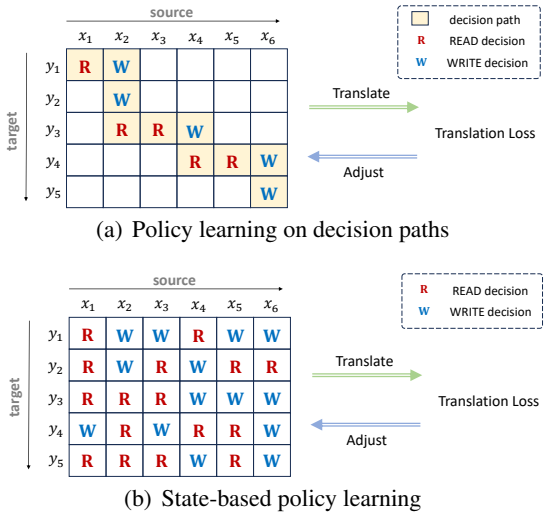
State-based Policy Learning for Simultaneous Machine Translation

Anonymous ACL submission

Abstract

Simultaneous **M**achine **T**ranslation (SiMT) generates target outputs while receiving stream source inputs, and needs a read/write policy to decide whether to wait for a new source token or generate a new target token. These decisions form a *decision path*, and traditional SiMT methods learn policies by exploring various decision paths. However, it is challenging to find the optimal policy for the number of decision paths grows exponentially. To solve this issue, we propose **S**tate-based **P**olicy **L**earning (SPL) which utilizes the *decision independence* in the SiMT process. Specifically, each decision is made only based on current state and is not affected by any other states. Therefore, SPL can learn an effective policy by independently adjusting decisions for each state. Experiment results show that SPL outperforms existing models by learning a better policy and enables offline machine translation models to perform the SiMT task through fine-tuning.



(a) Policy learning on decision paths

(b) State-based policy learning

Figure 1: Illustration of policy learning. For policy learning on decision paths, the policy is encouraged to construct a reasonable sequence of R/W to make the SiMT model achieve high-quality translation in low latency. For state-based policy learning, the policy is directly adjusted to make suitable decisions at all states.

1 Introduction

Simultaneous **M**achine **T**ranslation (SiMT) (Gu et al., 2017; Ma et al., 2019; Zhang et al., 2020) generates translation while receiving the streaming source sentences. Different from normal offline machine translation (OMT) (Vaswani et al., 2017), SiMT needs a suitable read/write policy to decide whether to wait for the coming source inputs (READ) or generate target tokens (WRITE).

Current policies include fixed policy and adaptive policy. Fixed policies (Ma et al., 2019; Zhang and Feng, 2021) make decisions based on predefined rules, which cannot balance well between translation quality and latency. In contrast, adaptive policies (Zheng et al., 2019; Miao et al., 2021; Zhang and Feng, 2023) will be dynamically adjusted during training to achieve better performance. Taking Figure 1(a) as an example, a series of READ and WRITE decisions made by the policy form a *decision path*, along which the SiMT model can output a complete translation. Then, the policy is adjusted based on the SiMT performance along this decision path. By exploring various decision paths like this, the SiMT model can continuously refine its policy (Zhang and Feng, 2022b).

However, learning policy on decision paths has inherent limitations. Since the number of potential decision paths grows exponentially, it is challenging to explore all of them. Therefore, some paths are usually restricted or excluded for stable training (Zheng et al., 2019; Miao et al., 2021; Zhang and Feng, 2023), but this causes insufficient exploration and sub-optimal policies. Additionally, building complete decision paths requires model structure compromises such as unidirectional encoders (Elbayad et al., 2020; Ma et al., 2020), which negatively affect the translation quality and compatibility with existing OMT models (Iranzo-Sánchez

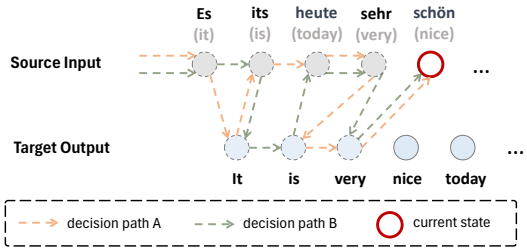


Figure 2: An example of decision independence for each state in De→En SiMT. The decision path A is "RWRRRWWR" and the decision path B is "RRWWRRWR". At current state, the received source tokens are "Es its heute sehr schön", and the generated target tokens are "It is very". Therefore, it is enough to generate "nice" right now. This conclusion is not affected by whether the state is on path A or B.

et al., 2022; Kim and Cho, 2023).

In the policy learning process, it is unnecessary to construct complete decision paths due to the *decision independence* of each state. As shown in Figure 2, the decision at each step is only based on current context. If the received source tokens at the current decision state provide enough information for generating new target tokens, the SiMT model should opt for WRITE; otherwise, it should choose READ. Therefore, it is feasible to learn a reasonable policy by independently optimizing decisions at each state, rather than across the entire path.

