Self-Modifying State Modeling for Simultaneous Machine Translation

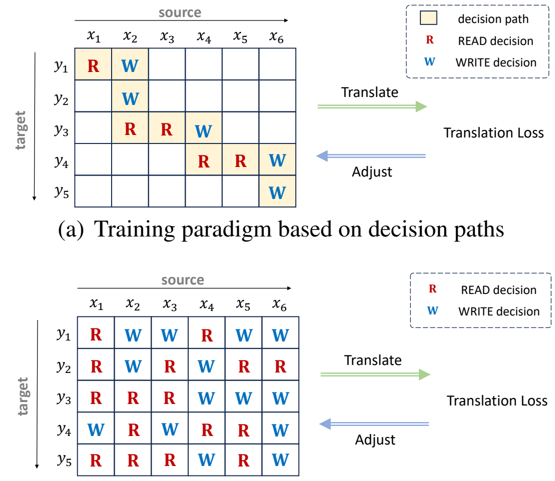
Donglei Yu[1,2], **Xiaomian Kang**[1,2], **Yuchen Liu**[1,2], **Yu Zhou**[1,3*], **Chengqing Zong** [1,2]
[1]State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] Fanyu AI Laboratory, Zhongke Fanyu Technology Co., Ltd, Beijing, China
{yudonglei2021,kangxiaomian2014,yuchen.liu,yu.zhou,chengqing.zong}@ia.ac.cn

Abstract

Simultaneous Machine Translation (SiMT) generates target outputs while receiving stream source inputs and requires a read/write policy to decide whether to wait for the next source token or generate a new target token, whose decisions form a *decision path*. Existing SiMT methods, which learn the policy by exploring various decision paths in training, face inherent limitations. These methods not only fail to precisely optimize the policy due to the inability to accurately assess the individual impact of each decision on SiMT performance, but also cannot sufficiently explore all potential paths because of their vast number. Besides, building decision paths requires unidirectional encoders to simulate streaming source inputs, which impairs the translation quality of SiMT models. To solve these issues, we propose **S**elf-**M**odifying **S**tate **M**odeling (SM^2), a novel training paradigm for SiMT task. Without building decision paths, SM^2 individually optimizes decisions at each state during training. To precisely optimize the policy, SM^2 introduces Self-Modifying process to independently assess and adjust decisions at each state. For sufficient exploration, SM^2 proposes Prefix Sampling to efficiently traverse all potential states. Moreover, SM^2 ensures compatibility with bidirectional encoders, thus achieving higher translation quality. Experiments show that SM^2 outperforms strong baselines. Furthermore, SM^2 allows offline machine translation models to acquire SiMT ability with fine-tuning [1].

1 Introduction

Simultaneous Machine Translation (SiMT) (Gu et al., 2017; Ma et al., 2019; Zhang et al., 2020) outputs translation while receiving the streaming source sentence. 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).



	x_1	x_2	x_3	x_4	x_5	x_6
y_1	R	W				
y_2		W				
y_3		R	R	W		
y_4				R	R	W
y_5						W

(a) Training paradigm based on decision paths

	x_1	x_2	x_3	x_4	x_5	x_6
y_1	R	W	W	R	W	W
y_2	R	W	R	W	R	R
y_3	R	R	R	W	W	W
y_4	W	R	W	R	R	W
y_5	R	R	R	W	R	W

(b) Our proposed Self-Modifying State Modeling

Figure 1: Illustration of different paradigms. **(a)** Training paradigm based on decision paths. All decisions along a path are optimized in an integrated manner. **(b)** Self-Modifying State Modeling. The decisions at each state are optimized individually.

As shown in Figure 1(a), to learn a suitable policy, existing SiMT methods usually require building a *decision path* (i.e., a series of READ and WRITE decisions made by the policy) to simulate the complete SiMT process during training (Zhang and Feng, 2022c). Methods of fixed policies (Ma et al., 2019; Zhang and Feng, 2021) build the decision path based on pre-defined rules, and only optimize translation quality along the path. Methods of adaptive policies (Zheng et al., 2019; Miao et al., 2021; Zhang and Feng, 2023) dynamically build the decision path and optimize the policy based on the SiMT performance along this path.

However, the current training paradigm based on decision paths faces inherent limitations. First, it can lead to **imprecise optimization** of the pol-

[*] Corresponding Author
[1]Our source code is available at https://github.com/EurekaForNLP/SM2

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9781–9795
August 11-16, 2024 ©2024 Association for Computational Linguistics

icy during training. For fixed policies, pre-defined rules cannot ensure optimal decisions at each state. For adaptive policies, there exists a credit assignment problem (Minsky, 1961), which means it is difficult to identify the impact of each individual decision on the global SiMT performance along a path, thus hindering the precise optimization of each decision. Second, due to numerous potential decision paths, existing methods (Zheng et al., 2019; Miao et al., 2021; Zhang and Feng, 2023) often prohibit the exploration of some paths during training, but this **insufficient exploration** cannot ensure the optimal policy. Third, for building decision paths in training, existing methods require **unidirectional encoders** to simulate streaming source inputs and avoid the leakage of source future information (Elbayad et al., 2020), which impairs SiMT models' translation quality (Iranzo-Sánchez et al., 2022; Kim and Cho, 2023).

To address these issues, we propose **S**elf-**M**odifying **S**tate **M**odeling (SM^2), a novel training paradigm for SiMT task. As shown in Figure 1(b), instead of constructing complete decision paths, SM^2 individually optimizes decisions at all potential states during training. This paradigm necessitates addressing two critical issues: firstly, how to independently optimize each decision based on its own contribution to SiMT performance; and secondly, how to sufficiently explore all potential states during training. To realize the independent optimization, SM^2 assesses each decision by estimating confidence values which measure the translation credibility. High confidence means the SiMT model can predict a credible target token at current state and WRITE is beneficial for SiMT performance; otherwise, READ is preferred. Since golden confidence values are unavailable, SM^2 introduces **Self-Modifying** process to learn accurate confidence estimation (DeVries and Taylor, 2018; Lu et al., 2022). Specifically, during training, the SiMT model is allowed to modify its prediction based on the received source prefix with the prediction based on the complete source sentence, and the confidence is estimated to determine whether the modification is necessary to ensure a credible prediction at current state. To sufficiently explore all potential states, SM^2 conducts **Prefix Sampling** to divide all states into groups according to the number of their received source prefix tokens, and sample one group for optimization in each iteration.

Compared to the training paradigm based on decision paths, SM^2 presents significant advantages. First, the Self-Modifying process can assess each decision independently, which realizes the precise optimization of policy without the credit assignment problem. Second, Prefix Sampling ensures sufficient exploration of all potential states, promoting the discovery of the optimal policy. These benefits enable SM^2 to learn a more effective policy. Furthermore, without building decision paths in training, SM^2 ensures compatibility with bidirectional encoders, thereby improving translation quality. This compatibility also allows OMT models to acquire the SiMT capability via fine-tuning. Our contributions are outlined in the following:

- We propose **S**elf-**M**odifying **S**tate **M**odeling (SM^2), a novel training paradigm that individually optimizes decisions at all states without building complete decision paths.
- SM^2 can learn a better policy through precise optimization of each decision and sufficient exploration of all states. With bidirectional encoders, SM^2 achieves higher translation quality and compatibility with OMT models.
- Experimental results on Zh→En, De→En and En→Ro SiMT tasks show that SM^2 outperforms strong baselines under all latency levels.

