Gaussian Multi-head Attention for Simultaneous Machine Translation

Shaolei Zhang [1,2], **Yang Feng** [1,2*]

[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
{zhangshaolei20z, fengyang}@ict.ac.cn

Abstract

Simultaneous machine translation (SiMT) outputs translation while receiving the streaming source inputs, and hence needs a policy to determine where to start translating. The alignment between target and source words often implies the most informative source word for each target word, and hence provides the unified control over translation quality and latency, but unfortunately the existing SiMT methods do not explicitly model the alignment to perform the control. In this paper, we propose *Gaussian Multi-head Attention* (*GMA*) to develop a new SiMT policy by modeling alignment and translation in a unified manner. For SiMT policy, GMA models the aligned source position of each target word, and accordingly waits until its aligned position to start translating. To integrate the learning of alignment into the translation model, a Gaussian distribution centered on predicted aligned position is introduced as an alignment-related prior, which cooperates with translation-related soft attention to determine the final attention. Experiments on En→Vi and De→En tasks show that our method outperforms strong baselines on the trade-off between translation and latency.

1 Introduction

Simultaneous machine translation (SiMT) (Gu et al., 2017; Ma et al., 2019; Arivazhagan et al., 2019), which outputs translation before receiving the complete source sentence, is mainly used for streaming translation tasks, such as simultaneous interpretation, live broadcast and online translation. Different from full-sentence machine translation which waits for the complete source sentence, SiMT requires a policy to determine where to start translating when given the streaming inputs. The SiMT policy has to trade off between translation quality and latency and an ideal one should wait

---

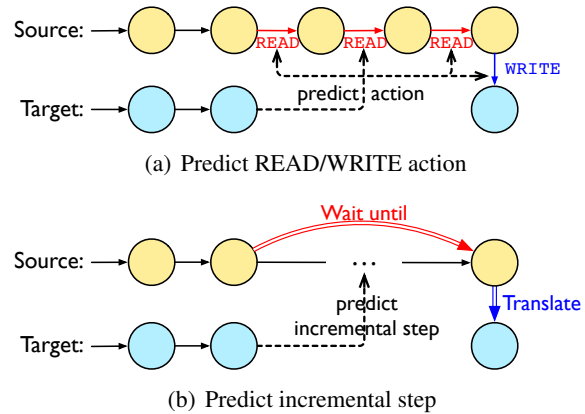(a) Predict READ/WRITE action

(b) Predict incremental step

Figure 1: Comparison diagram of previous and proposed SiMT policy. The previous policies always predict the READ/WRITE action step by step, while our method directly predicts *incremental step* (i.e., number of waiting words) between two adjacent target words.

for the right number of source words, which are sufficient but not excess, until deciding to output target words (Arivazhagan et al., 2019).

For full-sentence translation, each target word is generated based on the attended source information, where each source word provides different amount of information for the target word. Among them, the most informative source word can be considered as an aligned word to the target word (Garg et al., 2019). Then for SiMT, the alignment can be a good guider for the policy to determine where to start translating. For high translation quality, the SiMT policy is supposed to start translating after receiving the aligned source word to ensure enough source information for the translation. To consider low latency, the SiMT policy is expected not to wait for too many words after receiving the aligned source word. Therefore, if the alignment can be modeled in the SiMT model explicitly, translation quality and latency can be controlled in a unified manner for an ideal SiMT policy.

However, the existing SiMT methods, mostly employing fixed or adaptive policy, do not reflect

the alignment information in the modeling, which makes them trade off between translation quality and latency in a separate way. Fixed policy, such as wait-k policy (Ma et al., 2019), waits for a fixed number of source words and then performs READ and WRITE operations one after another, so it is a rule-based policy and precludes alignment modeling. Adaptive policy, such as MILk (Arivazhagan et al., 2019) and MMA (Ma et al., 2020), determines READ/WRITE operations by sampling from a Bernoulli distribution, as shown in Figure 1(a), where the decisions are made independently and no relationship between the decision and translation is introduced, so it has to employ an additional loss to control the latency. Besides, some methods (Wilken et al., 2020; Arthur et al., 2021) apply external ground-truth alignment as an ideal position to start translating, but the performance is inferior to MMA since separating translation and alignment.

In this paper, to explicitly involve alignments in the SiMT modeling, we propose *Gaussian Multi-head Attention* (*GMA*) to develop a SiMT policy with the guidance of alignments. To determine where to start translating with alignment, GMA first models the aligned source position of the current target word via predicting the relative distance from the previous aligned source position, called *incremental steps*, shown in Figure 1(b). Meanwhile, a relaxation offset after the aligned position is set to allow the model to wait for some additional source inputs, thereby providing a controllable trade-off between translation quality and latency in practice. Accordingly, GMA starts translating after receiving the aligned source position and waiting for the extra relaxation offset. To jointly learn alignments (i.e., SiMT policy) and translation, a Gaussian distribution centered on predicted aligned position is introduced as a prior attention over the received source words. As a result, GMA finally uses the posterior attention for translation derived from the alignment-related Gaussian prior and translation-related soft attention. Experiments on En→Vi and De→En SiMT tasks show that GMA outperforms strong baselines on the trade-off between translation quality and latency.

2 Background

GMA is applied on the multi-head attention in Transformer (Vaswani et al., 2017), so we briefly introduce SiMT and the multi-head attention.