Under these grounds, we propose **S**tate-based **P**olicy **L**earning (SPL). Without constructing complete decision paths, SPL directly explores all possible decision states in the SiMT process, as shown in Figure 1(b). Specifically, SPL utilizes unsupervised confidence estimation (DeVries and Taylor, 2018; Lu et al., 2022) to assess the appropriate decision for each state. To traverse all decision states efficiently and systematically, SPL divides them into groups according to their received prefixes, and one group is sampled for confidence estimation in each iteration. By directly exploring all decision states, SPL can adjust the policy with polynomial-level complexity. Besides, it is easy to explore all possible decision states comprehensively without constraints, which facilitates the discovery of the optimal policy. Furthermore, with no need to construct complete decision paths, SPL can employ model structures such as bidirectional encoders, thus ensuring compatibility with existing pre-trained OMT models.

Our contributions can be summarized in the following:

- We propose **S**tate-based **P**olicy **L**earning (SPL). It is a novel paradigm for policy learning that directly explores the decision states without constructing complete decision paths.
- Our method can perform policy learning more efficiently and systematically traverse all potential decision states to learn a better policy. Besides, our method also ensures compatibility with existing pre-trained OMT models.
- Experiments show that SPL can outperform strong baselines on Zh→En, De→En and En→Ro SiMT tasks. Besides, SPL can utilize pre-trained OMT models to obtain a superior SiMT performance with fine-tuning.

2 Background

Simultaneous Machine Translation For the SiMT task, we denote the source sentence as $\mathbf{x} = (x_1, ..., x_M)$ and the corresponding target sentence as $\mathbf{y} = (y_1, .., y_N)$. Since the source inputs are streaming, we denote the number of source tokens available when generating y_i as g_i, and hence the prediction probability of y_i is $p(y_i \mid \mathbf{x}_{\leq g_i}, \mathbf{y}_{\leq i-1})$ (Ma et al., 2019). Therefore, the decoding probability of $\mathbf{y}$ is calculated as:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{N} p(y_i \mid \mathbf{x}_{\leq g_i}, \mathbf{y}_{<i}) \tag{1}$$

Decision state and decision path Denote the decision state $s_{i,j}$ as the given context $(\mathbf{x}_{\leq j}, \mathbf{y}_{<i})$ when decide whether to generate y_i or to wait for x_{j+1}. During the entire SiMT process, a series of decisions are made by a read/write policy $\mathrm{P}(\cdot)$, guiding the translation model to complete the translation in SiMT way. These decisions form a decision path from $s_{0,0}$ to $s_{M,N}$, as shown in Figure 1(a).

Decision Independence of states Each decision in the SiMT process is made based on the given context (Zhao et al., 2023). Specifically, at $s_{i,j}$, the decision $d_{i,j} \in \{\mathrm{WRITE}, \mathrm{READ}\}$ can be made as follows:

$$d_{i,j} = \mathrm{P}(s_{i,j}) = \mathrm{P}(x_{\leq j}, y_{<i}), \tag{2}$$

This formula shows $d_{i,j}$ is only determined by $s_{i,j}$ and not affected by any other states. Therefore, every decision state can be explored and optimized independently without complete decision paths. If $P(\cdot)$ can output reasonable decisions at any states, it is undoubtedly a suitable policy for SiMT.

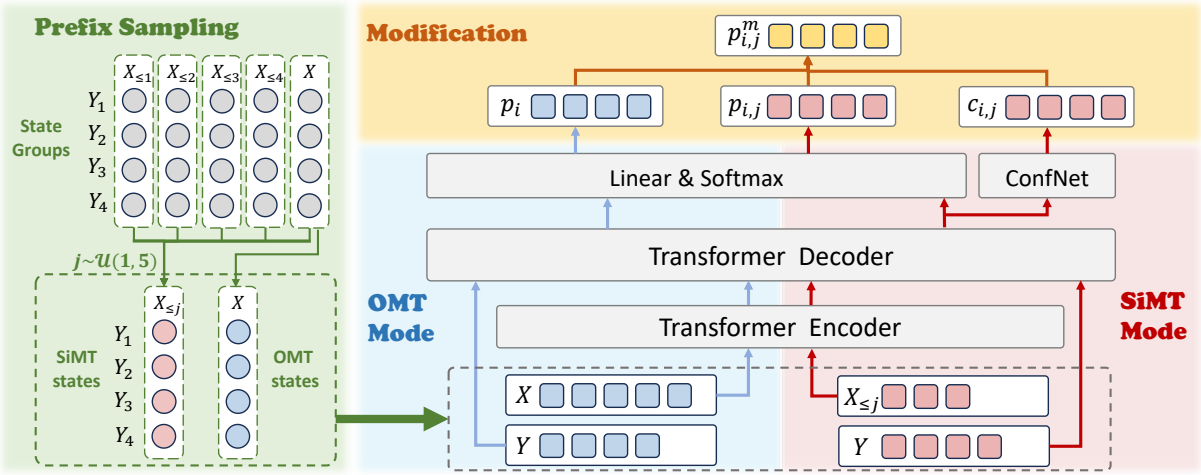


Figure 3: Overview of State-based Policy Learning. The "ConfNet" is our added confidence net. In this figure, we set the lengths of the source sentence and target sentence respectively $|X| = 5$, $|Y| = 4$, and $j = 3$ in the prefix sampling step.

3 The Proposed Method

We propose **S**tate-based **P**olicy **L**earning (SPL), which enables the SiMT model to learn a read/write policy directly on decision states without constructing complete decision paths. During training, SPL evaluates the reasonable decisions of all potential decision states with unsupervised confidence estimation (Sec.3.1). During inference, SPL is allowed to generate a target token when it is confident, or SPL needs to wait for more source inputs (Sec.3.2).