2 Background

Simultaneous machine translation For SiMT task, we respectively 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 $p(y_i \mid \mathbf{x}_{\leq g_i}, \mathbf{y}_{<i})$ (Ma et al., 2019). Thus, the decoding probability of $\mathbf{y}$ is given by:

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

Decision state and decision path We define the state s_{ij} as the condition in which the source prefix $\mathbf{x}_{\leq j}$ has been received and the target prefix $\mathbf{y}_{<i}$ has been generated. At s_{ij}, a decision $d_{ij} \in \{\text{WRITE,READ}\}$ can be made based on the context $(\mathbf{x}_{\leq j}, \mathbf{y}_{<i})$ (Grissom II et al., 2014; Gu et al., 2017; Zhao et al., 2023). Specifically, if $\mathbf{x}_{\leq j}$ is sufficient for the SiMT model to predict y_i accurately, d_{ij} should be WRITE; otherwise, d_{ij} should be READ. As shown in Figure 1(a), a series

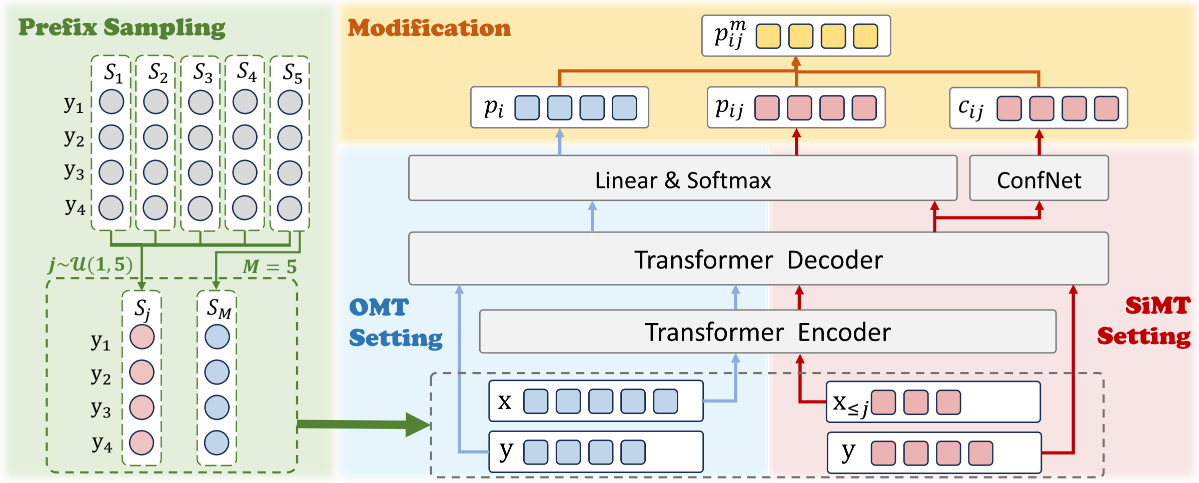


Figure 2: Overview of SM^2. S_j contains the states where $\mathbf{x}_{\leq j}$ is received. S_M contains the states where complete $\mathbf{x}$ is received. We introduce a confidence net (ConfNet) to estimate the confidence of each state. The model parameters in SiMT setting and OMT setting are shared. In this figure, the sentence lengths of the source and target sides are set to $M = 5$ and $N = 4$ respectively, and $j = 3$ in the Prefix Sampling step.

of decisions $[d_{00}, ..., d_{NM}]$ are made in the SiMT process, which forms a decision path from s_{00} to s_{NM}. Along the decision path, the SiMT model can finish reading the whole $\mathbf{x}$ and outputting the complete $\mathbf{y}$ (Zhang and Feng, 2022c). Since these concepts are usually used in SiMT methods based on reinforcement learning (RL) (Grissom II et al., 2014; Gu et al., 2017), we compare our method with RL-based methods in Appendix A for clarity.

3 The Proposed Method

We propose **S**elf-**M**odifying **S**tate **M**odeling (SM^2), which individually optimizes decisions at all states. The overview of SM^2 is shown in Figure 2. To independently optimize each decision, SM^2 learns confidence estimation to assess decisions at each state by modeling the Self-Modifying process (Sec.3.1). To ensure sufficient exploration during training, SM^2 conducts Prefix Sampling to traverse all potential states (Sec. 3.2). Then, based on estimated confidence at each state, SM^2 can determine whether the received source tokens are sufficient to generate a credible token and make suitable decisions during inference (Sec.3.3).

3.1 Self-Modifying for Confidence Estimation

Intuitively, when a translation model has access to the complete input $\mathbf{x}$ (i.e., OMT setting), it can produce credible outputs. Therefore, a prediction made by the translation model at s_{ij} (i.e. SiMT setting) is considered credible if it aligns with that in OMT setting. Conversely, if the prediction in SiMT setting is incredible, it will be modified in OMT setting. Based on this insight and *Ask For Hints* (DeVries and Taylor, 2018; Lu et al., 2022), we model the Self-Modifying process to assess the translation credibility of each state. Specifically, we provide the SiMT model an option to modify its prediction in SiMT setting with that in OMT setting, and confidence estimation is defined as a binary classification determining whether the current generation requires the modification to ensure a credible prediction. Through measuring translation credibility, decisions at each state can be independently assessed. High confidence means the SiMT model can generate a credible token at s_{ij} without modification and the WRITE decision is beneficial for SiMT performance; whereas low confidence indicates the prediction is inaccurate at s_{ij} and the READ decision is preferred.

During training, the Self-Modifying process is conducted in two steps: *prediction in SiMT setting & OMT setting* and *confidence-based modification*.

For *prediction in SiMT setting& OMT setting*, the SiMT model outputs different predictions at s_{ij} in SiMT setting and OMT setting respectively. These predictions are calculated as follows:

$$\begin{aligned} p_{ij} &= p(y_i \mid \mathbf{x}_{\leq j}, \mathbf{y}_{<i}) \\ p_i &= p(y_i \mid \mathbf{x}, \mathbf{y}_{<i}) \end{aligned} \tag{2}$$

It is noted that the model parameters in SiMT setting and OMT setting are shared.

