Adaptive Simultaneous Sign Language Translation with Confident Translation Length Estimation

Tong Sun[1,2*], Biao Fu[1,2*], Cong Hu[1,2], Liang Zhang[1,2], Ruiquan Zhang[1,2], Xiaodong Shi[1,2], Jinsong Su[1,2], Yidong Chen[1,2†]
[1]School of Informatics, Xiamen University, China
[2]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
tongsun@stu.xmu.edu.cn, ydchen@xmu.edu.cn

Abstract

Traditional non-simultaneous Sign Language Translation (SLT) methods, while effective for pre-recorded videos, face challenges in real-time scenarios due to inherent inference delays. The emerging field of simultaneous SLT aims to address this issue by progressively translating incrementally received sign videos. However, the sole existing work in simultaneous SLT adopts a fixed gloss-based policy, which suffers from limitations in boundary prediction and contextual comprehension. In this paper, we delve deeper into this area and propose an adaptive policy for simultaneous SLT. Our approach introduces the concept of "confident translation length", denoting the maximum accurate translation achievable from current input. An estimator measures this length for streaming sign video, enabling the model to make informed decisions on whether to wait for more input or proceed with translation. To train the estimator, we construct training data of confident translation length based on the longest common prefix between translations of partial and complete inputs. Furthermore, we incorporate adaptive training, utilizing pseudo prefix pairs to refine the offline translation model for optimal performance in simultaneous scenarios. Experimental results on PHOENIX2014T and CSL-Daily demonstrate the superiority of our adaptive policy over existing methods, particularly excelling in situations requiring extremely low latency.

Keywords: Sign Language Translation, Simultaneous Translation, Real-time Translation

1. Introduction

Sign language is a type of visual language that conveys meaning through gestures and employed by deaf and hard-of-hearing people to communicate in everyday life (Yin et al., 2021b). Sign language translation (SLT) (Camgoz et al., 2018), which aims to convert a sign language video into its corresponding natural sentence, can bridge the communication gap between the deaf and the hearing and therefore received widespread attention in recent years (Camgoz et al., 2020b; Zhou et al., 2021a; Chen et al., 2022a,b; Zhang et al., 2022a; Fu et al., 2023b; Yu et al., 2023; Gan et al., 2023).

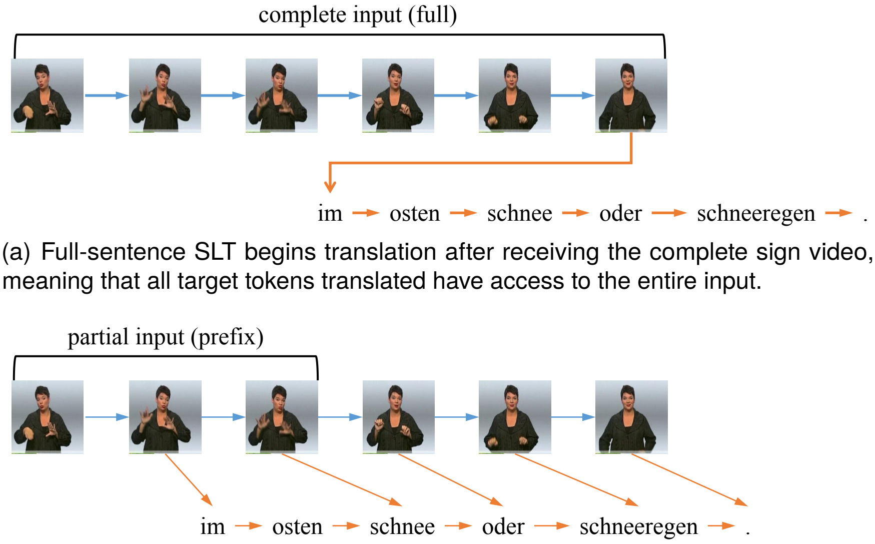
Currently, research efforts in the domain of SLT have been directed towards non-simultaneous translation methods (Camgoz et al., 2020b; Zhou et al., 2021a; Chen et al., 2022a; Zhang et al., 2022a), commonly referred to as full-sentence SLT. In this approach, model needs to wait for the complete input of sign video before translation can take place. While this method may work well for pre-recorded video, it poses challenges in real-time communication environments, especially in low-latency scenarios(Yin et al., 2021a). The inherent inference delays of the full-sentence approach make it suboptimal for such applications.

Simultaneous SLT, as an alternative, differs from its full-sentence counterpart in that it gradually generates a translation of an incrementally received input (Figure 1). This approach requires a well-designed policy that allows the model to decide whether to wait for more video input (i.e. READ) or to continue translating (i.e. WRITE), thus achieving a balance between translation quality and latency (Gu et al., 2017). Despite its potential, simultaneous SLT is also more challenging, resulting in limited research on this front. Yin et al. (2021a) pioneered research in this task and proposed a fixed gloss-based policy that first divides the sign video into segments corresponding to gloss[1] by a boundary predictor and then generates a target token every time a new gloss is detected. Nonetheless, this approach exhibits three limitations: 1) Its translation quality primarily relies on the performance of the boundary predictor. However, the predictor is optimized from a weak supervision signal (the total length of the gloss sequence) and is not guaranteed to align each gloss with the detected boundary during streaming inference; 2) The dependence of gloss annotations for training the boundary predictor restricts its applicability in more realistic scenarios due to the labor-intensive and

* Equal contribution.
† Corresponding author.