3.1 State-based Policy Learning

Confidence Estimation Due to the challenge of obtaining labeled $d_{i,j}$ for $s_{i,j}$, it is difficult to learn an effective policy through supervised learning. Borrowing the idea from *Ask For Hints* (DeVries and Taylor, 2018; Lu et al., 2022), SPL evaluates and optimizes the suitable decision for each decision state with unsupervised confidence estimation.

Intuitively, if a translation model generates a correct prediction of y_i at $s_{i,j}$ (i.e. SiMT mode), it is expected to maintain this prediction even after receiving the complete X (i.e. OMT mode). Conversely, if the prediction in SiMT mode is incorrect, the model in OMT mode should rectify this prediction. Based on this insight, we develop a framework to quantitatively assess the confidence of the predictions at each decision state. Specifically, the SiMT model is provided with the opportunity to modify its output probability with the prediction in OMT mode, and the confidence estimation task is modeled as a binary classification determining whether the current generation requires the modification. High confidence means the SiMT model can generate a correct target token at $s_{i,j}$, whereas low confidence indicates additional source tokens are necessary for accurate prediction.

To enable the training of confidence estimation, the modification process is modeled, which is shown in Figure 3. The modeling is conducted in two steps: firstly, prediction in SiMT mode & OMT mode; and secondly, confidence-based modification. As an example, we will introduce details of these steps at $s_{i,j}$ in the following.

For *prediction in SiMT mode& OMT mode*, the SiMT model predicts y_i at $s_{i,j}$ in SiMT mode and OMT mode respectively. These predictions are calculated as follows:

$$\begin{aligned} p_{i,j} &= p(y_i \mid \mathbf{x}_{\leq j}, \mathbf{y}_{<i}) \\ p_i &= p(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) \end{aligned} \tag{3}$$

where $p_{i,j}$ is the predicted by the SiMT model based on $s_{i,j}$, and p_i is the prediction based on the complete $\mathbf{x}$.

For *confidence-based modification*, we add an additional confidence net to predict the confidence $c_{i,j}$ at $s_{i,j}$. The confidence net is represented as:

$$c_{i,j} = \text{sigmoid}(W^T \cdot h_{i,j} + b) \tag{4}$$

where $h_{i,j}$ is the hidden representation from the top decoder layer in SiMT mode and $\theta = \{W, b\}$ are trainable parameters. If the prediction in SiMT mode is accurate, $c_{i,j}$ should be close to 1. Otherwise, $c_{i,j}$ should be close to 0. To optimize $c_{i,j}$ in the training process, the modification is integrated

Algorithm 1: Confidence-based Policy

Input : Streaming inputs $\mathbf{x}_{\leq j}$, Threshold γ, $i = 1$, $j = 1$, $y_0 \leftarrow \langle \text{BOS} \rangle$

Output : Target outputs $\mathbf{y}$

```
while y_{i-1} ≠ ⟨EOS⟩ do
    calculate confidence c_{i,j} as Eq.(4);
    if c_{i,j} ≥ γ then                    // WRITE
        translate y_i with x_{≤j}, y_{<i};
        i ← i + 1;
    else                                   // READ
        wait for next source token x_{j+1};
        j ← j + 1;
    end
end
```

in the following manner:

$$p^m_{i,j} = c_{i,j} \cdot p_{i,j} + (1 - c_{i,j}) \cdot p_i \tag{5}$$

Subsequently, the translation loss is calculated using the modified probability:

$$\mathcal{L}_{s_{i,j}} = -y_i \log(p^m_{i,j}) \tag{6}$$

Notably, the SiMT model can achieve better translation by predicting a lower $c_{i,j}$. However, this approach may cause the over-reliance on the modification. To avoid that, an additional penalty term for $c_{i,j}$ is introduced:

$$\mathcal{L}_{c_{i,j}} = -\log(c_{i,j}) \tag{7}$$

Prefix Sampling During training, it is necessary to systematically explore all potential decision states so that the policy can be optimized comprehensively. Therefore, prefix sampling is used in SPL. As shown in Figure 3. the states are divided into groups, which are alternately evaluated and optimized during the training process.

Specifically, all possible decision states of (X, Y) can be divided into M groups according to their received source tokens, with each group comprising N states:

$$S_j = \{s_{i,j} \mid 1 \leq i \leq N\}, j \in [1, M] \tag{8}$$

In each iteration, we sample $j \sim \mathcal{U}(1, M)$, predict the target tokens in SiMT mode based on S_j, and predict target tokens based on OMT states S_M. The translation loss and the penalty item of each iteration are computed as follows:

$$\begin{aligned} \mathcal{L}_s &= \sum_{i=1}^{N} \mathcal{L}_{s_{i,j}} \\ \mathcal{L}_c &= \sum_{i=1}^{N} \mathcal{L}_{c_{i,j}} \end{aligned} \tag{9}$$

Besides, to ensure the modification from OMT is reliable, the cross entropy loss in OMT mode is needed, which is formulated as follows:

$$\mathcal{L}_{omt} = -\sum_{i=1}^{N} \log(p_i) \tag{10}$$

The total training loss is the following:

$$\mathcal{L} = \mathcal{L}_{omt} + \mathcal{L}_s + \lambda \mathcal{L}_c \tag{11}$$

where λ is the super parameter controlling the weight of $\mathcal{L}_c$. We set $\lambda = 0.1$ for the following experiments.

