $\lambda = 0.08$. While there is a significant difference in word order between source and target, the PsFuture method can still make correct decisions. Specifically, at step 9, the PsFuture Policy reads an additional six tokens in sequence to ensure the accuracy of the translation.

	Main Results (Figure 4)									
	Mp Wait-k		ITST		DaP-SiMT		PsFuture-W		PsFuture-O	
	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU
Zh→En	1.31	11.7	0.7	8.91	1.18	13.07	0.06	10.99	0.31	12.12
	2.23	13.46	1.46	11.92	1.85	14.67	0.77	12.62	0.92	13.49
	2.96	14.37	2.16	14.35	2.8	16.7	1.52	14.06	2.08	15.26
	3.87	15.15	2.76	15.55	3.72	17.25	2.03	14.78	3.2	17.1
	4.76	16.34	3.5	17.06	4.54	17.73	2.98	15.76	4.31	17.88
	5.63	16.98	4.27	17.72	5.06	18.14	3.88	16.77	5.22	18.5
	6.45	17.61	4.79	17.95	5.85	18.19	4.72	17.69	6.48	19.22
	7.27	17.87	5.74	18.07	6.83	18.76	5.83	18.17	7.59	19.45
	8.09	18.05	6.82	18.63	8.36	18.88	6.38	18.42	9.16	19.93
	8.82	18.54	7.66	18.58	10.71	18.9	7.21	18.59	10.1	20.13
	9.56	18.45	8.74	18.61			8.08	18.92	12.29	20.1
	10.26	18.55	9.96	18.75			9.18	18.71		
	10.9	18.55	13.68	19.15			11.9	19.28		
	11.46	18.76								
	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU
De→En	0.47	21.08	1.57	19.2	0.49	21.65	1.06	23.41	1.49	25.3
	1.45	23.97	2.17	24.71	1.3	24.51	1.76	25.88	2.24	27.76
	2.12	26.21	2.77	28.26	2.17	27.12	2.36	27.24	2.95	28.83
	3.12	27.15	3.31	28.85	3.25	29.19	2.99	28.26	3.74	29.62
	4.1	28.53	4.01	29.55	4.31	29.97	3.73	29.03	4.52	30.54
	5.05	29.16	4.82	30.35	5.87	30.84	4.57	29.94	5.54	31.27
	6.03	29.72	5.66	30.52	7.65	31.29	5.55	30.53	6.59	31.68
	6.97	30.16	6.65	30.91	8.98	31.52	7.69	31.02	7.34	31.78
	7.9	30.69	7.7	31.05	10.53	31.6	8.27	31.18	9.11	32.12
	8.78	30.86	8.73	31.08	12.53	31.79	8.97	31.25	12.27	32.51
	9.7	31.11	9.79	31.2			9.91	31.45		
	10.57	31.2	12.6	31.32			10.65	31.52		
	11.42	31.41					12.18	31.75		
	12.24	31.41								
	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU	AL	BLEU
En→Vi	3.21	27.87	1.29	23.06	0.89	21.89	0.84	21.16	0.21	18.08
	3.93	29.4	1.85	26.33	1.41	27.11	1.65	27.26	0.86	21.96
	4.73	30.11	2.44	28.7	1.99	29.31	2.19	28.99	1.41	25.86
	5.57	30.14	3.23	29.37	3.06	29.63	2.9	29.29	2	28.34
	6.43	30.08	3.76	29.5	4.6	30.15	3.45	29.66	2.63	29.21
	7.28	30.13	4.42	29.48	5.44	30.09	4.54	29.86	3.7	29.92
	8.12	30.14	5.15	29.79	6.25	30.13	5.42	29.9	5.67	30.08
	8.93	30.11	5.91	29.83	7.49	30.15	6.61	29.91	7.82	30.2
	9.7	30.1	6.7	29.94	8.08	30.2	7.56	29.95	9.91	30.14
	10.43	30.2	7.69	29.95	8.74	30.17	9.1	29.96		
	11.13	30.16	8.67	29.84	9.61	30.01				
	11.79	30.13	9.93	29.95	10.67	30.11				
	12.41	30.16	12.58	30.01	11.69	30.1				
	13.01	30.18								

Table 3: Numerical results in Figure 4.