For *confidence-based modification*, an additional confidence net is used to predict the confidence c_{ij}

Algorithm 1: Confidence-based Policy

```
Input  : Streaming inputs $\mathbf{x}_{\leq j}$, Threshold
         $\gamma, i = 1, j = 1, y_0 \leftarrow \langle \text{BOS} \rangle$
Output : Target outputs $\mathbf{y}$
while $y_{i-1} \neq \langle \text{EOS} \rangle$ do
    calculate confidence $c_{ij}$ as Eq.(3);
    if $c_{ij} \geq \gamma$ then                 // WRITE
        generate $y_i$ with $\mathbf{x}_{\leq j}, \mathbf{y}_{<i}$;
        $i \leftarrow i + 1$;
    else                                     // READ
        wait for next source token $x_{j+1}$;
        $j \leftarrow j + 1$;
    end
end
```

at s_{ij}. The confidence net is represented as:

$$c_{ij} = \text{sigmoid}(W^T \cdot h_{ij} + b) \tag{3}$$

where h_{ij} is the hidden representation from the top decoder layer in SiMT setting and $\theta = \{W, b\}$ are trainable parameters. If p_{ij} is credible, c_{ij} should be close to 1; otherwise, c_{ij} should be close to 0. To accurately calibrate c_{ij} in the training process, we integrate the modification into the prediction probability as follows:

$$p_{ij}^m = c_{ij} \cdot p_{ij} + (1 - c_{ij}) \cdot p_i \tag{4}$$

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

$$\mathcal{L}_{s_{ij}} = -y_i \log(p_{ij}^m) \tag{5}$$

Notably, the SiMT model can enhance the prediction credibility by estimating a lower c_{ij} for more modification. However, this manner may cause an over-reliance on p_i. To avoid that, an additional penalty term for c_{ij} is introduced:

$$\mathcal{L}_{c_{ij}} = -\log(c_{ij}) \tag{6}$$

Through Self-Modifying process, SM^2 independently optimizes each decision based on their individual effect on the SiMT performance, thus realizing the precise optimization of the policy without credit assignment problem. We provide a gradient analysis of the independent optimization in Appendix B for further explanation.

3.2 Prefix Sampling

To sufficiently explore all potential states during training, Prefix Sampling is conducted in SM^2. As shown in Figure 2, states are categorized into groups, and one group is randomly sampled for optimization in each iteration. Specifically, all possible states of $(\mathbf{x}, \mathbf{y})$ are divided into M groups according to the number of their received source prefix tokens, and each group comprises N states, which can be formulated as follows:

$$S_j = \{s_{ij} \mid 1 \leq i \leq N\}, j \in [1, M] \tag{7}$$

In each iteration, we sample $j \sim \mathcal{U}(1, M)$. Then, SM^2 respectively predicts target translation in SiMT setting based on S_j and those in OMT setting based on S_M, where the complete source sentence is received. Thus, the modified translation loss and the penalty item of each iteration are computed as follows:

$$\begin{aligned} \mathcal{L}_{S_j} &= \sum_{i=1}^{N} \mathcal{L}_{s_{ij}} \\ \mathcal{L}_{C_j} &= \sum_{i=1}^{N} \mathcal{L}_{c_{ij}} \end{aligned} \tag{8}$$

Besides, to ensure the p_i in OMT setting can provide effective modification, the translation loss in OMT setting is required, which is formulated as:

$$\mathcal{L}_{omt} = -\sum_{i=1}^{N} \log(p_i) \tag{9}$$

The total training loss is the following:

$$\mathcal{L} = \mathcal{L}_{omt} + \mathcal{L}_{S_j} + \lambda \mathcal{L}_{C_j} \tag{10}$$

where λ is the super parameter. We discuss the effect of λ in Appendix C.

Through Prefix Sampling, SM^2 explores all potential states without building any decision paths. Therefore, SM^2 can employ bidirectional encoders without the leakage of source future information in the training process.

3.3 Confidence-based Policy in Inference

During inference, SM^2 utilizes c_{ij} to assess the credibility of current prediction, thus making suitable decisions between READ and WRITE at s_{ij}. Specifically, a confidence threshold γ is introduced to serve as a criterion for making decisions. As shown in Algorithm 1, if $c_{ij} > \gamma$, SM^2 selects WRITE; otherwise, SM^2 selects READ. This decision process is constantly repeated until the complete translation is finished. It is noted that we only utilize SiMT setting in the inference process.

By adjusting γ, SM^2 can perform the SiMT task under different latency levels. A higher γ encourages the SiMT model to predict more credible target tokens and the latency will be longer. Conversely, a lower γ reduces the latency but may lead to a decrease in translation quality. The values of γ employed in our subsequent experiments are detailed in Appendix D.

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.

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 newsdev 2016 as the validation set and newstest 2016 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. Additional experiments on WMT15 En→Vi are provided in Appendix E.

4.2 System Settings

The models used in our experiments are introduced as follows. All baselines are built based on Transformer (Vaswani et al., 2017) with the unidirectional encoder unless otherwise stated. More details are presented in Appendix D.

OMT-Uni/OMT-Bi(Vaswani et al., 2017): OMT model with an unidirectional/bidirectional encoder.

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): a fix policy, which improves wait-k by randomly sampling different k during training.

ITST (Zhang and Feng, 2022b): an adaptive policy, which models the SiMT task as a transport problem of information from source to target.

HMT (Zhang and Feng, 2023): an adaptive policy, which models the SiMT task as a hidden Markov model, by treating the states as hidden events and the predicted tokens as observed events.

SM^2-Uni/SM^2-Bi: Our proposed method with an unidirectional/bidirectional encoder.

4.3 Evaluation Metric

For SiMT, both translation quality and latency require evaluation. 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 measure the translation quality with both SacreBLEU (Post, 2018) and COMET[2] scores. For latency evaluation, we choose Average Lagging (AL) (Ma et al., 2019) as the metric.

Furthermore, to assess the quality of read/write policy in different SiMT models, we follow Zhang and Feng (2022b) and Kim and Cho (2023) to use Satisfied Alignments (SA), the proportion of the ground-truth aligned source tokens received before translating. 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. Thus, SA(↑) can be calculated as:

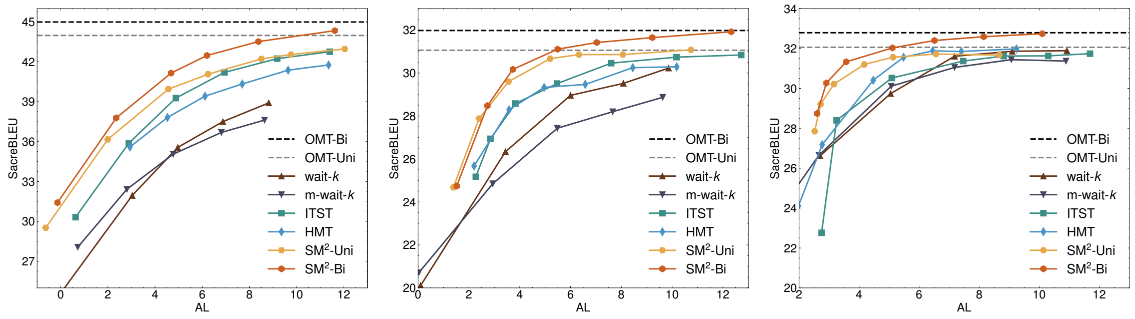
$$\text{SA} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(a_i \le g_i) \tag{11}$$