3.2 Confidence-based Policy in Inference

During inference, the SiMT model makes decisions between READ and WRITE based on predicted $c_{i,j}$ at $s_{i,j}$. High $c_{i,j}$ means the SiMT model is confident in current prediction and generating y_i is credible. In the meanwhile, if $c_{i,j}$ is low, it is reasonable to wait for more source tokens before continuing the translation. Therefore, a confidence threshold γ is introduced as a criterion for making decisions. As shown in Algorithm 1, if $c_{i,j} > \gamma$, SPL selects WRITE; otherwise, SPL selects READ. This decision process is constantly repeated until the complete translation is finished.

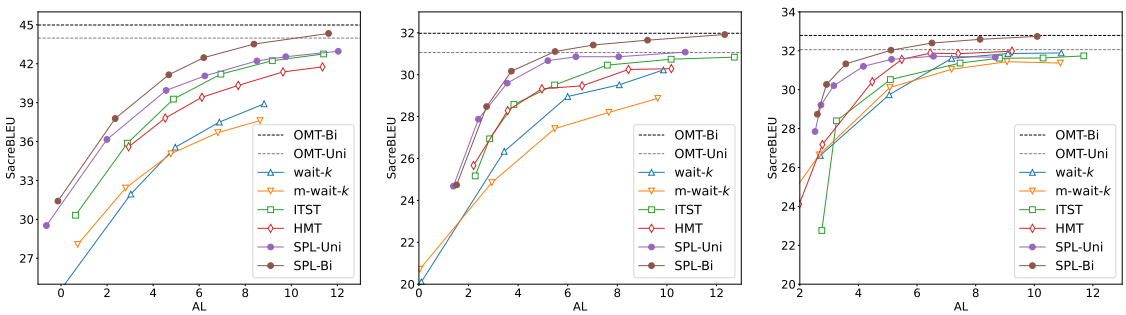
By adjusting γ, we can enable the SiMT model to perform the SiMT task under different latency levels. A higher γ encourages the SiMT model to predict more accurately and the latency will be longer. Conversely, a lower γ reduces the latency but may lead to a decrease in translation quality.

4 Experiments

4.1 Datasets

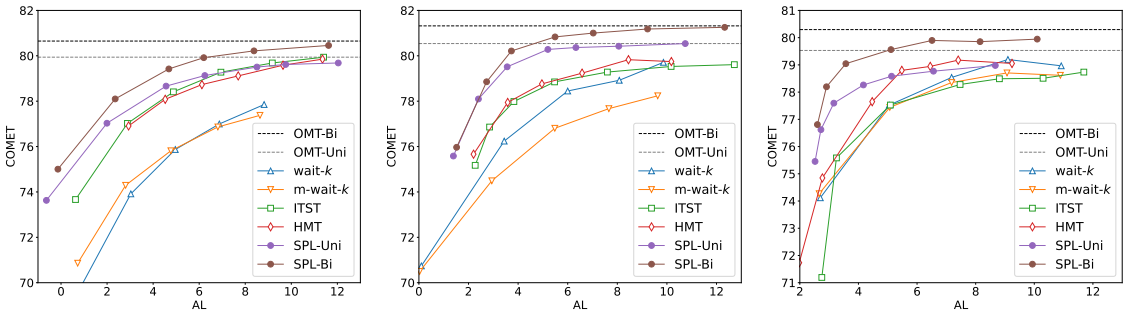
We conduct experiments on three datasets:

Zh→En We use LDC corpus which contains 2.1M sentence pairs as the training set, NIST 2008 for the validation set and NIST 2003, 2004, 2005, and 2006 for the test sets.



(a) Zh→En (b) De→En (c) En→Ro

Figure 4: SacreBLEU against Average Lagging (AL) on Zh→En, De→En and En→Ro.



(a) Zh→En (b) De→En (c) En→Ro

Figure 5: COMET against Average Lagging (AL) on Zh→En, De→En and En→Ro.

De→En We choose WMT15 for training, which contains 4.5M sentence pairs. Newstest 2013 are used as the validation set and newstest 2015 are used as the test set.

En→Ro WMT16 (0.6M) is used as the training set. We choose newsdev2016 as the validation set and newstest2016 as the test test.

We apply BPE (Sennrich et al., 2016) for all language pairs. In Zh→En, the vocabulary size is 30k for Chinese and 20k for English. In both De→En and En→Ro, a shared vocabulary is learned with 32k merge operations.

4.2 Settings

System Settings The models we used are introduced as follows. We use greedy search during inference for all baselines. All baselines are built based on Transformer (Vaswani et al., 2017) with the unidirectional encoder unless otherwise stated. More details are put in Appendix A.

OMT: Transformer performing the OMT task. We implement the Transformer with a unidirectional encoder (OMT-Uni) and with a bidirectional encoder (OMT-Bi) respectively.

wait-k (Ma et al., 2019): a fixed policy, which first reads k tokens, then writes one token and reads one token in turns.

m-wait-k (Elbayad et al., 2020): an improved version of wait-k, which randomly samples different values of k during training.

ITST (Zhang and Feng, 2022a): an adaptive policy is learned by quantifying the transported information weight that each source token accounts to current target token.

HMT (Zhang and Feng, 2023): the simultaneous translation task is modeled as a hidden Markov model. The decision states as are treated as hidden events and the predicted target tokens are treated as observed events.