[1] Glosses represent written descriptions of signs

LREC-COLING 2024, pages 372–384
20-25 May, 2024.


complete input (full)
im → osten → schnee → oder → schneeregen → .
partial input (prefix)
im → osten → schnee → oder → schneeregen → .

(a) Full-sentence SLT begins translation after receiving the complete sign video, meaning that all target tokens translated have access to the entire input.

(b) Simultaneous SLT begins translation before receiving the complete sign video, with target tokens having access only to the input prefix as they are being generated.

Figure 1: Differences between full-sentence SLT and simultaneous SLT. The translation of the target sentence in the example is "In the east, snow or sleet."

time-consuming nature of annotating glosses. 3) The detection of one new gloss is not an appropriate translation timing for one new target token, since it lacks adequate contextual consideration.

In this paper, we propose an adaptive policy for simultaneous sign language translation. Our policy is guided by the concept of "confident translation length", which denotes the maximum number of target-side tokens that can be accurately translated from the current input. Specifically, we introduce an estimator to measure the confident translation length for the streaming sign video input. If the confident length is greater than the length of the translation history, the model makes a write decision, otherwise, the model continues to wait for more video frames. To train the estimator, we construct training data of confident translation length by calculating the length of the longest common prefix between translations of partial and complete inputs. Our policy is more reasonable than simply detecting boundaries for individual gloss segments in the translation because it focuses more on the content of the sign video and aims to convey as much translation information as possible within a limited time. Furthermore, we conduct adaptive training for the translation model by generating pseudo prefix pairs. The goal is to adapt our original offline translation model to simultaneous scenarios.

We conduct experiments on PHOENIX2014T and CSL-Daily datasets. Experimental results demonstrate that our method outperforms strong baselines, especially in scenarios with extremely low latency. Moreover, our policy can be applied to more SLT datasets without gloss annotations since it does not rely on additional gloss information.

Given that *SimulSLT* (Yin et al., 2021a) has yet to release implementation of the latency measurement, the evaluation toolkit, SimulEval (Ma et al., 2020a), traditionally utilized for simultaneous translation, remains exclusively suited for text and speech. In order to provide researchers with a standard and unified evaluation method for simultaneous SLT systems, we augment the SimulEval toolkit to extend its applicability to video inputs (called SimulEval-SLT[2]), which will facilitate future research in simultaneous SLT.

2. Background and Related Work

SLT systems can be divided into two types: non-simultaneous and simultaneous. In terms of inference mode, the former emphasizes accuracy, while the latter focuses on low latency and fluency. Nevertheless, SLT models typically employ the encoder-decoder architecture and are trained on a corpus $D = \{\mathbf{x}, \mathbf{y}\}$, where $\mathbf{x} = \{x_1, ..., x_t, ..., x_T\}$ and $\mathbf{y} = \{y_1, ..., y_j, ..., y_J\}$ denote a sign video and its corresponding target translation, respectively.

Sign Language Translation. Camgoz et al. (2018) first formulated SLT as a neural machine translation (NMT) problem. Most existing methods (Camgoz et al., 2020b; Zhou et al., 2021a; Chen et al., 2022a; Zhang et al., 2022a) are non-simultaneous and thus require an entire sign video input. They mostly adopt an autoregressive mechanism and the decoding process is defined as:

[2] Codes of our method and SimulEval-SLT are available at `https://github.com/tongsun99/CTL`

$$p(\mathbf{y} \mid \mathbf{x})=\prod_{j=1}^{J} p\left(y_j \mid \mathbf{x}, \mathbf{y}_{<j}\right) \tag{1}$$

New techniques and information have been introduced in SLT recently to further improve the translation performance of the systems, including multi-task learning (Camgoz et al., 2020a; Zhang et al., 2022a), transfer learning (Chen et al., 2022a; Hu et al., 2023), contrastive learning (Fu et al., 2023b; Gan et al., 2023), data augmentation (Zhou et al., 2021a), multi-cue fusion (Camgoz et al., 2020a; Zhou et al., 2021b; Chen et al., 2022b), non-autoregressive decoder (Yu et al., 2023), signer-independent settings (Jin and Zhao, 2021; Jin et al., 2022) and variational autoencoder (Zhao et al., 2024).

Simultaneous Translation. A simultaneous translation system generates the j-th target token based on streaming input prefix $\mathbf{x}_{\leq g(j)}$ and the previous tokens $\mathbf{y}_{<j}$, where $g(j)$ is a monotonic non-decreasing function based on a READ/WRITE policy. The decoding probability is calculated as:

$$p(\mathbf{y} \mid \mathbf{x})=\prod_{j=1}^{J} p\left(y_j \mid \mathbf{x}_{\leq g(j)}, \mathbf{y}_{<j}\right) \tag{2}$$