2.1 Simultaneous Machine Translation

In a translation task, we denote the source sentence as $\mathbf{x} = \{x_1, \cdots, x_J\}$ and the corresponding source hidden states as $\mathbf{z} = \{z_1, \cdots, z_J\}$ with source length J. The model generates a target sentence $\mathbf{y} = \{y_1, \cdots, y_I\}$ and the corresponding target hidden states $\mathbf{s} = \{s_1, \cdots, s_I\}$ with target length I. Different from the full-sentence machine translation, the source words received by SiMT model are incremental and hence the model needs to decide where to output translation.

Output position Define $g(i)$ (Ma et al., 2019) as a monotonic non-decreasing function of step i, to denote the number of source words received by SiMT model when translating y_i, i.e., $g(i)$ is the output position of y_i.

In SiMT, $g(i)$ is determined by the specific policy, and the probability of generating the target word y_i is $p\left(y_i \mid \mathbf{x}_{\leq g(i)}, \mathbf{y}_{<i}\right)$, where $\mathbf{x}_{\leq g(i)}$ is first $g(i)$ source words and $\mathbf{y}_{<i}$ is previous target words. Therefore, the decoding probability of $\mathbf{y}$ is calculated as:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{|\mathbf{y}|} p\left(y_i \mid \mathbf{x}_{\leq g(i)}, \mathbf{y}_{<i}\right) \quad (1)$$

2.2 Multi-head Attention

Multi-head attention (Vaswani et al., 2017) contains multiple attention heads, where each attention head performs scaled dot-product attention. Our method is based on the cross-attention, where the queries are the target hidden states $\mathbf{s}$, the keys and values both come from the source hidden states $\mathbf{z}$. The soft attention weight α_{ij} is calculated as:

$$\alpha_{ij} = \text{Softmax}(\frac{Q(s_i)\,K(z_j)^\top}{\sqrt{d_k}}) \quad (2)$$

where $Q(\cdot)$ and $K(\cdot)$ are projection functions from the input space to the query and key space respectively, and d_k is the dimension of inputs. Then the context vector c_i is calculated as:

$$c_i = \sum_{j=1}^{J} \alpha_{ij} V(z_j) \quad (3)$$

where $V(\cdot)$ is a projection function to value space.

3 The Proposed Method

The architecture of GMA is shown in Figure 2. For SiMT policy, GMA predicts the aligned source position of the current target word, and accordingly
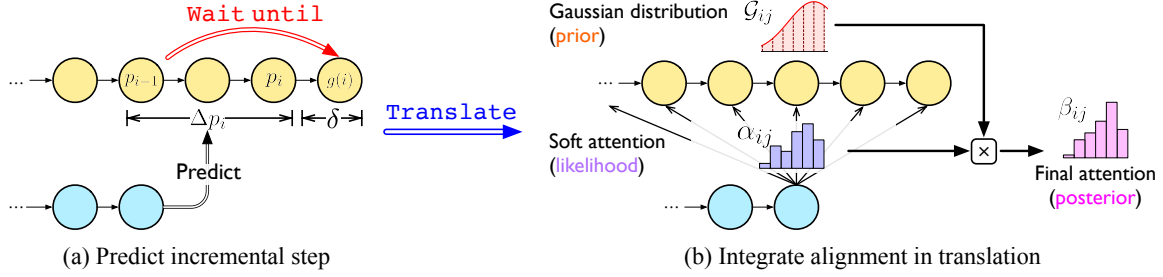
Figure 2: The architecture of GMA. (a) GMA first models the aligned source position of the current target word via predicting incremental step Δp_i, and then waits until the aligned position to start translating. (b) To integrate the learning of alignment (which determines the latency) in translation, we introduce a Gaussian distribution centered on aligned source position as alignment-related prior probability, which is multiplied with soft attention (likelihood probability) to get final attention distribution (posterior probability).

determines the output position. To integrate the learning of SiMT policy within the translation, we introduce a Gaussian prior centered on the predicted aligned position, which is multiplied with soft attention (Eq.(2)) to get final attention distribution. Due to the unimodality of Gaussian prior, it enables the model to learn the position that gets the highest soft attention (i.e., alignment), thereby developing a reasonable SiMT policy. Besides, since the Gaussian prior is continuous and differentiable, it can be integrated into the translation model directly and adjusted with the learning of translation.

3.1 Alignment-Guided SiMT Policy

Alignments prediction To develop a SiMT policy with alignments, GMA first predicts the aligned source position of the current target word. Due to the incrementality of streaming inputs in SiMT, it is unstable to directly predict the absolute position of the aligned source word. Instead, we predict the relative distance from the previous aligned source position, called *incremental step*.

Formally, we denote the aligned source position of the ith target word as $p_i \in [1, J]$ and the incremental step as $\Delta p_i \in (0, +\infty)$. Therefore, the aligned source position p_i is calculated as:

$$p_i = \begin{cases} 1 & i = 0 \\ p_{i-1} + \Delta p_i & i > 0 \end{cases} \quad (4)$$

where we set the initial aligned position p_0 to the first source word, and the incremental step Δp_i is predicted through a multi-layer perceptron (MLP) based on the previous target hidden state s_{i-1}:

$$\Delta p_i = \exp\left(\mathbf{V_p}^\top \tanh(\mathbf{W_p}\, Q(s_{i-1}))\right) \quad (5)$$

where $\mathbf{W_p}, \mathbf{V_p}$ are learnable parameters of MLP.