SPL: Our propose method. We implement SPL with a unidirectional encoder (SPL-Uni) and with a bidirectional encoder (SPL-Bi) respectively.

Evaluation Metric For SiMT, both translation quality and latency require evaluation. We measure the translation quality with SacreBLEU (Post, 2018) and COMET (Rei et al., 2020) respectively. Besides, Average Lagging (AL) (Ma et al., 2019) is used for the latency evaluation.

4.3 Main Results

The SacreBLEU scores of different models are shown in Figure 4. Both SPL-Uni and SPL-Bi out-

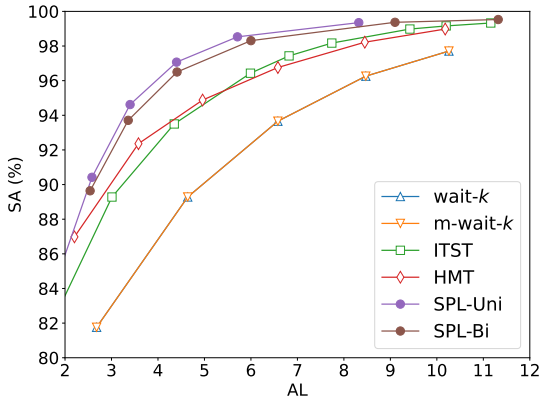
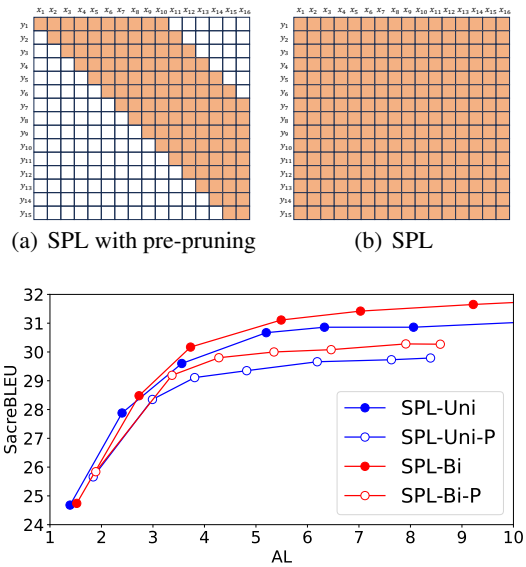


Figure 6: Evaluation of different SiMT policies. We calculate SA (↑) under different latency levels.



(a) SPL with pre-pruning (b) SPL

(c) Effect of pre-pruning to the performance of SPL.

Figure 7: The visualization and effect of pre-pruning.

perform previous methods across three language pairs. Compared to current state-of-the-art SiMT models like ITST and HMT, SPL-Uni achieves higher SacreBLEU scores at low latency levels and comparable performance at high latency levels. We attribute this improvement to the effectiveness of SPL in exploring all decision states in training and comprehensively optimizing the policy. Furthermore, SPL-Bi outperforms previous SiMT models significantly, attributed to the superior capabilities of the bidirectional encoder. It can be observed that all SiMT models with unidirectional encoders can approach the translation quality of OMT-Uni at high latency levels, but only SPL-Bi achieves similar performance to OMT-Bi as the latency increases.

Since existing datasets mainly focus on the OMT task, the metric based on n-gram may cause inaccurate evaluation (Rei et al., 2020). Therefore, we also evaluate the translation quality with COMET[1] scores. The results in Figure 5 show similar results, which proves the advantage of SPL in directly exploring all decision states.

It is worth noting that the performance gap between SPL-Bi and SPL-Uni becomes larger as the latency increases, which reflects that reasonable policy matters more at low latency levels. When the latency becomes higher, the better source representations from the bidirectional encoder will improve the translation quality more obviously.

5 Analysis

5.1 Evaluation of Read/Write Policy

Following Zhang and Feng (2022a) and Kim and Cho (2023), we quantitatively access the quality of different SiMT policies by calculating the proportion of the ground-truth aligned source tokens received before translating on RWTH[2], a De→En alignment dataset. Specifically, when generating y_i, the number of received source tokens g_i should be no less than the golden-truth aligned source position a_i, so that the alignment between y_i and x_{a_i} can be satisfied in the SiMT process. The proportion of the Satisfied Alignments (SA) can be calculated as:

$$\mathrm{SA} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(a_i \leq g_i) \tag{12}$$

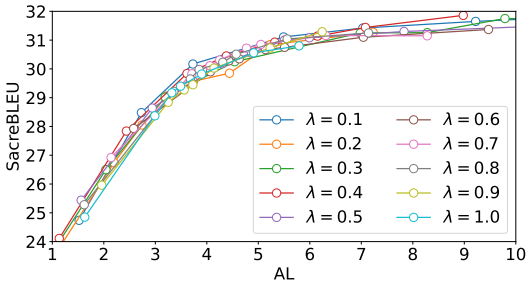
The results are presented in Figure 6. Compared with fixed policy (wait-k, m-wait-k) and existing adaptive policy (ITST, HMT), SPL receives more aligned source tokens before deciding to generate target tokens across all latency levels. This indicates that SPL learns a better read/write policy, which can read enough source tokens to generate faithful target tokens and reduce unnecessary latency. Besides, the SA of SPL-Uni and SPL-Bi are close. This shows that unidirectional encoder does not hinder the learning of SiMT policy.