Existing policies can be broadly classified into two categories: **Fixed** and **Adaptive**. **Fixed** policy generates translation based on predefined rules and the most representative method is *Wait-k* (Ma et al., 2019a), which first awaits k segments and then starts to alternate between writing a token and waiting for a new segment. A segment can be defined as a character (Zhang and Feng, 2021a) or word/subword (Ma et al., 2019a) in text, whereas in speech, it might be based on fixed length (Ma et al., 2020b), Connection Temporal Classification (Ren et al., 2020; Zeng et al., 2021), ASR outputs (Chen et al., 2021) and Integrate-and-Fire (Dong et al., 2022). In simultaneous SLT, the sole existing work (Yin et al., 2021a) considers a segment as a gloss. Based on its simplicity, recent efforts have employed various methods to enhance *Wait-k*, such as mixture-of-experts (Zhang and Feng, 2021b), multi-path training (Elbayad et al., 2020), and future information utilization (Zhang et al., 2021; Zhang and Feng, 2022c; Fu et al., 2023a). However, no matter what, fixed policies are always limited by their inherent inability to adjust according to complex inputs. **Adaptive** policy, in contrast, dynamically translates based on the current situation through learned decision modules. Previous works in this category design policies based on attention (Arivazhagan et al., 2019; Ma et al., 2019b; Liu et al., 2021; Zhang and Feng, 2022a; Papi et al., 2023), information modeling (Zhang et al., 2022c; Zhang and Feng, 2022b), externally trained decision models (Zhang et al., 2020, 2022b; Guo et al., 2023), etc.

We are the first to attempt an adaptive policy in simultaneous SLT and the translation timing is determined by an external learned length estimator. Although our approach shares similarities with *MU-ST* (Zhang et al., 2022b), there are significant differences: 1) Our approach attempts to learn the translation paths derived from the longest common prefix algorithm. In theory, this is much faster than the translation paths learned by *MU-ST*, as *MU-ST* employs a strict prefix-matching algorithm; 2) Our decision model comprehends the content of the current sign video by modeling based on target length. Compared to a simple binary classification (clear or not clear), it offers a more detailed and comprehensive understanding of the video, which should lead to better performance.

3. Methodology

In this section, we will provide a detailed explanation of our newly proposed policy for simultaneous sign language translation. Our approach determines READ/WRITE by estimating the confident translation length. The overall framework of our policy is illustrated in Figure 2. Given a streaming sign video $\mathbf{x}$, we incrementally estimate the confident translation length of current input $\mathbf{x}_{\leq t}(t=1,2,\ldots)$ by a trained network, where $\mathbf{x}_{\leq t}$ represents the first tF frames of the video and F is the detection interval. We compare the predicted length l with the length of the translation history l_p at each iteration t. If l is not greater than l_p, the model chooses to wait for the next F frames to be read. Otherwise, it proceeds with translation and writes until the length of translation $\mathbf{y}$ is equal to l or the end-of-sentence (EOS) token is emitted, with the translation history $\mathbf{y}_p$ force decoded as a translation prefix.

In the following, we will first introduce the architecture and training of the translation model (Section 3.1). Then, we will explain the concept of confident translation length in our policy and demonstrate how to construct corresponding training data (Section 3.2). After that, we will show how we utilize the generated data to train a confident translation length estimator (Section 3.3). Finally, we will give a detailed description of our adaptive training with pseudo prefix pairs. (Section 3.4).

3.1. Translation Model Architecture

Many previous works (Ma et al., 2019a; Ren et al., 2020; Liu et al., 2021; Yin et al., 2021a) addressed diverse latency requirements by training multiple streaming models, which are effective yet often computationally expensive. Recent researches suggest that a single offline model can also ef-

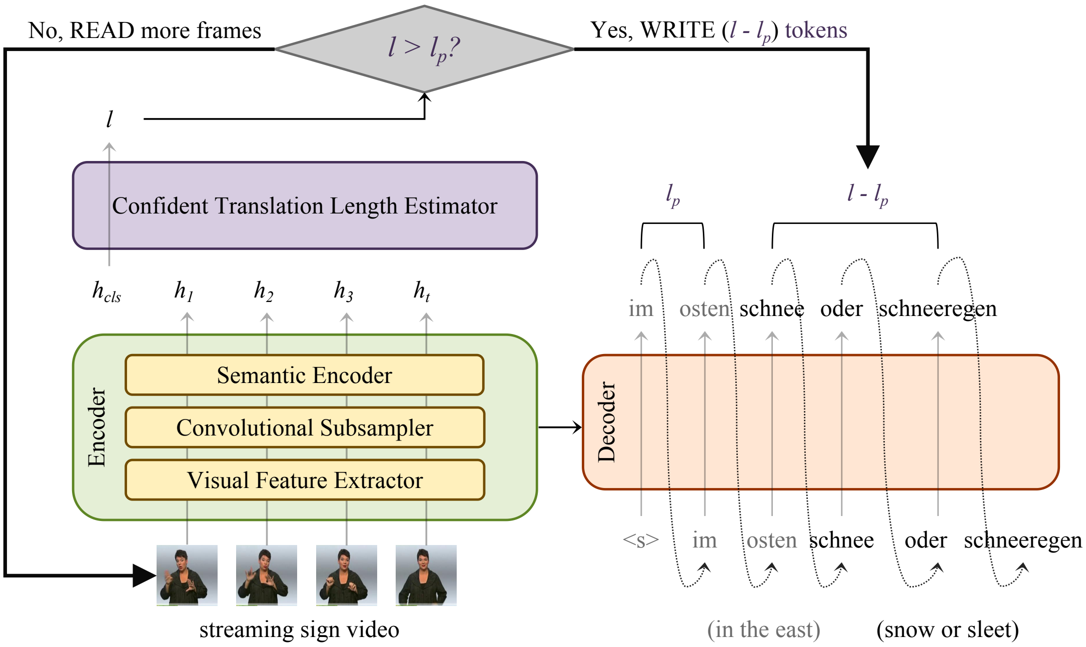


Figure 2: The overall framework of our policy. The estimator is used to measure the confidence translation length for streaming sign video. If this length exceeds the length of the translation history, a write action is triggered to emit $(l - l_p)$ tokens; otherwise, the model retains its waiting status for additional video frames.