5 Results and Analysis

5.1 Simultaneous Translation Quality

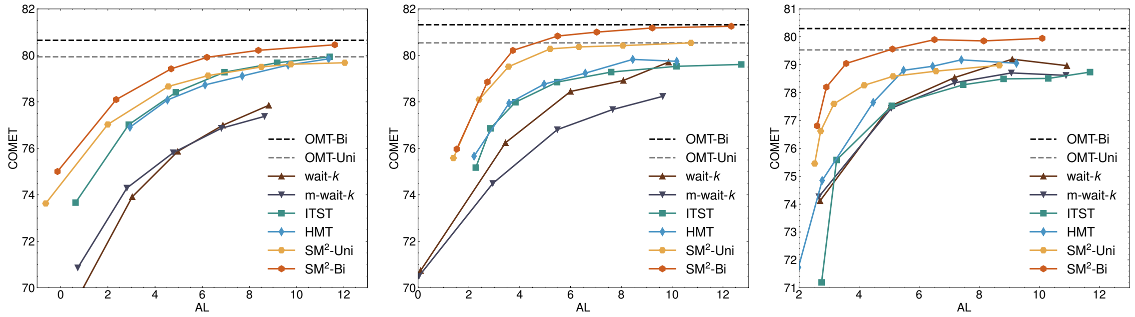
We present the translation quality under various latency levels of different SiMT models in Figure 3 and Figure 4. These results indicate that SM^2 outperforms previous methods across three language pairs in terms of both SacreBLEU and COMET scores. With the unidirectional encoder, SM-Uni achieves higher translation quality compared to current state-of-the-art SiMT models (ITST, HMT) at low and medium latency levels (AL$\in [0, 6]$), and maintains comparable performance at high latency level (AL$\in [6, 12]$). We attribute this improvement to the effectiveness of learning a better policy during training. Furthermore, with the superior capabilities of the bidirectional encoder, SM^2-Bi outperforms previous SiMT models more significantly

[2] Unbabel/wmt22-cometkiwi-da



(a) Zh→En (b) De→En (c) En→Ro

Figure 3: SacreBLEU against Average Lagging (AL) on Zh→En, De→En and En→Ro.



(a) Zh→En (b) De→En (c) En→Ro

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

across all latency levels. All SiMT models with unidirectional encoders can approach the translation quality of OMT-Uni at high latency levels, but only SM^2-Bi achieves similar performance to OMT-Bi as the latency increases. These experimental results prove that SM^2 achieves better performance than other SiMT methods for learning better policy and improving translation quality. Detailed numerical results are provided in Appendix E, supplemented with additional evidence demonstrating the robustness of SM^2 to sentence length variations.

5.2 Superiority of SM^2 in Learning Policy

To verify whether SM^2 can learn a more effective policy, we compare SA(↑) under various latency levels of different SiMT models. Following Zhang and Feng (2022b) and Kim and Cho (2023), we conduct the analysis on RWTH[3], a De→En alignment dataset. The results are presented in Figure 5. Compared with existing methods, both SM^2-Uni and SM^2-Bi receive more aligned source tokens before generating target tokens under the same latency. Especially at medium latency level (AL$\in [4, 6]$), SM^2 can receive about 8% more source tokens than fixed policies (wait-k, m-wait-k) and 3.6% more than adaptive policies (ITST, HMT). We attribute these improvements to the advantages of SM^2 in learning policy. Through precise optimization, SM^2 can make more suitable decisions at each state, which generates faithful translations once receiving sufficient source tokens and waits for more source inputs when the predicted tokens are incredible. With sufficient exploration, SM^2 can investigate all possible situations and reduce unnecessary latency in the SiMT process.

5.3 Precise Optimization for Each Decision

To validate whether the confidence-based policy is precisely optimized at each state, we examine the relationship between estimated confidence c_{ij} and the probability of the correct token y_i in the prediction, denoted as p_{ij}^c. Specifically, we employ SM^2 to decode the validation set in a teacher-forcing manner, calculating the c_{ij} and p_{ij}^c for all possible states. Subsequently, a correlation analysis is performed between c_{ij} and p_{ij}^c. The results in Table 1 demonstrate a strong correlation, evidenced by high values in Pearson (0.82) and Spearman (0.84) coefficients, with a slightly moderate but signifi-

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

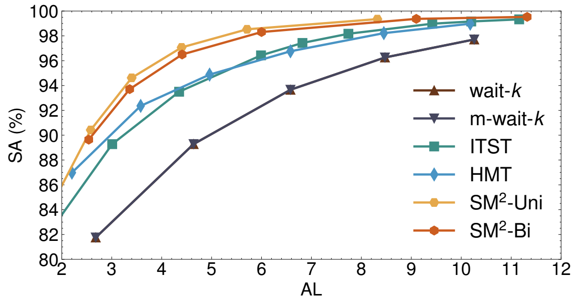


Figure 5: Evaluation of different SiMT policies. We calculate SA (↑) under different latency levels.

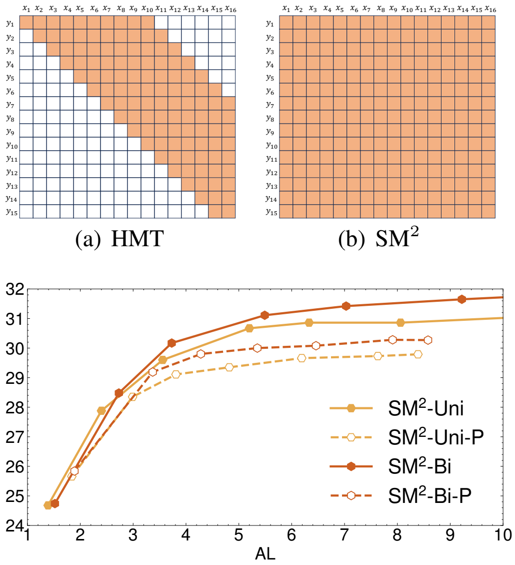
Correlation Coefficient	Pearson	Spearman	Kendall's τ
Value	0.82	0.84	0.65

Table 1: Correlation between c_{ij} and p_{ij}^c.

cant Kendall's τ coefficient (0.65). These results suggest a robust linear and monotonic relationship between c_{ij} and p_{ij}^c, indicating the capacity of c_{ij} to accurately assess the credibility of the current predicted token. Consequently, this confirms the effectiveness of the confidence-based policy in making precise decisions at each state.

5.4 Advantage of Sufficient Exploration

Existing methods often prohibit the exploration of some paths due to the possible decision paths being numerous (Zheng et al., 2019; Miao et al., 2021; Zhang and Feng, 2023). To investigate the impact of the prohibition on SiMT models and the superiority of SM^2 in sufficiently exploring all states, we attempt to train these methods without prohibition, but they fail to converge. Therefore, we analyze the impact by employing the same prohibition in HMT (Zhang and Feng, 2023) and RIL(Zheng et al., 2019) to train SM^2, which restricts SM^2 to explore states only between wait-k_1 and wait-k_2 paths in training. As shown in Figure 6(a), we set $k_1 = 1$ and $k_2 = 10$ in our experiments. The performances of SM^2 with prohibition (SM^2-Uni-P and SM^2-Bi-P) are shown in Figure 6(c), indicating a decline in performance. These results suggest that the prohibition causes insufficient exploration, leading to diminished performance. In contrast, SM^2 ensures comprehensive exploration, which is shown in Figure 6(b), thereby achieving higher performance. Further analysis of the policy quality is provided in Appendix F.