5.2 Superiority of Exploring all States

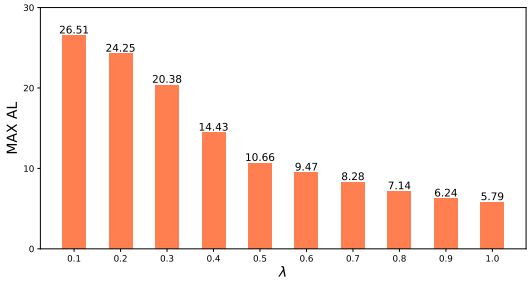
Existing adaptive policies often adopt constraints during the policy learning process for the stability of training (Miao et al., 2021; Zhang and Feng,

[1] Unbabel/wmt22-cometkiwi-da

[2] https://www-i6.informatik.rwth-aachen.de/goldAlignment/



(a) Performance of SPL with different λ.



(b) Max Latency of SPL with different λ.

Figure 8: Effect of λ on SPL.

$c_{i,j}$	(0,0.2]	(0.2,0.4]	(0.4,0.6]	(0.6,0.8]	(0.8,1.0]
p_i	66.67	60.51	57.06	58.68	81.75
$p^m_{i,j}$	66.70	60.46	56.57	58.49	81.77
$p_{i,j}$	13.79	36.42	52.13	57.87	81.65

Table 1: The accuracy of different predictions under different confidence levels.

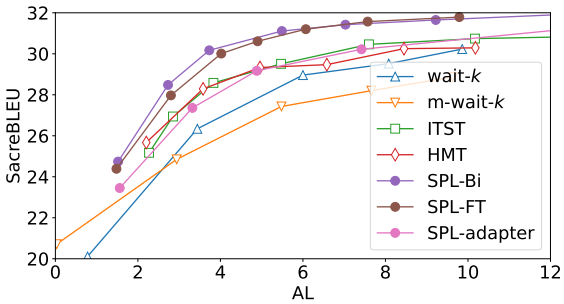


Figure 9: Comparison between existing SiMT models and OMT models fine-tuned according to SPL.

2023). To further analyze the advantages of exploring all the potential decision states, we analyze the effect of pre-pruning constraints on SPL, which is usually used in existing methods (Zheng et al., 2019; Miao et al., 2021; Zhang and Feng, 2023). Specifically, as shown in Figure 7(a), only the decision states between wait-k_1 and wait-k_2 paths are allowed for SPL to explore. We set $k_1 = 1$ and $k_2 = 10$ in our experiments.

The performance of SPL with pre-pruning (SPL-Uni-P and SPL-Bi-P) is shown in Figure 7(c), from which we can observe the performance drops under the constraints of pre-pruning. This indicates that exploring all potential decision states without constraints is beneficial for improving the ability of the SiMT model.

5.3 Confidence Analysis

To analyze whether the predicted confidence can reflect the credibility of current predictions, we calculate the token-level prediction accuracy of SPL in different confidence levels. As shown in Table 1, we can see that the accuracy of the SiMT model increases with the confidence, which indicates that the confidence can be used as the evaluation of current prediction quality, thus guiding the SiMT model to reasonably decide between READ and WRITE at each decision state. Besides, with the modification of $c_{i,j}$, $p^m_{i,j}$ achieves comparable accuracy with p_i, which shows that the confidence is trained according to our design.

5.4 Effect of λ

We analyze the effect of λ, which is the weight of the penalty during training. We train SPL with different λ ranging from 0.1 to 1, in increments of 0.1. As shown in Figure 8(a), the SPL models trained with different λ show comparable performance across all latency. This indicates that the performance of SPL is robust to variations in hyperparameters λ. When λ becomes larger, the corresponding γ at the same latency will also increase. Therefore, we further analyze the effect of λ on the applicable latency range of SPL. We denote the "MAX AL" as the latency of SPL when γ is set as 0.99 during inference. The results are shown in Figure 8(b). When λ becomes larger, "MAX AL" also decreases, which means a smaller applicable latency range. For example, when $\lambda = 1.0$ in training, it is hard for SPL to perform SiMT task under the latency levels where AL is larger than 5.79 since the threshold γ has been close to 1.

5.5 Compatibility of SPL

As SPL does not require constructing complete decision paths in the training process, it allows for the parallel training of the bidirectional encoder without augmenting computational complexity. Due to this compatibility, existing pre-trained OMT models can be fine-tuned according to SPL to gain the capability of performing the SiMT task. We

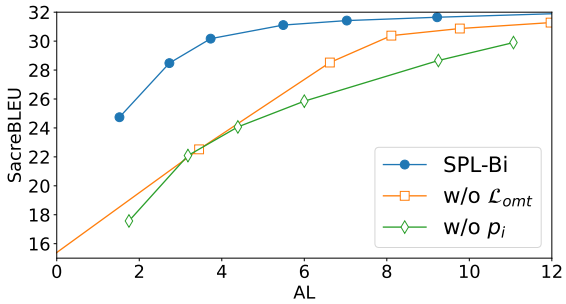


Figure 10: Effect of $\mathcal{L}_{omt}$ and OMT modification on the SiMT ability."w/o $\mathcal{L}_{omt}$" is SPL trained without $\mathcal{L}_{omt}$, and "w/o p_i" means SPL trained using one-hot rather than OMT prediction for modification.