fectively meet the demands of streaming scenarios (Papi et al., 2022; Fu et al., 2023a). Aligned with this perspective, our approach also involves training only one offline SLT model, thereby significantly reducing training costs compared to Yin et al. (2021a). The architecture consists of an encoder and a decoder, where the encoder comprises a visual feature extractor, a convolutional subsampler, and a semantic encoder, as illustrated in Figure 2. We first pass the input sign video $\mathbf{x}$ through a pre-trained CNN-based visual feature extractor (Camgoz et al., 2020b) to obtain time-agnostic visual features. Then, the 1D convolutional subsampling layer helps the model effectively capture short-term information while reducing the computational complexity. Finally, the semantic encoder consists of multiple Transformer (Vaswani et al., 2017) encoder layers, capturing long-term information in the visual features.

The translation decoder generates each token in an autoregressive way based on the previously generated tokens as well as the source sign video representations. The translation loss is defined as:

$$\mathcal{L}_{SLT} = -\sum_{j=1}^{J} \log p\left(y_j \mid \mathbf{x}, \mathbf{y}_{<j}\right) \tag{3}$$

3.2. Constructing Confident Translation Length Training Data

The concept of confident translation length is motivated by the translation strategy of simultaneous interpreters. Specifically, when interpreters have a deep and accurate understanding of current input during simultaneous translation and are confident in the translation result, they may tend to provide a more detailed and comprehensive translation. In such cases, the translation length is relatively long. Conversely, if the interpreter has a lower level of understanding of the current input or lacks confidence in the translation result, they may choose to simplify the translation, retaining only the most crucial information, resulting in a shorter translation length. Therefore, we denote the confident translation length as the maximum number of target-side tokens that can be accurately translated from the current input. It highlights the correlation between understanding the source-side input and expressing the target-side translation and reflects the interpreter's confidence level during translation.

Given the absence of standard corpora for confident translation length and its inherent nature, we propose a straightforward method to generate pseudo data. The method measures the translation confidence of our model through the Longest Common Prefix (LCP) algorithm, allowing us to train our confident translation length estimator.

The whole process is described in Algorithm 1. The algorithm iteratively reads in a fixed number of F frames until the end of the video (Line 3). We apply an offline-trained SLT model M_{slt}, as described in Section 3.1, to translate the current input $\mathbf{x}_{\leq t}$ (Line 4). Then, we extract the longest common prefix between the translation result $\mathbf{y}_t$ and the full

Algorithm 1: Constructing Training Data

```
    Input: $\mathbf{x} = \{x_1, x_2, ..., x_T\}$
    Output: $\mathbf{S} = \{(\mathbf{x}_{\leq t}, l_{\leq t}) \mid 1 \leq t \leq T\}$
1   $\mathbf{S} = \{\}, \mathbf{y}_p = \{\langle s \rangle\}$
2   $\tilde{\mathbf{y}} = M_{slt}(src = \mathbf{x}, tgt = \mathbf{y}_p)$
3   for $t = 1, 2, ..., T$ do
4       $\mathbf{y}_t = M_{slt}(src = \mathbf{x}_{\leq t}, tgt_{force} = \mathbf{y}_p)$
5       $\mathbf{y}_p = LCP(\mathbf{y}_t, \tilde{\mathbf{y}})$
6       $S = S \cup (\mathbf{x}_{\leq t}, |\mathbf{y}_p|)$
7   end
8   return $\mathbf{S}$
```

input's translation $\tilde{\mathbf{y}}$, using it as the forced decoding prefix for the next round of translation (Line 5). Its length can be understood as the number of target-side tokens confidently translatable from the current input, with no need for additional video frames. This step maintains consistency with the inference stage. Finally, we obtain a data pair $(\mathbf{x}_{\leq t}, |\mathbf{y}_p|)$ consisting of the current input and the model's confident translation length. (Line 6).

3.3. Confident Translation Length Estimator

With the training data extracted in Section 3.2, we now can proceed to train our confident translation length estimator. Our estimator consists of a Transformer encoder layer and a length prediction head. The estimator first extracts the output of the translation model's encoder.

$$\mathbf{h}_{\leq t} = Encoder(\mathbf{x}_{\leq t}) \tag{4}$$

It then concatenates this output with a learnable $\langle CLS \rangle$ token. Subsequently, after passing through a Transformer encoder layer, it extracts the vector corresponding to the $\langle CLS \rangle$ token and feeds it into a multi-layer fully connected neural network (the prediction head) to predict the final length.

$$l'_{\leq t} = f_c(E([h_{cls}; \mathbf{h}_{\leq t}])[0]) \tag{5}$$

Then the estimator is optimized through Mean Square Error (MSE) loss as a regression problem.

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \left(l'_{\leq t} - l_{\leq t} \right)^2 \tag{6}$$

To maintain consistency between estimator training and translation decoding, no gradient is backpropagated to the encoder and decoder of the SLT model.