(a) HMT

(b) SM^2

(c) Effect of prohibition to the SiMT performance.

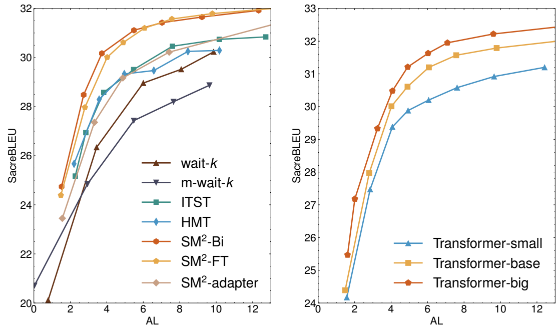
Figure 6: The visualization and effect of prohibition. In (a) and (b), the shaded areas represent the states allowed for exploration in training. We apply the same prohibition in HMT (Zhang and Feng, 2023) to train SM^2-Uni-P and SM^2-Bi-P.

5.5 Compatibility with OMT Models

SM^2 allows for the parallel training of the bidirectional encoder. Due to this compatibility, SM^2-Bi achieves superior translation quality than existing SiMT methods with unidirectional encoders (Figure 3,4). To further present the superiority of this compatibility, we propose fine-tuning OMT models according to SM^2, so that the translation ability in OMT models can be easily utilized to gain SiMT models. Specifically, two distinct methods are used: fine-tuning all model parameters (SM^2-FT) and fine-tuning with adapters (SM^2-adapter)[4]. As shown in Figure 7(a), SM^2-adapter can achieve comparable performance with current state-of-the-art SiMT models, and SM^2-FT closely matches the performance of SM^2-Bi.

Additionally, we further explore the effect of the OMT models' translation abilities on the corresponding SiMT abilities after fine-tuning. We conduct the full-parameter fine-tuning on OMT models with Transformer-small, Transformer-base, and Transformer-big respectively. The OMT and SiMT capabilities of these models are illustrated in Table 2 and Figure 7(b), which reveal that models

[4] We add adapters after the feed-forward networks of each encoder and decoder layer. For each adapter, the input dimension and output dimension are 512, and the hidden layer dimension is 128.



(a) Comparison with SiMT models.

(b) Comparison between different OMT models.

Figure 7: The SiMT performance of different OMT models after fine-tuning according to SM^2.

OMT model	Parameters	SacreBLEU	
		before FT	after FT
Transformer-small	47.9M	30.86	31.33
Transformer-base	60.5M	31.93	31.87
Transformer-big	209.1M	32.99	32.75

Table 2: The OMT performance of different OMT models before/after fine-tuning according to SM^2.

with stronger OMT abilities achieve better SiMT performance after fine-tuning. Besides, the results in Table 2 show that these models' original OMT abilities are not hurt, indicating that SM^2 enables models to support both OMT and SiMT abilities.

5.6 Ablation Study

We conduct ablation studies on SM^2 to analyze the effect of $\mathcal{L}_{omt}$ and modification from OMT setting.

Effect of $\mathcal{L}_{omt}$ As shown in Figure 8, 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 3 show that the OMT performance of SM^2 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 setting. Specifically, we denote t_i as the ground-truth label of the i-th target token, and hence the modification in SM^2 is adjusted as:

$$p_{ij}^m = c_{ij} \cdot p_{ij} + (1 - c_{ij}) \cdot t_i \tag{12}$$

As shown in Figure 8, the performance of SM^2 trained with modification in Eq.(12) also drops.

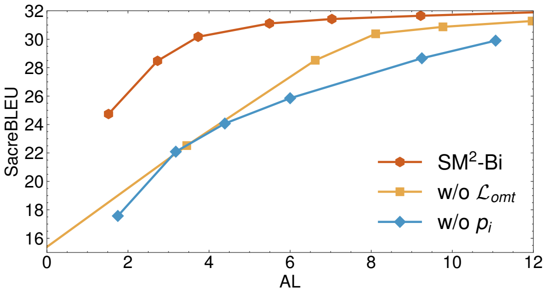


Figure 8: Effect of $\mathcal{L}_{omt}$ and modification from OMT setting on the SM^2."w/o $\mathcal{L}_{omt}$" is SM^2 trained without $\mathcal{L}_{omt}$, and "w/o p_i" means SM^2 trained using one-hot rather than OMT setting for modification.

Model	SM^2-Bi	w/o $\mathcal{L}_{omt}$	w/o p_i	OMT
SacreBLEU	31.87	30.33	31.95	31.93

Table 3: Effect of $\mathcal{L}_{omt}$ and p_i on the OMT ability.

We argue that the modification from t_i cannot reflect the real available gain from the modification after receiving the complete source sentence, thus learning a worse policy.

6 Related Work

Simultaneous Machine Translation Different from offline machine translation (Vaswani et al., 2017; Zhao et al., 2020; Wu et al., 2024), existing SiMT methods are divided into fixed policy and adaptive policy. For fixed policy, Ma et al. (2019) proposed wait-k, which starts translation after receiving k tokens. 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. (2020b) integrated multi-head monotonic attention to model the decision process, where each head independently makes decisions. Similarly, Zhang and Feng (2022a) utilized Gaussian multi-head attention to model the alignment, thus improving the decision-making ability of each head. Miao et al. (2021) proposed a generative framework to learn a read/write policy. Zhang and Feng (2022b) measured the information SiMT had received and proposed an information-based policy. Zhang and Feng (2023) used the Hidden Markov model in SiMT task to learn an adaptive policy.

Previous methods based on decision paths are limited in policy learning and model structure. Our proposed SM^2 individually explores all states during training, overcoming these limitations.

Confidence Estimation for OMT Confidence estimation is used to measure the models' credibility. Wang et al. (2019) used Monte Carlo dropout to propose an uncertainty-based confidence estimation. Wan et al. (2020) utilized the confidence score to guide self-paced learning. DeVries and Taylor (2018) evaluated the confidence by measuring the level it asks for hints from the ground-truth label, and Lu et al. (2022) transferred it to OMT to improve the out-of-distribution detection.

7 Conclusion

In this paper, we propose **S**elf-**M**odifying **S**tate **M**odeling (SM^2), a novel training paradigm for SiMT. SM^2 eschews the construction of complete decision paths during training, opting to explore all potential states individually instead. By introducing the Self-Modifying process, SM^2 independently assesses each state to precisely optimize the read/write policy without the credit assignment problem. Through Prefix Sampling, SM^2 ensures sufficient exploration of all potential states. Experimental results across three language pairs validate the superior performance of SM^2, and our analyses further confirm that SM^2 can learn a more effective read/write policy. More promisingly, SM^2 demonstrates the potential to endow OMT models with SiMT capability through fine-tuning.