	SPL-Bi	w/o $\mathcal{L}_{omt}$	w/o p_i	OMT
SacreBLEU	31.87	30.33	31.95	31.93

Table 2: Effect of $\mathcal{L}_{omt}$ and p_i on the OMT ability.

fine-tune the OMT model using two distinct methods: fine-tuning all model parameters (SPL-FT) and fine-tuning with adapters (SPL-adapter). As shown in Figure 9, SPL-adapter can achieve comparable performance with current state-of-the-art SiMT models, and SPL-FT closely matches the performance of SPL-Bi.

5.6 Ablation Study

We conduct ablation studies on SPL, investigating the effect of $\mathcal{L}_{omt}$ and OMT modification.

Effect of $\mathcal{L}_{omt}$ As shown in Figure 10, the SiMT model without $\mathcal{L}_{omt}$ drops quickly. We argue this is because training without $\mathcal{L}_{omt}$ may cause a worse modification. The results in Table 2 show that the OMT performance of SPL trained without $\mathcal{L}_{omt}$ is significantly affected, even worse than its SiMT performance in the high latency levels. This poor OMT ability cannot provide accurate modification, thus disrupting the policy learning process.

Effect of OMT modification Following *Ask For Hints* (DeVries and Taylor, 2018; Lu et al., 2022), we use the one-hot label as the "hints" to modify the prediction in SiMT mode. We denote the ground-truth label of the i-th target token as t_i, and the modification in SPL is adjusted as shown:

$$p^m_{i,j} = c_{i,j} \cdot p_{i,j} + (1 - c_{i,j}) \cdot t_i \qquad (13)$$

As shown in Figure 10, the performance of SPL trained with modification in Eq.(13) also drops. We argue that the modification from t_i cannot reflect the real available gain from the modification of the SiMT model after receiving the complete source sentence, thus learning a worse policy.

6 Related Works

Simultaneous Machine Translation Existing SiMT methods are divided into fixed policy and adaptive policy. For fixed policy, Ma et al. (2019) proposed wait-k, which instructs the SiMT model to read k tokens first, and alternately perform "write" and "read" actions. Elbayad et al. (2020) proposed multipath wait-k, which randomly samples k during training. For adaptive policy, heuristic rules Cho and Esipova (2016) and reinforcement learning Gu et al. (2017) are used to realize the SiMT task. Ma et al. (2020) integrated monotonic attention into Transformer to model the decision process. Miao et al. (2021) proposed a generative framework to realize SiMT with a OMT model. Zhang and Feng (2022a) added a module measuring information SiMT had received and proposed an information-based policy. Zhang and Feng (2023) used the Hidden Markov Chain to model SiMT task and proposed a policy based on predicted confidence.

Previous policy learning methods are based on exploring various decision paths, which is challenging and inefficient. Conversely, our proposed SPL directly explores all decision states, which learns a better policy more efficiently.

Confidence Estimation for OMT Confidence estimation is used to measure the models' reliability in current prediction. Wang et al. (2019) use Monte Carlo dropout to propose an uncertainty-based confidence estimation in OMT. Wan et al. (2020) utilize the confidence score to guide self-paced learning in OMT. DeVries and Taylor (2018) propose to evaluate the model's confidence by measuring the level it asks for hints from the ground-true label, and Lu et al. (2022) transfer this method to machine translation to improve the out-of-distribution detection.

7 Conclusion

In this paper, we propose **S**tate-based **P**olicy **L**earning (SPL), which is a novel paradigm for policy learning. During training, SPL directly explores all potential decision states without constructing complete decision paths, which can perform policy learning more efficiently. Experiments on three language pairs show the superiority of SPL in terms of performance, learned policy and compatibility.

8 Limitations

In this paper, we propose SPL, which learns a read/write policy by directly exploring all potential decision states. Although our experiments show the superiority of the learning policy in a state-based way, there are still some parts to be further improved. For example, using a more effective way to learn the reasonable decisions for each decision state. Besides, how to improve the ability of existing pre-trained OMT models is still a promising direction to reduce the increase in training costs. These will be considered as objectives for our future work.

Acknowledgements

References

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv e-prints*, pages arXiv–1606.

Terrance DeVries and Graham W Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient wait-k models for simultaneous machine translation.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK 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.

Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. 2022. From simultaneous to streaming machine translation by leveraging streaming history. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6972–6985.

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

Yu Lu, Jiali Zeng, Jiajun Zhang, Shuangzhi Wu, and Mu Li. 2022. Learning confidence for transformer-based neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2353–2364.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, et al. 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.

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

Yishu Miao, Phil Blunsom, and Lucia Specia. 2021. A generative framework for simultaneous machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6697–6706.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, page 186. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

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.

Yu Wan, Baosong Yang, Derek F Wong, Yikai Zhou, Lidia S Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. *arXiv preprint arXiv:2010.04505*.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. *arXiv preprint arXiv:1909.00157*.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289.

Shaolei Zhang and Yang Feng. 2021. Universal simultaneous machine translation with mixture-of-experts wait-k policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7306–7317.

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*, pages 992–1013, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022b. Modeling dual read/write paths for simultaneous machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2461–2477.

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

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.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simultaneous translation with flexible policy via restricted imitation learning. *arXiv preprint arXiv:1906.01135*.

A Hyper-parameters

The system settings in our experiments are shown in Table 3.