Algorithm 1: SiMT Policy of GMA

Input : Streaming inputs $\mathbf{x}$ (incremental),
Initial aligned position $p_0 = 1$,
$i = 1, y_0 = \langle \text{BOS} \rangle$

Output : Target outputs $\mathbf{y}$

1 **while** $y_{i-1} \neq \langle \text{EOS} \rangle$ **do**
2   calculate *incremental step* Δp_i as Eq.(5)
3      $p_i \leftarrow p_{i-1} + \Delta p_i$
4      $g(i) = \lfloor p_i + \delta \rfloor$
5   **if** $g(i) > |\mathbf{x}|$ **then**   //▷Wait
6   **Wait Until** receive $g(i)$ source words
7   **continue**
8   **else**   //▷Translate
9   **Translate** y_i with $\mathbf{x}_{\leq g(i)}$ and $\mathbf{y}_{<i}$
10   **end**
11      $i \leftarrow i + 1$
12 **end**

SiMT policy Besides the predicted aligned position p_i, we also introduce a *relaxation offset* δ to allow the model to wait for some additional source inputs, thereby providing a controllable trade-off between translation quality and latency in practice. Specifically, the output position $g(i)$ (i.e., wait for the first $g(i)$ source words and then translate the ith target word) is calculated as:

$$g(i) = \lfloor p_i + \delta \rfloor \quad (6)$$

where $\lfloor \cdot \rfloor$ is a floor operation. In our experiments, relaxation offset δ is a hyperparameter we set to obtain the translation quality under different latency. Overall, the SiMT policy is shown in Algorithm 1.

3.2 Integrating Alignment in Translation

To jointly learn the SiMT policy (i.e., aligned positions which determine latency) with translation, we

weaken the attention of source words far away from the predicted aligned position in advance, thereby forcing the model to move the predicted aligned position to the source word that is most informative for translation (i.e., with the highest soft attention).

To this end, for i^{th} target word, we introduce a Gaussian distribution $\mathcal{G}_i$ centered on aligned position p_i as the prior probability, calculated as:

$$
\mathcal{G}_{ij} = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(j-p_i)^2}{2\sigma_i^2}\right) & \text{if } j \leq g(i) \\ 0 & \text{otherwise} \end{cases} \quad (7)
$$

where the Gaussian distribution is limited in first $g(i)$ source words. σ_i is the variance used to control the attenuation degree of the prior probability as away from the aligned position. To prevent the prior probability of the furthest source word from being too small, we set $\sigma_i = p_i/2$ according to the "two-sigma rule" (Pukelsheim, 1994). We will compare the performance of different settings of prior probability in Sec.6.1. Note that since the source position is discrete, we normalize the Gaussian distribution with $\mathcal{G}_{ij}/\sum_{k=1}^{g(i)}\mathcal{G}_{ik}$.

Given the prior probability $\mathcal{G}_{ij}$ and soft attention α_{ij} (calculated as Eq.(2)), which is considered as likelihood probability, we calculate the posterior probability $\widehat{\beta}_{ij}$ and normalize it as the final attention distribution β_{ij}:

$$
\widehat{\beta}_{ij} = \alpha_{ij} \times \mathcal{G}_{ij} \quad (8)
$$

$$
\beta_{ij} = \frac{\widehat{\beta}_{ij}}{\sum_{k=1}^{g(i)}\widehat{\beta}_{ik}} \quad (9)
$$

Then, the context vector c_i is calculated as:

$$
c_i = \sum_{j=1}^{g(i)} \beta_{ij} V(z_j) \quad (10)
$$

3.3 Adaptation to Multi-head Structure

When GMA is integrated into the Transformer with L decoder layers and H attention heads per layer, if multiple heads (totally $L \times H$ heads) independently predict their alignments, some outlier[1] heads will cause unnecessary latency (Ma et al., 2020; Zhang and Feng, 2022a).

Therefore, to better adapt to multi-head attention and capture alignments, for each decoder layer, H

---

[1]Outlier heads mean that most of the heads are aligned in the front position, while some individual heads are aligned to the farther position, which requires the model to wait until the farthest aligned word is received, causing unnecessary latency.

heads in GMA jointly predict the aligned source position and share it among H heads, while the predicted alignments in each decoder layer still remain independent. Since the output position (determined by predicted alignments) in each layer may be different, the model starts translating after reaching the furthest one. We will compare the performance of different sharing settings in Sec.6.2.

4 Related Work

A reasonable policy is the key to the SiMT performance. Early policies used segmented translation (Bangalore et al., 2012; Cho and Esipova, 2016; Siahbani et al., 2018). Gu et al. (2017) used reinforcement learning to train an agent to decide read/write. Alinejad et al. (2018) added a predict operation to the agent based on Gu et al. (2017).

Recent SiMT policies fall into fixed and adaptive. For fixed policy, Dalvi et al. (2018) proposed STATIC-RW, which alternately read and write RW words after reading S words. Ma et al. (2019) proposed a wait-k policy, which translates after lagging k source words. Elbayad et al. (2020a) enhanced wait-k policy by sampling different k. Zhang et al. (2021) proposed future-guide training for wait-k policy. Han et al. (2020) applied meta-learning in wait-k. Zhang and Feng (2021a) proposed a char-level wait-k policy. Zhang and Feng (2021c) proposed mixture-of-experts wait-k policy.

For adaptive policy, Zheng et al. (2019a) trained an agent with golden READ/WRITE actions generated by rules. Zheng et al. (2019b) added a "delay" token to read source words. Arivazhagan et al. (2019) proposed MILk, using a Bernoulli variable to determine READ/WRITE. Ma et al. (2020) proposed MMA to implement MILK on Transformer. Bahar et al. (2020) and Wilken et al. (2020) used the external ground-truth alignments to train the policy. Zhang et al. (2020) proposed an adaptive segmentation policy MU for SiMT. Liu et al. (2021) proposed cross-attention augmented transducer for SiMT. Alinejad et al. (2021) introduced an full-sentence model to generate a ground-truth action sequence and accordingly train a SiMT policy. Miao et al. (2021) proposed a generative SiMT policy.