Limitations

In this paper, we propose SM^2, a novel paradigm that individually optimizes decisions at each state. Although our experiments show the superiority of not building decision paths during training, there are still some parts to be further improved. For example, using a more effective way to independently assess the individual effect of each decision on the SiMT performance. Besides, how to leverage other pre-trained encoder-decoder models like BART and T5, to gain SiMT models, is still a promising direction to explore. These will be considered as objectives for our future work.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62206295 and 62206286.

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.

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.

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.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2242–2254.

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.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020a. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3505–3511.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020b. 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.

Marvin Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

Masato Neishi and Naoki Yoshinaga. 2019. On the relation between position information and sentence length in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338.

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.

Dušan Variš and Ondřej Bojar. 2021. Sequence length is a domain: Length-based overfitting in transformer models. *arXiv preprint arXiv:2109.07276*.

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*.

Junhong Wu, Yuchen Liu, and Chengqing Zong. 2024. F-malloc: Feed-forward memory allocation for continual learning in neural machine translation. *arXiv preprint arXiv:2404.04846*.

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. Gaussian multihead attention for simultaneous machine translation. *arXiv preprint arXiv:2203.09072*.

Shaolei Zhang and Yang Feng. 2022b. 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. 2022c. 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*.

Zhiyang Zhang, Yaping Zhang, Yupu Liang, Lu Xiang, Yang Zhao, Yu Zhou, and Chengqing Zong. 2023. Layoutdit: Layout-aware end-to-end document image translation with multi-step conductive decoder. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10043–10053.

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.

Yang Zhao, Lu Xiang, Junnan Zhu, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge graph enhanced neural machine translation via multitask learning on sub-entity granularity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4495–4505.

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 Comparison between SM^2 and RL-based SiMT methods

In the following, we will compare the similarities and differences between our proposed SM^2 and RL-based methods.

On the one hand, both SM^2 and RL-based methods train SiMT models to learn read/write actions at each state s_{ij}. Specifically, the read/write policy $\pi(a_{ij} \mid s_{ij})$ in SM^2 can be described as:

$$\pi(a_{ij} \mid s_{ij}) = \begin{cases} c_{ij} & a_{ij} = \text{WRITE} \\ 1 - c_{ij} & a_{ij} = \text{READ} \end{cases} \tag{13}$$

For each state s_{ij}, the reward in SM^2 can be described as:

$$r_{ij} = y_i \log(p_{ij}^m) \tag{14}$$

During training, the policy can be optimized based on the reward and converge to the optimal policy.

On the other hand, SM^2 offers additional advantages over RL-based methods. Firstly, the reward in SM^2 is differentiable, allowing the policy to be optimized by directly using the reward as the objective. In contrast, the reward in RL-based methods (Grissom II et al., 2014; Gu et al., 2017) is undifferentiable, which can hinder stable training. Secondly, SM^2 independently assesses each state, avoiding the credit assignment problem in existing RL-based methods.

B Gradient Analysis

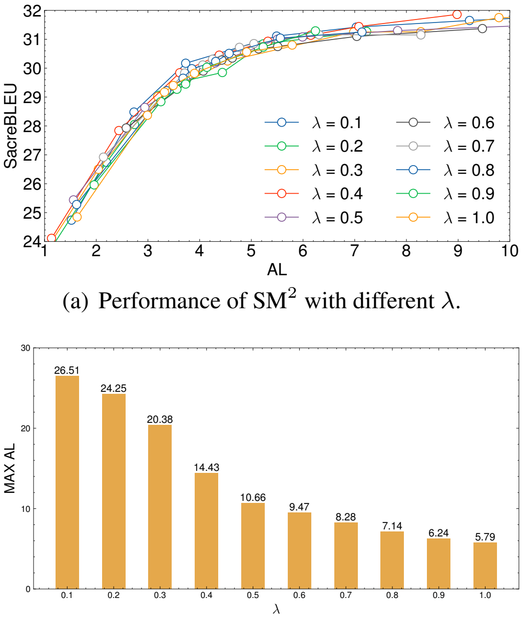
In this section, we provide a gradient analysis of the independent optimization in SM^2. The training loss function $\mathcal{L}$ of SM^2 is formulated in Eq. (10). During training, this loss function adjusts each decision d_{ij} at state s_{ij} by changing the value of corresponding confidence c_{ij}. Specifically, the gradient of $\mathcal{L}$ with respect to c_{ij} is calculated as:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial c_{ij}} &= \frac{\partial \mathcal{L}_{s_{ij}}}{\partial c_{ij}} + \lambda \frac{\partial \mathcal{L}_{c_{ij}}}{\partial c_{ij}} \\ &= -\frac{y_i}{p_{ij}^m} \cdot \frac{\partial p_{ij}^m}{\partial c_{ij}} - \frac{\lambda}{c_{ij}} \\ &= -\frac{y_i(p_{ij} - p_i)}{c_{ij} \cdot p_{ij} + (1 - c_{ij}) \cdot p_i} - \frac{\lambda}{c_{ij}} \end{aligned} \tag{15}$$

It is evident that this gradient does not contain any $c_{i'j'}$ ($i' \neq i$ or $j' \neq j$). Therefore, in the training process, the estimated value of c_{ij} is adjusted only based on its current value and the prediction probability of the current state, without being affected by the decisions at other states, thus allowing for the independent optimization of c_{ij}.

In contrast, existing SiMT methods usually conduct training on decision paths and can not ensure independent optimization. Taking ITST (Zhang and Feng, 2022b) as an example, whose loss function $\mathcal{L}'$ is formulated as:

$$\begin{gathered} \mathcal{L}' = \mathcal{L}_{ce} + \mathcal{L}_{latency} + \mathcal{L}_{norm} \\ \mathcal{L}_{latency} = \sum_{i=1}^{I} \sum_{j=1}^{J} T_{ij} \times C_{ij} \\ \mathcal{L}_{norm} = \sum_{i=1}^{I} \left\| \sum_{j=1}^{J} T_{ij} - 1 \right\|_2 \end{gathered} \tag{16}$$



(a) Performance of SM^2 with different λ.

(b) Max Latency of SM^2 with different λ.

Figure 9: Effect of λ on SM^2.

where $\mathcal{L}_{ce}$ is the cross-entropy for learning translation ability, and C_{ij} is the latency cost for each state. During training, the decision is dominated by T_{ij}. The gradient of $\mathcal{L}'$ with respect to T_{ij} is calculated as:

$$\frac{\partial \mathcal{L}'}{\partial T_{ij}} = \frac{\partial \mathcal{L}_{ce}}{\partial T_{ij}} + C_{ij} + 2(\sum_{j=1}^{J} T_{ij} - 1) \tag{17}$$

It is noted that the gradient of T_{ij} is also affected by the current values of $T_{ij'} (j' = 1, 2, ..., J)$. These decisions are coupled in the optimization, thus not enabling the independent optimization of each decision. This can trigger mutual interference during training (Zhang and Feng, 2023) and lead to a credit assignment problem.