B Numerical Results

Table 4, 5, 6 respectively report the numerical results on LDC Zh→En, WMT15 De→En, MWT16 En→Eo measured by AL, SacreBLEU and COMET.

Hyper-parameter	
encoder layers	6
encoder attention heads	8
encoder embed dim	512
encoder ffn embed dim	1024
decoder layers	6
decoder attention heads	8
decoder embed dim	512
decoder ffn embed dim	1024
dropout	0.1
optimizer	adam
adam-β	(0.9, 0.98)
clip-norm	1e-7
lr	5e-4
lr scheduler	inverse sqrt
warmup-updates	4000
warmup-init-lr	1e-7
weight decay	0.0001
label-smoothing	0.1
max tokens	8192

Table 3: Hyper-parameters of our experiments.

Zh→En							
wait-k	AL	-0.60	3.03	4.96	6.87	8.82	
	SacreBLEU	23.14	31.94	35.56	37.50	38.90	
	COMET	67.06	73.91	75.87	76.99	77.85	
m-wait-k	AL	0.72	2.80	4.76	6.81	8.64	
	SacreBLEU	28.06	32.41	35.05	36.68	37.61	
	COMET	70.85	74.29	75.81	76.86	77.37	
HMT	AL	2.93	4.52	6.11	7.69	9.64	11.35
	SacreBLEU	35.59	37.81	39.41	40.33	41.37	41.75
	COMET	76.90	78.08	78.73	79.11	79.58	79.85
ITST	AL	0.62	2.88	4.88	6.94	9.17	11.40
	SacreBLEU	30.31	35.87	39.27	41.20	42.23	42.75
	COMET	73.66	77.02	78.41	79.27	79.68	79.93
SPL-Uni	AL	-0.63	1.99	4.56	6.24	8.51	9.75
	SacreBLEU	29.52	36.16	39.94	41.06	42.21	42.54
	COMET	73.62	77.02	78.66	79.13	79.50	79.61
SPL-Bi	AL	-0.14	2.35	4.68	6.19	8.37	11.61
	SacreBLEU	31.41	37.77	41.15	42.47	43.51	44.34
	COMET	75.00	78.09	79.42	79.91	80.21	80.45

Table 4: Numerical results on LDC Zh→En.

De→En								
wait-k	AL	0.10	3.44	6.00	8.08	9.86		
	SacreBLEU	20.11	26.34	28.96	29.52	30.23		
	COMET	70.74	76.24	78.44	78.92	79.71		
m-wait-k	AL	0.03	2.94	5.48	7.66	9.63		
	SacreBLEU	20.71	24.85	27.43	28.2	28.87		
	COMET	70.49	74.49	76.80	77.67	78.23		
HMT	AL	2.20	3.58	4.96	6.58	8.45	10.18	
	SacreBLEU	25.67	28.29	29.33	29.47	30.25	30.29	
	COMET	75.66	77.94	78.76	79.23	79.82	79.74	
ITST	AL	2.27	2.85	3.83	5.47	7.60	10.17	12.72
	SacreBLEU	25.17	26.94	28.58	29.51	30.46	30.74	30.84
	COMET	75.17	76.86	77.98	78.85	79.28	79.53	79.61
SPL-Uni	AL	1.39	2.4	3.56	5.2	6.33	8.06	10.74
	SacreBLEU	24.68	27.88	29.6	30.67	30.86	30.86	31.08
	COMET	75.58	78.09	79.51	80.28	80.36	80.42	80.53
SPL-Bi	AL	1.52	2.73	3.73	5.49	7.03	9.22	12.33
	SacreBLEU	24.74	28.48	30.17	31.11	31.42	31.65	31.92
	COMET	75.96	78.85	80.21	80.83	81.00	81.18	81.25

Table 5: Numerical results on WMT15 De→En.

En→Ro								
wait-k	AL	2.70	5.05	7.18	9.10	10.92		
	SacreBLEU	26.62	29.74	31.61	31.86	31.89		
	COMET	74.12	77.52	78.54	79.20	78.97		
m-wait-k	AL	2.66	5.07	7.18	9.07	10.89		
	SacreBLEU	26.65	30.11	31.05	31.44	31.37		
	COMET	74.27	77.44	78.35	78.71	78.62		
HMT	AL	1.98	2.77	4.47	5.48	6.45	7.41	9.24
	SacreBLEU	24.11	27.18	30.41	31.56	31.88	31.85	31.98
	COMET	71.73	74.85	77.65	78.80	78.94	79.17	79.05
ITST	AL	2.75	3.25	5.09	7.47	8.81	10.30	11.69
	SacreBLEU	22.76	28.40	30.52	31.37	31.62	31.63	31.74
	COMET	71.19	75.58	77.53	78.28	78.49	78.51	78.73
SPL-Uni	AL	2.52	2.72	3.16	4.17	5.13	6.56	8.67
	SacreBLEU	27.85	29.21	30.21	31.20	31.56	31.72	31.67
	COMET	75.45	76.62	77.59	78.26	78.58	78.77	78.98
SPL-Bi	AL	2.60	2.91	3.57	5.11	6.51	8.15	10.10
	SacreBLEU	28.74	30.27	31.33	32.03	32.40	32.59	32.74
	COMET	76.81	78.20	79.04	79.56	79.90	79.85	79.95

Table 6: Numerical results on WMT16 En→Ro.