Previous methods often neglected to jointly model alignments with translation, and meanwhile introduce additional loss functions to control the latency. However, GMA jointly learns alignment and translation, and thereby controls the latency through a simple Gaussian prior probability.

(a) En→Vi, Transformer-Small (b) De→En, Transformer-Base (c) De→En, Transformer-Big
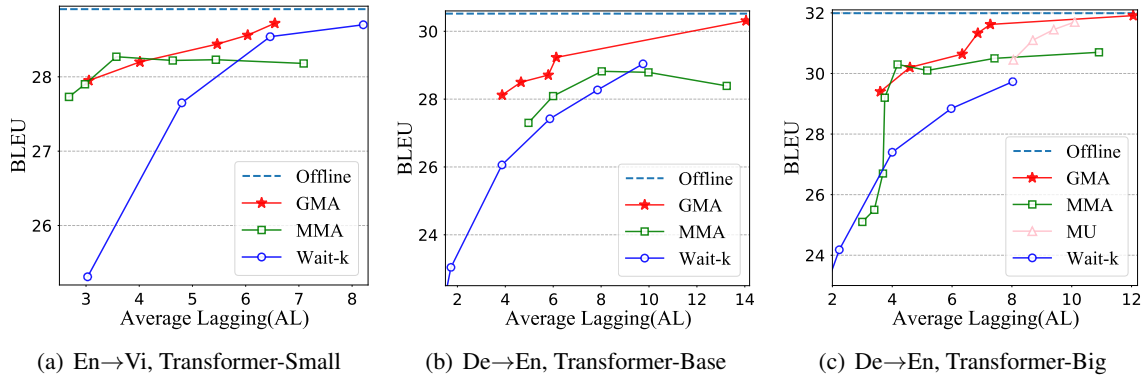
Figure 3: Translation quality (BLEU) against latency (AL) on the En→Vi(Small), De→En(Base) and De→En(Big), showing the results of GMA, Wait-k, MU, the SOTA adaptive policy MMA and offline model.

5 Experiments

5.1 Datasets

We evaluate GMA on the following public datasets.

IWSLT15[2] **English → Vietnamese (En→Vi)** (133K pairs) (Cettolo et al., 2015) We use TED tst2012 as validation set (1553 pairs) and TED tst2013 as test set (1268 pairs). Following the previous setting (Raffel et al., 2017; Ma et al., 2020), we replace tokens that the frequency less than 5 by $\langle unk \rangle$, and the vocabulary sizes are 17K and 7.7K for English and Vietnamese respectively.

WMT15[3] **German→English (De→En)** (4.5M pairs) Following Ma et al. (2019), Arivazhagan et al. (2019) and Ma et al. (2020), we use newstest2013 as validation set (3000 pairs) and newstest2015 as test set (2169 pairs). BPE (Sennrich et al., 2016) was applied with 32K merge operations and the vocabulary is shared across languages.

5.2 Systems Setting

We conduct experiments on the following systems.

Offline Conventional Transformer (Vaswani et al., 2017) model for full-sentence translation.

Wait-k Wait-k policy proposed by Ma et al. (2019), the most widely used fixed policy with strong performance and simple structure, which first waits for k source words and then translates a target word and waits for a source word alternately.

MU A segmentation policy proposed by Zhang et al. (2020), which classify whether the current inputs is a complete meaning unit (MU), and then fed MU into the full-sentence MT model for translation until generating ⟨EOS⟩. We compare our method

with MU on De→En(Big) since they report their results on De→En with Transformer-Big.

MMA[4] Monotonic multi-head attention (MMA) proposed by (Ma et al., 2020), the state-of-the-art adaptive policy. At each step, MMA predicts a Bernoulli variable to decide whether to start translating or wait for the next source token.

GMA Proposed method in Sec.3.

The implementations of all systems are adapted from Fairseq Library (Ott et al., 2019) based on Transformer (Vaswani et al., 2017). The setting is the same as Ma et al. (2020). For En→Vi, we apply Transformer-small (6 layers, 4 heads per layer). For De→En, we apply Transformer-Base (6 layers, 8 heads per layer) and Transformer-Big (6 layers, 16 heads per layer). We evaluate these systems with BLEU (Papineni et al., 2002) for translation quality and Average Lagging (AL) (Ma et al., 2019) for latency. Average lagging is currently the most widely used latency metric, which evaluates the number of words lagging behind the ideal policy. Given $g(i)$, AL is calculated as:

$$ \text{AL} = \frac{1}{\tau} \sum_{i=1}^{\tau} g(i) - \frac{i-1}{|\mathbf{y}|/|\mathbf{x}|} \qquad (11) $$

$$ \text{where} \quad \tau = \operatorname*{argmax}_i \left(g(i) = |\mathbf{x}| \right) \qquad (12) $$

where $|\mathbf{x}|$ and $|\mathbf{y}|$ are the length of the source sentence and target sentence respectively.

5.3 Main Results

We compared GMA with the Wait-k, MU and MMA, the current best representative of fixed policy, segmentation policy and adaptive policy re-

---

spectively, and plot latency-quality curves in Figure 3, where GMA curve is drawn with various δ (in Eq.(6)), Wait-k curve is drawn with various lagging numbers k, MU curve is drawn with various classification thresholds of the meaning unit, MMA curve is drawn with various latency loss weights λ.