C Effect of λ

We analyze the effect of λ, which is the weight of the penalty during training. We train SM^2 with

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 4: Hyper-parameters of our experiments.

different λ ranging from 0.1 to 1, in increments of 0.1. As shown in Figure 9(a), the SM^2 models trained with different λ show comparable performance across all latency. This indicates that SM^2 is robust to variations in hyper-parameters 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 SM^2. We denote the "MAX AL" as the latency of SM^2 when γ is set as 0.99 during inference. The results are shown in Figure 9(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 SM^2 to perform SiMT task under the latency levels where AL is larger than 5.79 since the threshold γ has been close to 1.

D Hyper-parameters

The system settings in our experiments are shown in Table 4. We set $\lambda = 0.1$ during training. Besides, we follow Ma et al. (2020b) to use greedy search during inference for all baselines. The values of γ we used are 0.3,0.4,0.5,0.55,0.6,0.65 for Zh→En, 0.3,0.4,0.5,0.55,0.6,0.65,0.7 for De→En, and 0.3,0.4,0.5,0.6,0.65,0.7,0.75 for En→Ro.

E Main Results Supplement

E.1 Numerical Results

Table 5, 6, 7 respectively report the numerical results on LDC Zh→En, WMT15 De→En, WMT16 En→Ro measured by AL, SacreBLEU and COMET. Figure 10, 11 and Table 8 report the results on WMT15 En→Vi with Transformer-small, which also present the superior performance of SM^2-Uni and SM^2-Bi.

E.2 Robustness of SM^2 to Sentence Length

To validate the robustness of SM^2 to Sentence Length, we conduct additional experiments on De→En SiMT tasks. Specifically, we divide the test set into two groups based on sentence length: LONG group and SHORT group. The average lengths and the number of sentences in each group are shown in Table 9. Then, we test SM^2-Bi and SM^2-Uni separately on these two groups. The translation quality under different latency levels for SM^2-Bi and SM^2-Uni are presented in Figure 12. For clearer comparison, we also provide the performances of OMT models (OMT-Bi, OMT-Uni) on LONG and SHORT groups.

The results in Figure 12 indicate that when applied to longer sentences, the performance changes of SM^2 are similar to OMT models in both unidirectional and bidirectional encoder settings. Since the performance of OMT models unavoidably drops as the sentences become longer (Neishi and Yoshinaga, 2019; Kang et al., 2020; Ma et al., 2020a; Variš and Bojar, 2021; Zhang et al., 2023), it is not SM^2 that triggers the decrease of translation quality. Therefore, SM^2 is still effective on long sentences.

F Effect of Prohibition on Policy

To further validate that the prohibition of exploration negatively affects the policy. We compare the SA of SM^2 with and without the prohibition on RWTH dataset. The results in Figure 13 indicate that the prohibition makes SM^2 learn a worse policy. Therefore, we can conclude that the prohibition will hurt the quality of policy. This further presents the advantage of SM^2 in sufficiently exploring all states through Prefix Sampling.

Chinese→English			
wait-k			
k	AL	SacreBLEU	COMET
1	-0.60	23.14	67.06
3	3.03	31.94	73.91
5	4.96	35.56	75.87
7	6.87	37.50	76.99
9	8.82	38.90	77.85
m-wait-k			
k	AL	SacreBLEU	COMET
1	0.72	28.06	70.85
3	2.80	32.41	74.29
5	4.76	35.05	75.81
7	6.81	36.68	76.86
9	8.64	37.61	77.37
HMT			
(L, K)	AL	SacreBLEU	COMET
(2,4)	2.93	35.59	76.90
(3,6)	4.52	37.81	78.08
(5,6)	6.11	39.41	78.73
(7,6)	7.69	40.33	79.11
(9,8)	9.64	41.37	79.58
(11,8)	11.35	41.75	79.85
ITST			
δ	AL	SacreBLEU	COMET
0.2	0.62	30.31	73.66
0.3	2.88	35.87	77.02
0.4	4.88	39.27	78.41
0.5	6.94	41.20	79.27
0.6	9.17	42.23	79.68
0.7	11.40	42.75	79.93
SM^2-Uni			
γ	AL	SacreBLEU	COMET
0.3	-0.63	29.52	73.62
0.4	1.99	36.16	77.02
0.5	4.56	39.94	78.66
0.55	6.24	41.06	79.13
0.6	8.51	42.21	79.50
0.65	9.75	42.54	79.61
SM^2-Bi			
γ	AL	SacreBLEU	COMET
0.3	-0.14	31.41	75.00
0.4	2.35	37.77	78.09
0.5	4.68	41.15	79.42
0.55	6.19	42.47	79.91
0.6	8.37	43.51	80.21
0.65	11.61	44.34	80.45

Table 5: Numerical results on LDC Zh→En.

German→English			
wait-k			
k	AL	SacreBLEU	COMET
1	0.10	20.11	70.74
3	3.44	26.34	76.24
5	6.00	28.96	78.44
7	8.08	29.52	78.92
9	9.86	30.23	79.71
m-wait-k			
k	AL	SacreBLEU	COMET
1	0.03	20.71	70.49
3	2.94	24.85	74.49
5	5.48	27.43	76.80
7	7.66	28.2	77.67
9	9.63	28.87	78.23
HMT			
(L, K)	AL	SacreBLEU	COMET
(2,4)	2.20	25.67	75.66
(3,6)	3.58	28.29	77.94
(5,6)	4.96	29.33	78.76
(7,6)	6.58	29.47	79.23
(9,8)	8.45	30.25	79.82
(11,8)	10.18	30.29	79.74
ITST			
δ	AL	SacreBLEU	COMET
0.2	2.27	25.17	75.17
0.3	2.85	26.94	76.86
0.4	3.83	28.58	77.98
0.5	5.47	29.51	78.85
0.6	7.60	30.46	79.28
0.7	10.17	30.74	79.53
0.8	12.72	30.84	79.61
SM^2-Uni			
γ	AL	SacreBLEU	COMET
0.3	1.39	24.68	75.58
0.4	2.4	27.88	78.09
0.5	3.56	29.6	79.51
0.55	5.2	30.67	80.28
0.6	6.33	30.86	80.36
0.65	8.06	30.89	80.42
0.7	10.74	31.08	80.53
SM^2-Bi			
γ	AL	SacreBLEU	COMET
0.3	1.52	24.74	75.96
0.4	2.73	28.48	78.85
0.5	3.73	30.17	80.21
0.55	5.49	31.11	80.83
0.6	7.03	31.42	81.00
0.65	9.22	31.65	81.18
0.7	12.33	31.92	81.25

Table 6: Numerical results on WMT15 De→En.