3.4. Adaptive Training with Prefix Pairs

During the inference stage, we employ an offline-trained SLT model to incrementally generate translation. However, there exists a mismatch, where during training, the model has access to the complete input, while during inference, only partial input is available (Fu et al., 2023a). Therefore, we further utilize pseudo prefix pairs to train our translation model. By training in simulated inference scenarios, our translation model becomes more adapted to actual inference and should achieve better performance.

Algorithm 2: Generating Prefix Pairs

```
     Input: $\mathbf{x} = \{x_1, x_2, ..., x_T\}$
     Output: $\mathbf{P} = \{(\mathbf{x}_{\leq t}, \mathbf{y}_{\leq t}) \mid 1 \leq t \leq T\}$
1    $\mathbf{S} = \{\}, \mathbf{y}_p = \{\langle s \rangle\}$
2    $\tilde{\mathbf{y}} = M_{slt}(src = \mathbf{x}, tgt = \mathbf{y}_p)$
3    for $t = 1, 2, ..., T$ do
4        $\mathbf{y}_t = M_{slt}(src = \mathbf{x}_{\leq t}, tgt_{force} = \mathbf{y}_p)$
5        if $|LCP(\mathbf{y}_t, \tilde{\mathbf{y}})| > |\mathbf{y}_p|$ then
6            $P = P \cup (\mathbf{x}_{\leq t}, LCP(\mathbf{y}_t, \tilde{\mathbf{y}}))$
7        end
8        $\mathbf{y}_p = LCP(\mathbf{y}_t, \tilde{\mathbf{y}})$
9    end
10   return $\mathbf{P}$
```

The process of generating prefix pairs is similar to the construction of confident translation length training data and can be carried out concurrently. The generation method is described in Algorithm 2. The difference starts from Line 5, where we check whether the length of translation to be generated at time t is greater than the already translated length. In other words, it checks if new correct words have been translated. If so, a prefix pair $(\mathbf{x}_{\leq t}, LCP(\mathbf{y}_t, \tilde{\mathbf{y}}))$ is obtained, containing partial input and its corresponding translation.

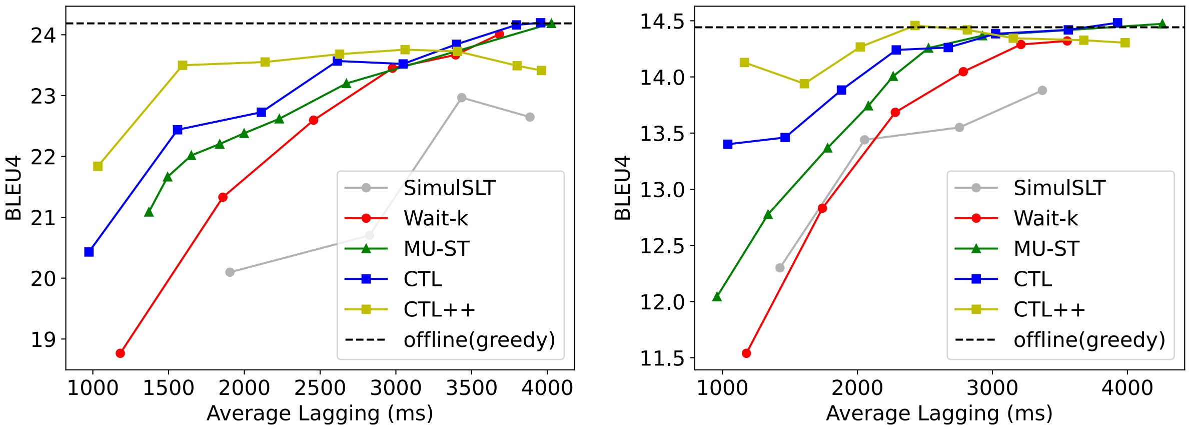
After obtaining all the prefix pairs, we mix them with full sentence pairs in a certain ratio to create new data for training the translation model. The SLT model is optimized on the mixed data by the loss $\mathcal{L}_{SLT}$ as defined in the Equation 3. We employ this model, which is more adaptive to streaming scenarios, for inference.

4. Experiments

4.1. Experimental Settings

Datasets. We validate our approach on two widely used SLT datasets: PHOENIX2014T (Camgoz et al., 2018) and CSL-Daily (Zhou et al., 2021a). PHOENIX2014T is a German Sign Language (DGS) translation dataset with the topic of weather forecasting. CSL-Daily is a Chinese Sign Language (CSL) translation dataset focusing on the daily life of the deaf community.

Model Configuration. For our translation model, the convolution subsampler has 1 1D-convolutional layer with kernel size 5, stride size 2, padding 2,



(a) PHOENIX2014T

(b) CSL-Daily

Figure 3: Quality-latency results on PHOENIX2014T and CSL-Daily. The observed points on the *SimulSLT* curve correspond to $k = \{1, 3, 5, 7\}$. Points on the *Wait-k* correspond to $k = \{1, 3, 5, 7, 9, 11\}$. Points on the *MU-ST* correspond to $\delta = \{0.3, 0.5, 0.7, 0.8, 0.85, 0.9, 0.95, 1.0\}$. Points on the *CTL* and *CTL*++ correspond to $m = \{0, 2, 4, 6, 8, 10, 14, 18\}$. All results are based on our own implementation. Numeric results can be found in Appendix A.

and 1024 filters. We use 3 layers for the semantic encoder and the translation decoder, with 8 attention heads and 512 hidden units. The target vocabularies are learned with SentencePiece[3] (Kudo and Richardson, 2018) and the size are separately set to 4500 and 3000 for PHOENIX2014T and CSL-Daily. For our confident translation length estimator, we use 1 layer of Transformer (Vaswani et al., 2017) encoder with 8 attention heads and 512 hidden units. Our estimator's prediction head comprises two linear layers with input dimension 512, hidden dimension 512, and output dimension 1.