Compared with Wait-k, GMA has a significant improvement, since Wait-k ignores the alignments and thus the target word may be forced to be translated before receiving its aligned source word, which seriously affects the translation quality. Compared with MMA and MU, our method achieves better performance under most latency levels. Since MU first segments the source sentence based on the meaning unit, and then translates each segment with the full-sentence MT model, MU performed particularly well under high latency, but meanwhile it is difficult to extend to lower latency.

Compared with the SOTA adaptive policy MMA, GMA has stable performance and simpler training method. MMA introduces two additional loss functions to control the latency, and meanwhile applies the expectation training to train Bernoulli variables (Ma et al., 2020). GMA successfully balances the translation quality and latency without any additional loss function. Owing to the proposed Gaussian prior probability centered on predicted alignments, the source words far away from the aligned position get less Gaussian prior, so that the model is forced to move the aligned position close to the most informative source word, thereby capturing the alignments and controlling the latency. With GMA, the translation quality and latency are integrated into a unified manner and jointly optimized without any additional loss function.

6 Analysis

We conduct extensive analyses to study the specific improvements of GMA. Unless otherwise specified, all the results are reported on De→En(Base).

6.1 Ablation Study

We conduct sufficient ablation studies on the method of modeling aligned position and the proposed prior probability in Table 1. For modeling aligned source position, predicting incremental step performs better. Since the complete length of the streaming inputs in SiMT is unknown, predicting absolute position easily exceeds the source length, resulting in unnecessary latency. In practice, the value range of incremental step is more regular than

Variants	AL	BLEU
Aligned Source Position		
Incremental step	**4.66**	**28.50**
Absolute position	7.33	25.61
Prior Probability		
Gaussian ($\sigma_i = p_i/2$)	**4.66**	**28.50**
$- \sigma_i = p_i/1$	6.87	28.96
$- \sigma_i = p_i/3$	4.55	27.61
$-$ Predicted σ_i	5.34	27.12
Laplace	8.14	29.19
Linear	12.83	27.86
None	1.48	20.84

Table 1: Ablation studies on modeling aligned position and the proposed prior probability, with $\delta = 1$.

absolute position and thus easier to learn.

Among the prior probabilities of different distributions, the Gaussian distribution performs best. The Laplace distribution is more fat-tailed than the Gaussian distribution (i.e., more prior probability on the position that far away from the aligned position), resulting in learning a later aligned position and higher latency. The linear distribution performs worse since its attenuation with distance is smoother. When removing the prior probability, since the parameters to predict alignments do not get the back-propagation gradient, the model cannot learn the alignments at all, resulting in very low latency and poor translation quality. Focusing on the best performing Gaussian prior probability, when $\sigma_i = p_i/1$ (the attenuation degree is small), the predicted aligned position is much later and the latency is higher. when $\sigma_i = p_i/3$, the translation quality declines since the prior of some source words far away from the aligned position is too small. When the σ_i is predicted, some small predicted σ_i will make the prior probability of the distant source words almost 0, resulting in poor translation quality. In comparison, the Gaussian prior ($\sigma_i = p_i/2$) we proposed not only guarantees a certain attenuation degree, but also assigns the furthest source word some prior probability.

6.2 Effect of Sharing Alignments

When integrated into multi-head attention, to reduce the overall latency of the model, GMA shares the predicted alignments among H heads in each layer. Table 2 reports the performance of sharing alignments in different parts (all independent, among heads, among layers or among all).

	#Predicted Alignments	AL	BLEU
All independent	$6 \times 8 = 48$	7.85	29.18
Share among heads	$6 \times 1 = 6$	4.66	28.50
Share among layers	$1 \times 8 = 8$	4.46	27.82
Share all	$1 \times 1 = 1$	3.07	27.26

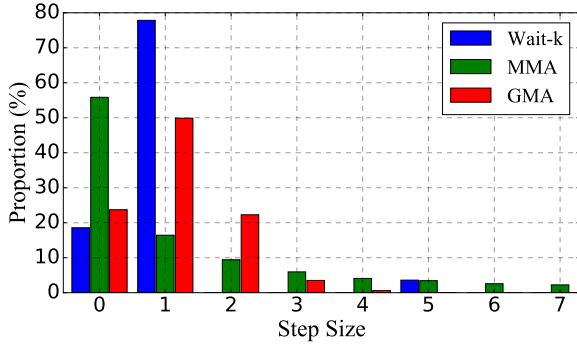Table 2: The performance of different alignments sharing settings with $\delta = 1$.

Figure 4: The distribution of the waiting source number between two adjacent outputs, on Wait-k ($k=5$), MMA ($\lambda=0.4$) and GMA ($\delta=1$) with similar latency.

'All independent' achieves the best translation quality and also brings a higher latency, as the overall latency of the model is determined by the farthest one among all predicted positions. Besides, 'Share all' gets the lowest latency but loses translation quality. Comparing 'Share among heads' and 'Share among layers', sharing among heads performs better, which is in line with the previous conclusion that there are obvious differences between the alignments in each decoder layer (Garg et al., 2019; Voita et al., 2019; Li et al., 2019).

6.3 Statistics of Incremental Steps

Unlike MMA predicting the READ/WRITE action, GMA directly predicts the incremental step between adjacent target outputs. To analyze the advantages of modeling the incremental step, we show the distribution of step size (i.e., the number of waiting source words between two adjacent outputs) in Figure 4, where we select the SiMT models with the similar latency (AL $\approx$ 4.5).