English→Romanian			
wait-k			
k	AL	SacreBLEU	COMET
1	2.70	26.62	74.12
3	5.05	29.74	77.52
5	7.18	31.61	78.54
7	9.10	31.86	79.20
9	10.92	31.89	78.97
m-wait-k			
k	AL	SacreBLEU	COMET
1	2.66	26.65	74.27
3	5.07	30.11	77.44
5	7.18	31.05	78.35
7	9.07	31.44	78.71
9	10.89	31.37	78.62
HMT			
(L, K)	AL	SacreBLEU	COMET
(1,2)	1.98	24.11	71.73
(2,2)	2.77	27.18	74.85
(4,2)	4.47	30.41	77.65
(5,4)	5.48	31.56	78.80
(6,4)	6.45	31.88	78.94
(7,6)	7.41	31.85	79.17
(9,6)	9.24	31.98	79.05
ITST			
δ	AL	SacreBLEU	COMET
0.1	2.75	22.76	71.19
0.2	3.25	28.40	75.58
0.3	5.09	30.52	77.53
0.4	7.47	31.37	78.28
0.45	8.81	31.62	78.49
0.5	10.30	31.63	78.51
0.55	11.69	31.74	78.73
SM^2-Uni			
γ	AL	SacreBLEU	COMET
0.3	2.52	27.85	75.45
0.4	2.72	29.21	76.62
0.5	3.16	30.21	77.59
0.6	4.17	31.20	78.26
0.65	5.13	31.56	78.58
0.7	6.56	31.72	78.77
0.75	8.67	31.67	78.98
SM^2-Bi			
γ	AL	SacreBLEU	COMET
0.3	2.60	28.74	76.81
0.4	2.91	30.27	78.20
0.5	3.57	31.33	79.04
0.6	5.11	32.03	79.56
0.65	6.51	32.40	79.90
0.7	8.15	32.59	79.85
0.75	10.10	32.74	79.95

Table 7: Numerical results on WMT16 En→Ro.

English→Vietnamese			
wait-k			
k	AL	SacreBLEU	COMET
1	2.49	25.29	68.89
3	4.28	28.03	70.28
5	6.07	28.73	70.56
7	7.89	28.72	70.72
9	9.57	28.78	70.75
m-wait-k			
k	AL	SacreBLEU	COMET
1	2.78	27.02	69.88
3	4.38	28.59	70.61
5	6.12	28.74	70.75
7	7.88	28.69	70.78
9	9.61	28.78	70.83
HMT			
(L, K)	AL	SacreBLEU	COMET
(1,2)	2.9	27.69	70.22
(4,2)	5.33	29.23	71.04
(5,4)	6.23	29.36	71.01
(6,4)	7.1	29.34	71.15
(7,6)	8.01	29.42	70.99
ITST			
δ	AL	SacreBLEU	COMET
0.1	3.28	28.55	70.61
0.15	4.52	29.04	70.90
0.2	5.72	29.01	70.89
0.25	8.38	29.13	70.83
0.3	9.69	29.24	70.95
SM^2-Uni			
γ	AL	SacreBLEU	COMET
0.6	1.87	28.29	70.46
0.7	3.04	28.82	70.84
0.75	5.31	29.28	71.16
0.8	6.25	29.41	71.27
0.9	9.45	29.57	71.30
SM^2-Bi			
γ	AL	SacreBLEU	COMET
0.6	2.62	28.70	70.74
0.7	4.47	29.38	71.27
0.75	5.95	29.73	71.44
0.8	7.07	29.75	71.40
0.9	8.18	29.79	71.34

Table 8: Numerical results on WMT15 En→Vi.

	LONG	SHORT
Average Sentence Length	36.95	14.07
Number of Sentences	1085	1084

Table 9: Statistics on the average sentence length and number of sentences for LONG and SHORT groups.

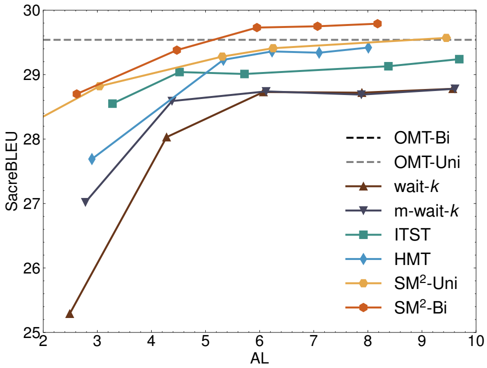


Figure 10: SacreBLEU against Average Lagging (AL) on En→Vi

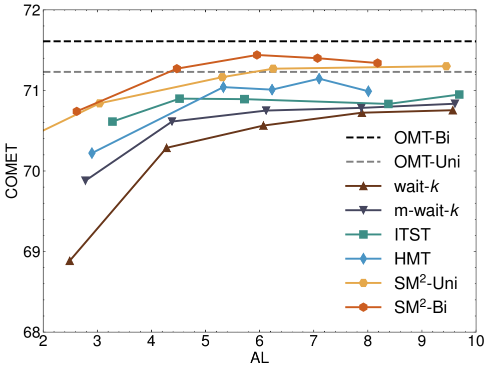
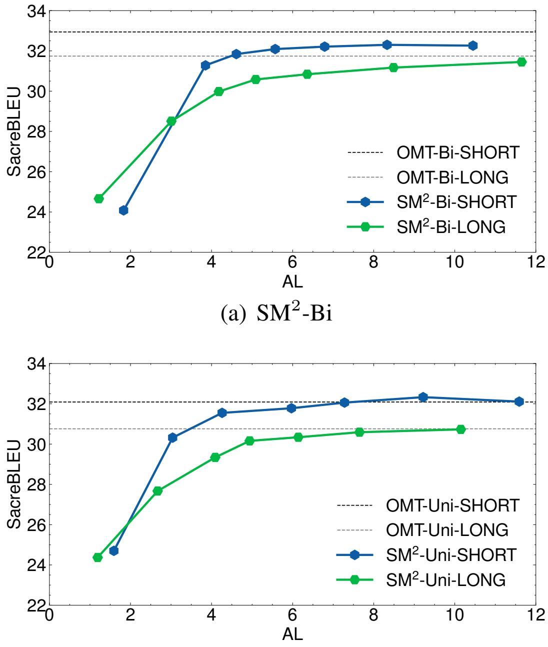


Figure 11: COMET against Average Lagging (AL) on En→Vi



(a) SM^2-Bi

(b) SM^2-Uni

Figure 12: Translation quality against latency of SM^2 on LONG and SHORT groups. We provide the performance of OMT models for a clearer comparison.

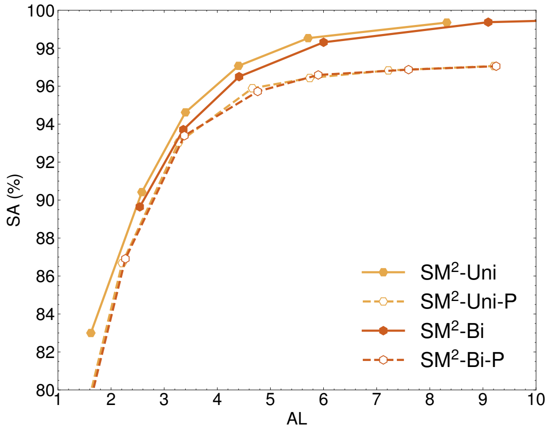


Figure 13: Evaluation of policies in SM^2 with and without prohibition. We calculate SA (↑) under different latency levels.