Training Details. We implement our models with Fairseq[4] (Ott et al., 2019). For our translation model, we use an Adam optimizer with learning rate 1e-3, warmup step 5000 and an inverse square root scheduler. For our confident translation length estimator, the learning rate is set to 2e-5 with a tri-stage scheduler with phase ratio (0.1, 0.0, 0.9). We perform min-max normalization on the length label and the batch size is 32. The detection interval F is set to 10 frames. During the adaptive training process, the ratio of prefix pairs to full sentence pairs in the training data is set to 0.5:1 for PHOENIX2014T and 0.25:1 for CSL-Daily.

Evaluation. We use SacreBLEU[5] to measure the translation quality. The latency is evaluated with Average Lagging (AL) (Ma et al., 2019a) in our SimulEval-SLT.

4.2. System Settings

We compare our method with several simultaneous translation approaches.

- *SimulSLT* (Yin et al., 2021a) is currently the only existing method applied to simultaneous sign language translation. It uses the integrate-and-fire method to segment the sign video to glosses and outputs one translation token for each detected gloss.

- *Wait-k* (Ma et al., 2019a) divides the sign video into segments of fixed length, with one token being output for every segment.

- *MU-ST* (Zhang et al., 2022b) segments the sign video based on meaningful units and trains a binary classification model to decide when to translate.

- *CTL* is our proposed policy based on estimating the confident translation length but utilizes a translation model without adaptive training.

- *CTL*++ is our complete policy based on estimating the confident translation length and utilizes a translation model with adaptive training.

Note that all these methods incorporate latency control feature to adapt to various project requirements. *SimulSLT*/*Wait-k* first waits for the detection of k glosses/segments before translation and achieves this by adjusting k. *MU-ST* adjusts the threshold δ of the MU detector to correspond to different latencies, with 6 and 2 truncated words for PHOENIX2014T and CSL-Daily. Our *CTL*/*CTL*++ achieves this by removing m tokens attempted at

[3] https://github.com/google/sentencepiece

[4] https://github.com/facebookresearch/fairseq

[5] https://github.com/mjpost/sacrebleu

each step as the actual translation, serving as a more conservative translation, corresponding to higher latency but better quality.

4.3. Main Results

Figure 3 shows the trade-off between translation quality and latency on PHOENIX2014T and CSL-Daily. We observe that:

- Our method outperforms all baselines in terms of translation accuracy and latency for PHOENIX2014T and CSL-Daily. Especially at extremely low latency (when AL is around 1000ms), our method outperforms *Wait-k* by 3.0 BLEU4 on PHOENIX2014T and 2.6 BLEU4 on CSL-Daily.

- As m increases, the quality and latency of *CTL* increases accordingly, while *CTL*++ exhibits a slight decrease in quality at high latency. We consider this reasonable, as adaptive training may trade a slight decrease in offline performance for improved low-latency performance.

- Our adaptive policy based on confident translation length has certain advantages in performance compared to *MU-ST*, especially on CSL-Daily. This can be attributed to our policy's superiority as mentioned in Section 2. We also provide a translation example in Section 4.5 to understand our advantages over *MU-ST*.

- Compared to *CTL*, *CTL*++ achieves higher translation quality at medium to low latency, with a slight increase in latency, demonstrating the effectiveness of adaptive training.

4.4. Ablation Study

We conduct experiments concerning various aspects of our approach in this section. All ablation results are trained and evaluated on PHOENIX2014T.

Method	Estimator	MSE	BLEU4	AL
CTL	Xfmr-Avg	22.39	19.29	944
	Avg	21.11	19.35	882
	Xfmr-Cls	**20.57**	**20.43**	975
CTL++	Xfmr-Avg	22.39	20.63	1003
	Avg	21.11	20.71	958
	Xfmr-Cls	**20.57**	**21.84**	1033

Table 1: Performance of different estimator architectures at low latency evaluated on PHOENIX2014T. Xmfr is an abbreviation for Transformer.

4.4.1. The Impact of Different Estimator Architectures

To measure the confidence translation length, we explore three architectures of estimator.

"Xfmr-Avg" feeds the output of the translation model's encoder into a Transformer encoder layer and then averages them before feeding into a multi-layer fully connected neural network to obtain the final length prediction.

$$l'_{\le t} = f_c(avg(E(\mathbf{h}_{\le t}))) \tag{7}$$

"Avg" directly averages the output of the translation model's encoder and feeds it into a multi-layer fully connected neural network to obtain the final length prediction.

$$l'_{\le t} = f_c(avg(\mathbf{h}_{\le t})) \tag{8}$$

"Xfmr-Cls" is the architecture we use, as described in Equation 5. The experimental results are shown in Table 1. We can observe that the smaller the prediction error, the better the performance, at low latency. The "Xfmr-Cls" architecture we use has the smallest prediction error and achieves the best translation quality. This result is within our expectations because our length estimator not only guides the READ/WRITE actions but also directly decides how many tokens are written. Deviations predicted at each time step accumulate, leading to a decrease in the overall translation quality of the sentence.