For Wait-k, the step size is blunt and there are only three cases, which occur before the start of translation (step size$= k$: first lag k words), during translation (step size$= 1$: wait and output one word alternately.), and after the end of the source inputs (step size$= 0$: output the translation at one time). The proposed GMA and MMA have obvi-
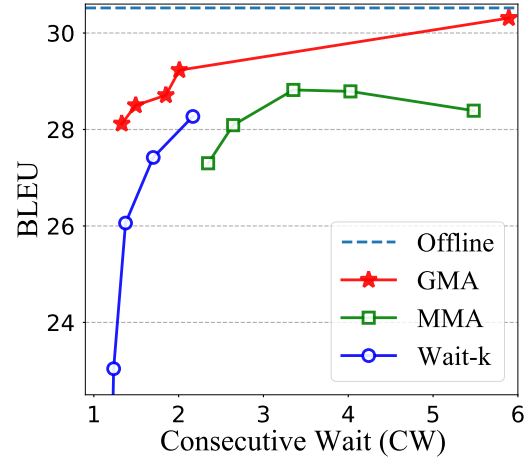

Figure 5: Translation quality against latency (CW), where CW reflects the streaming degree of SiMT.

ous differences in the step size distribution. The step size distribution of GMA is more even, only distributed between 0$\sim$4, which shows that each source segment is shorter and thus the translating process is more streaming. The step size of MMA is more widely distributed, most of which are 0 (consecutively output target words), and there is also a large proportion of step size greater than 5, which shows that MMA tends to consecutively wait for more source words and then output more target words, resulting in longer source segments. Therefore, although GMA and MMA have similar latency (AL), their performance in real applications is different, where the translation process of GMA is more streaming, while MMA is more segmented.

Furthermore, to more accurately evaluate the streaming degree of the translation process, we apply Consecutive Wait (CW) (Gu et al., 2017) as the latency metric to evaluate the systems. Consecutive wait evaluates the number of source words waited between two target words, which reflects the streaming degree of SiMT. Given $g(i)$, CW is calculated as:

$$\text{CW} = \frac{\sum_{i=1}^{|\mathbf{y}|}(g(i) - g(i-1))}{\sum_{i=1}^{|\mathbf{y}|} \mathbb{1}_{g(i)-g(i-1)>0}} \quad (13)$$

where $\mathbb{1}_{g(i)-g(i-1)>0} = 1$ counts the number of $g(i) - g(i-1) > 0$. In other words, CW measures the average source segment length (the best case is 1 for word-by-word streaming translation and the worst case is $|\mathbf{x}|$ for full-sentence MT), where the smaller the CW, the shorter the average source segment, and the translation is more streaming.

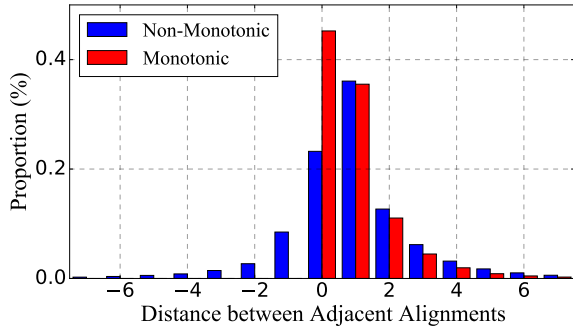As shown in Figure 5, GMA gets much smaller CW scores than MMA, where the average source

Figure 6: Distribution of distances between adjacent alignments (ground-truth) under monotonic and non-monotonic alignment setting.

segment length is about 2 words, which shows that GMA achieves more streaming translation than the previous adaptive policy.

Additionally, although the alignments between the two languages is not necessarily monotonic, we require the predicted incremental step $\Delta p_i \geq 0$ to guarantee the model performs READ/WRITE monotonically. This is due to two considerations. First, we don't want to waste any useful source content, i.e., to avoid the model moving p_i to the previous position and thereby ignoring some received source information caused by $\Delta p_i < 0$. Second, we argue that monotonic alignments are more friendly to SiMT learning. We use `fast-align`[5] (Dyer et al., 2013) to generate the ground-truth aligned source position of ith target token, denoted as A_i, and then show the distribution of distances between adjacent alignments under monotonic and non-monotonic alignments in Figure 6, where 'Non-Monotonic' measures $A_i - A_{i-1}$ and 'Monotonic' measures $\max(A_i - \max_{j<i} A_j, 0)$. The distance distribution between adjacent alignments under monotonic alignment is more concentrated, between 0 and 4, which is easier for the model to learn incremental step. Actually, the incremental steps predicted by GMA almost distribute between 0 and 4 (see Figure 4), which shows that GMA successfully learns the relative distance between monotonic alignments.

6.4 Quality of Predicted Alignments

To evaluate the quality of the aligned source position p_i predicted by GMA, we measure the alignment accuracy on the RWTH De→En alignment dataset [6], whose reference alignment was manually

Latency	AER	% of ground-truth alignments within $g(i)$
Low	0.49	81.00%
Mid	0.61	88.27%
High	0.76	95.58%

Table 3: Alignment quality under different latency levels, where 'within $g(i)$' means that starting translating after receiving the aligned source word.
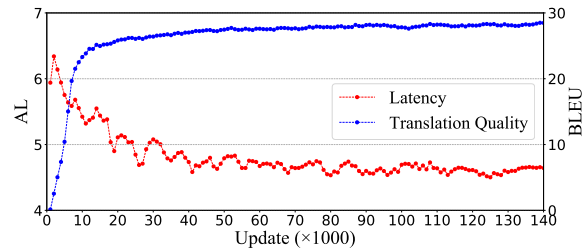

Figure 7: The learning curve of translation quality (BLEU) and latency (AL) during training, with $\delta = 1$.

annotated by experts (Liu et al., 2016; Zhang and Feng, 2021b). As shown in Table 3, we sample one decoder layer to calculate the alignments error rate (AER) (Vilar et al., 2006), and meanwhile count how many ground-truth aligned source words are located before the output position $g(i)$ (i.e., translate after receiving the aligned source word).

GMA achieves good alignment accuracy, especially at low latency, since the model is required to output immediately after receiving the aligned source words (δ in Eq.(6) is small). More importantly, most of the ground-truth alignments is within $g(i)$, showing that GMA guarantees that in most cases, the model starts translating a target word after receiving its aligned source words, which is beneficial to translation quality.

6.5 Balancing Translation and Latency

To study how GMA learns to balance translation quality and latency without any additional loss function during training, we draw a learning curve for translation quality and latency in Figure 7.

Initially, the high latency indicates that the model first moves the predicted aligned position to a further position, to learn the translation by seeing more source words. Then, as the number of training steps increases, the translation quality improves and the latency gradually decreases, which shows that for better translation, the model moves the predicted aligned source position to a more appropriate position due to the introduced Gaussian prior
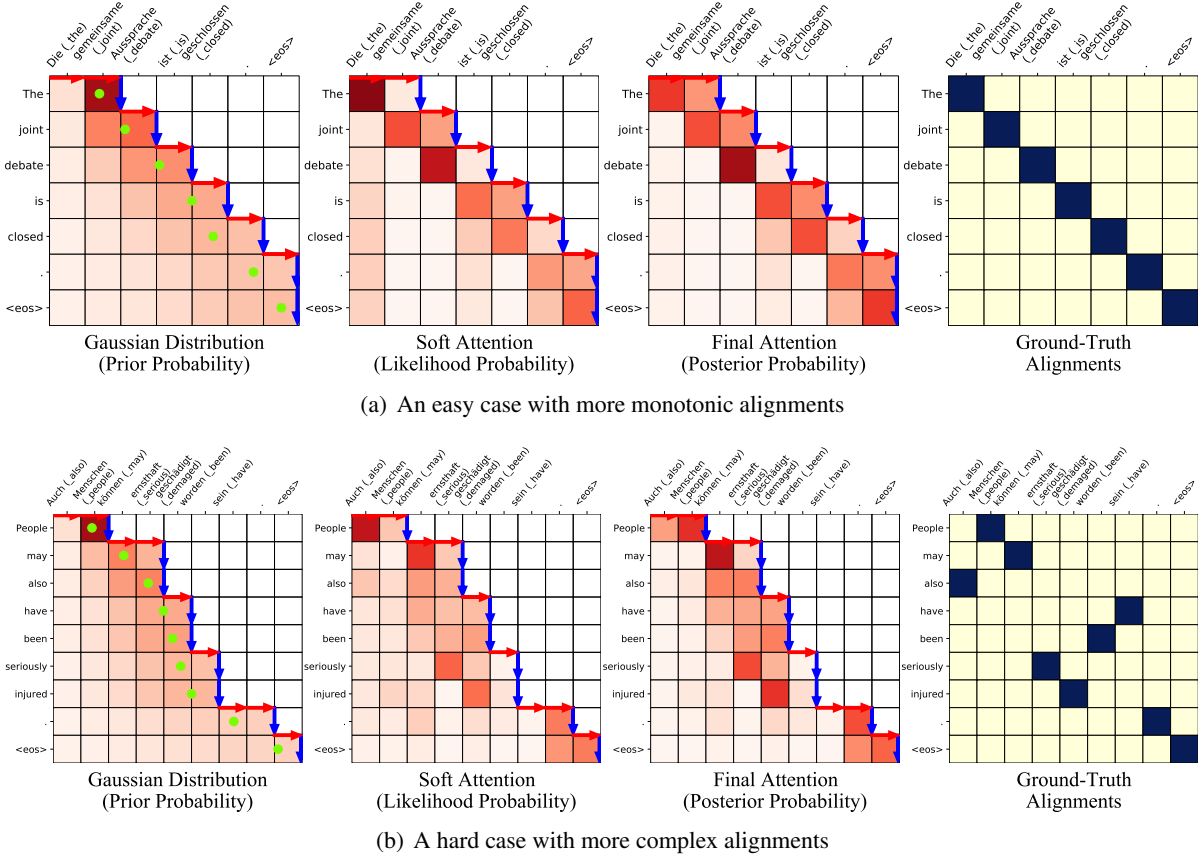
3026

(a) An easy case with more monotonic alignments

(b) A hard case with more complex alignments

Figure 8: Attention visualization of GMA on De→En SiMT with $\delta = 1$. The shade of the color indicates the attention weight. '●': the predicted aligned source position p_i (mean of Gaussian distribution), note that p_i is a float number. '■': the ground-truth alignments. '→': wait for a source word, '↓': translate a target word.

probability. Overall, for better translation, GMA constantly adjusts the predicted aligned source position to a suitable position and thereby controls the latency, which is completely different from the previous method of introducing the additional latency loss to constrain the latency with translation (Ma et al., 2020; Miao et al., 2021).

6.6 Characteristics of Attention in GMA

We explore the characteristics of GMA by visualizing the attention distributions in Figure 8. We show two cases with the alignments of different difficulty levels, where the reverse orders in alignments are considered as a major challenge for SiMT (Ma et al., 2019; Zhang and Feng, 2021c).