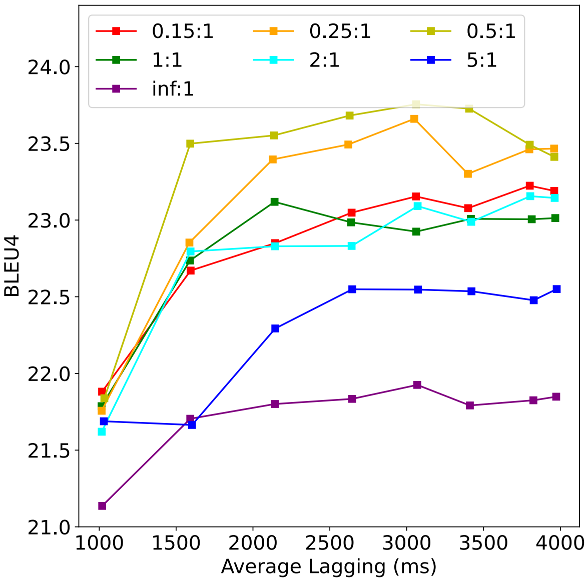
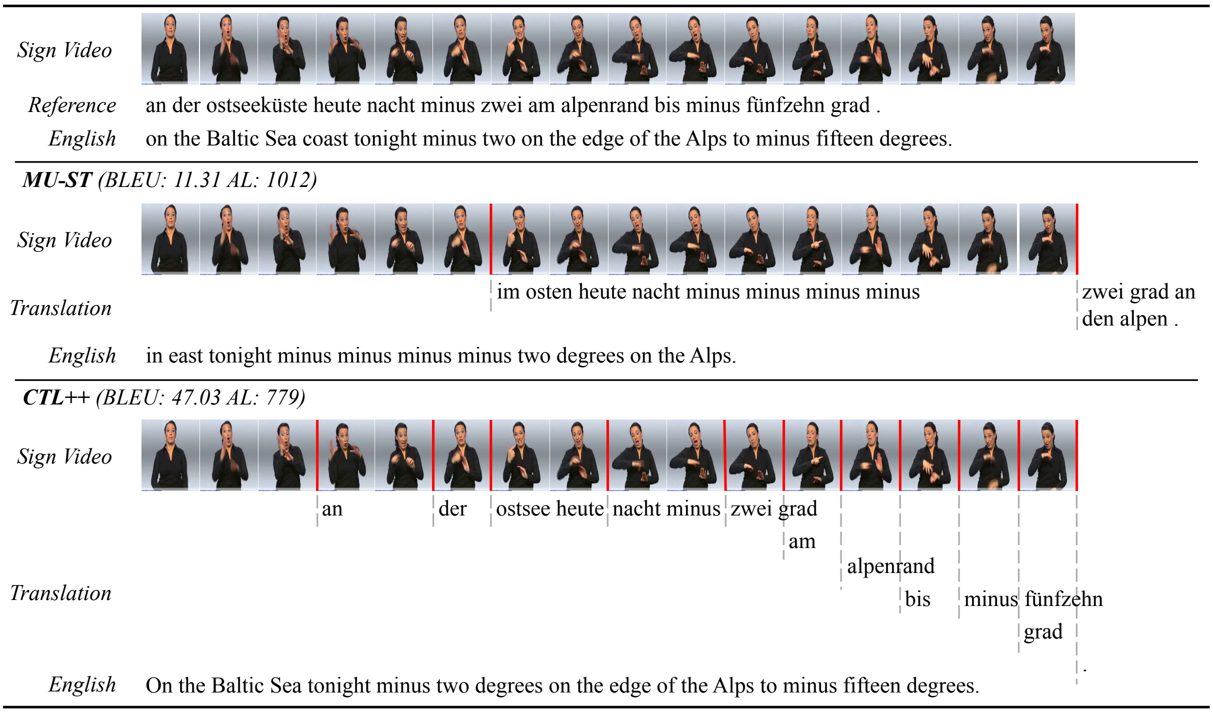


Figure 4: Performance of different prefix-full ratios for adaptive training stage evaluated on PHOENIX2014T. The best performance is achieved at a ratio of 0.5:1. Note that "inf:1" refers to training data that only includes prefix pairs, without full pairs.



Sign Video

Reference an der ostseeküste heute nacht minus zwei am alpenrand bis minus fünfzehn grad .

English on the Baltic Sea coast tonight minus two on the edge of the Alps to minus fifteen degrees.

***MU-ST** (BLEU: 11.31 AL: 1012)*

Sign Video

Translation im osten heute nacht minus minus minus minus | zwei grad an den alpen .

English in east tonight minus minus minus minus two degrees on the Alps.

***CTL++** (BLEU: 47.03 AL: 779)*

Sign Video

Translation an | der | ostsee heute | nacht minus | zwei grad | am | alpenrand | bis | minus | fünfzehn | grad | .

English On the Baltic Sea tonight minus two degrees on the edge of the Alps to minus fifteen degrees.