For more monotonic alignments, GMA can predict the aligned source position well and output the target word after receiving the aligned word. Meanwhile, due to the characteristics of Gaussian distribution, GMA can also avoid focusing too much on source words in the front position, and strengthen the attention on the newly received source words, which is proved to be beneficial to SiMT perfor-

mance (Elbayad et al., 2020b; Zhang and Feng, 2022b). For more complex alignments, the aligned position predicted by GMA is close to the ground-truth alignments, so that GMA starts translating after receiving most of aligned words. Besides, GMA learns some implicit prediction ability, e.g., before receiving "*worden sein*", GMA generates the correct translation "*have been*" based on the context. We consider this is because the predicted alignments during training are monotonic due to the incremental step, where modeling monotonic alignments forces the model to learn the correct translation from the incomplete source and previous outputs (Ma et al., 2019).

7 Conclusion

In this paper, we propose the Gaussian multi-head attention to develop a SiMT policy which starts translating a target word after receiving its aligned source word. Experiments and analyses show that our method achieves promising results on performance, alignments quality and streaming degree.

Acknowledgements

References

Ashkan Alinejad, Hassan S. Shavarani, and Anoop Sarkar. 2021. Translation-based supervision for policy generation in simultaneous neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1734–1744, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

Philip Arthur, Trevor Cohn, and Gholamreza Haffari. 2021. Learning coupled policies for simultaneous machine translation using imitation learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2709–2719, Online. Association for Computational Linguistics.

Parnia Bahar, Patrick Wilken, Tamer Alkhouli, Andreas Guta, Pavel Golik, Evgeny Matusov, and Christian Herold. 2020. Start-before-end and end-to-end: Neural speech translation by AppTek and RWTH Aachen University. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 44–54, Online. Association for Computational Linguistics.

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 437–445, Montréal, Canada. Association for Computational Linguistics.

Mauro Cettolo, Niehues Jan, Stüker Sebastian, Luisa Bentivogli, R. Cattoni, and Marcello Federico. 2015. The iwslt 2015 evaluation campaign.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation?

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

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020a. Efficient Wait-k Models for Simultaneous Machine Translation.

Maha Elbayad, Michael Ustaszewski, Emmanuelle Esperança-Rodier, Francis Brunet-Manquat, Jakob Verbeek, and Laurent Besacier. 2020b. Online versus offline NMT quality: An in-depth analysis on English-German and German-English. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5047–5058, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

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

Hou Jeung Han, Mohd Abbas Zaidi, Sathish Reddy Indurthi, Nikhil Kumar Lakumarapu, Beomseok Lee, and Sangha Kim. 2020. End-to-end simultaneous translation system for IWSLT2020 using modality agnostic meta-learning. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 62–68, Online. Association for Computational Linguistics.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. 2021. Cross attention augmented transducer networks for simultaneous translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 39–55, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

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

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Friedrich Pukelsheim. 1994. The three sigma rule. *The American Statistician*, 48(2):88–91.

Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2837–2846. PMLR.

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, Berlin, Germany. Association for Computational Linguistics.

Maryam Siahbani, Hassan Shavarani, Ashkan Alinejad, and Anoop Sarkar. 2018. Simultaneous translation using optimized segmentation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 154–167, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

David Vilar, Maja Popovic, and Hermann Ney. 2006. AER: do we need to "improve" our alignments? In *Proceedings of the Third International Workshop on Spoken Language Translation: Papers*, Kyoto, Japan.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Patrick Wilken, Tamer Alkhouli, Evgeny Matusov, and Pavel Golik. 2020. Neural simultaneous speech translation using alignment-based chunking. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 237–246, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2021a. ICT's system for AutoSimTrans 2021: Robust char-level simultaneous translation. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*, pages 1–11, Online. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2021b. Modeling concentrated cross-attention for neural machine translation with Gaussian mixture model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1401–1411, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2021c. 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022a. Modeling dual read/write paths for simultaneous machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022b. Reducing position bias in simultaneous machine translation with length-aware framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang, Yang Feng, and Liangyou Li. 2021. Future-guided incremental transformer for simultaneous translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14428–14436, Online.

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

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy. Association for Computational Linguistics.

A   Numerical Results

We additionally use Consecutive Wait (CW) (Gu et al., 2017), Average Proportion (AP) (Cho and Esipova, 2016), Average Lagging (AL) (Ma et al., 2019), and Differentiable Average Lagging (DAL) (Arivazhagan et al., 2019) to evaluate the latency of GMA, and the numerical results are shown in Table 4, 5 and 6.

	IWSLT15 En→Vi (Small)				
δ	CW	AP	AL	DAL	BLEU
0.9	1.20	0.65	3.05	4.08	27.95
1.0	1.27	0.68	4.01	4.77	28.20
2.0	1.49	0.74	5.47	6.37	28.44
2.2	1.60	0.77	6.04	6.96	28.56
2.5	1.74	0.78	6.55	7.55	28.72

Table 4: Numerical results of GMA on IWSLT15 En→Vi (Small).

	WMT15 De→En (Base)				
δ	CW	AP	AL	DAL	BLEU
0.9	1.33	0.64	3.87	4.61	28.12
1.0	1.49	0.67	4.66	5.56	28.50
2.0	1.85	0.72	5.79	7.75	28.71
2.2	2.01	0.73	6.13	8.43	29.23
2.4	5.89	0.96	14.05	25.76	31.31

Table 5: Numerical results of GMA on WMT15 De→En (Base).

	WMT15 De→En (Big)				
δ	CW	AP	AL	DAL	BLEU
1.0	1.54	0.68	4.60	5.89	30.20
2.0	1.98	0.74	6.34	8.18	30.64
2.2	2.13	0.75	6.86	8.91	31.33
2.4	2.28	0.76	7.28	9.59	31.62
2.5	3.10	0.88	12.06	20.43	31.91

Table 6: Numerical results of GMA on WMT15 De→En (Big).