Figure 5: An example in the PHOENIX2014T test set, which demonstrates the effectiveness of our *CTL*++.

4.4.2. The Impact of Different Prefix-Full Ratios in Adaptive Training

We explore the impact of the ratio of prefix pairs to full sentence pairs in the training data during the adaptive training stage on the final results. The experimental result is illustrated in Figure 4. The result indicates that with an increase in the ratio of prefix pairs, the performance improves and reaches an optimal level at a certain ratio. This aligns with our hypothesis because the translation model learns the ability to output the correct translation prefix based on only partial input. However, too many prefix pairs will affect translation quality, especially under medium latency. This is because our adapted translation model should not deviate too far from the original model, as our length estimator used for READ/WRITE decisions is based on the offline translation model. In the most extreme case, training a translation model exclusively on prefix pairs predictably results in significant performance degradation across various latency scenarios.

4.5. Case Study

We conduct case study to demonstrate the superiority of *CTL*++ model over *MU-ST* model. As shown in the Figure 5, we can observe that: (1) *MU-ST* often needs to wait for more video frames before starting translation as its READ/WRITE decisions hinge on meaningful units, thus introducing a higher latency. Conversely, the *CTL*++ model initiates translations once the confident translation length corresponding to the current input surpasses the length of the translation history, offering faster real-time translations and making it more efficient in simultaneous scenarios. (2) In situations where the sign language sequences in the streaming video are yet to be fully rendered, the *MU-ST* model demonstrates a propensity to prematurely translate these incomplete actions. This often leads to inaccurate translations due to the absence of video semantics, such as the repetition of word "minus" in the figure above. In contrast, *CTL*++ can effectively avoid this error. By relying on the constraint of confident translation length, it ensures that only gestures with sufficient confidence, indicative of their completeness, are translated, thereby eliminating potential translation anomalies.

5. Conclusion

In this paper, we further explore the field of simultaneous sign language translation and adopt an adaptive policy for the first time in this task. Our newly proposed policy introduces the concept of confident translation length and determines when and how to write translations by training an external length estimator. Additionally, we generate pseudo prefix pairs through the longest common prefix algorithm, further adapting our translation model to streaming inference. Our policy places a greater emphasis on understanding the current sign video content and can be applied to more SLT datasets without gloss annotation. We conduct comprehensive experiments using our SimulEval-SLT toolkit and results on PHOENIX2014T and CSL-Daily show the superiority of our policy, especially in low-latency scenarios.

We hope our work can inspire future studies on simultaneous SLT and relevant tasks. In the future, we are interested in leveraging more powerful and efficient sign video encoders, utilizing prior knowledge of sign language to assist READ/WRITE decisions, and other methods to enhance our system.

6. Limitations

This paper focuses on simultaneous sign language translation, which performs translation synchronously during the reception of the sign video. Its low-latency feature can facilitate seamless communication between sign language users and spoken language users. While our work has made some progress, it is evident that there is still a long road ahead. It is important to note that this research is limited to public datasets with a restricted number of samples collected under constrained conditions. As a result, the findings may not directly apply to more complex real-world applications. When using this technology to make critical decisions, it is crucial to incorporate domain expertise and human supervision, as there is a possibility of generating erroneous or potentially harmful translations.

7. Acknowledgements

We would like to thank all the anonymous reviewers for their insightful and valuable comments. This work was supported in part by the National Natural Science Foundation of China under Grant No. 62076211 and in part by the University-Industry Cooperation Programs of Fujian Province of China under Grant No. 2023H6001.

8. Bibliographical References

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.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7784–7793.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020a. Multi-channel Transformers for Multi-articulatory Sign Language Translation. In *European Conference on Computer Vision*, pages 301–319.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020b. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 2021. Direct Simultaneous Speech-to-Text Translation Assisted by Synchronized Streaming ASR. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4618–4624.

Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022a. A simple multi-modality transfer learning baseline for sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5120–5130.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. 2022b. Two-stream network for sign language recognition and translation. *Advances in Neural Information Processing Systems*, 35:17043–17056.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. Learning When to Translate for Streaming Speech. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient Wait-k Models for Simultaneous Machine Translation. In *Interspeech 2020-Conference of the International Speech Communication Association*, pages 1461–1465.

Biao Fu, Minpeng Liao, Kai Fan, Zhongqiang Huang, Boxing Chen, Yidong Chen, and Xiaodong Shi. 2023a. Adapting Offline Speech Translation Models for Streaming with Future-Aware Distillation and Inference. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16600–16619.

Biao Fu, Peigen Ye, Liang Zhang, Pei Yu, Cong Hu, Xiaodong Shi, and Yidong Chen. 2023b. A Token-Level Contrastive Framework for Sign Language Translation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Shiwei Gan, Yafeng Yin, Zhiwei Jiang, Kang Xia, Lei Xie, and Sanglu Lu. 2023. Contrastive Learning for Sign Language Recognition and Translation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 763–772. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 2017. Learning to Translate in Realtime 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.

Shoutao Guo, Shaolei Zhang, and Yang Feng. 2023. Learning Optimal Policy for Simultaneous Machine Translation via Binary Search. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2318–2333.

Hezhen Hu, Weichao Zhao, Wengang Zhou, and Houqiang Li. 2023. SignBERT+: Hand-model-aware Self-supervised Pre-training for Sign Language Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tao Jin and Zhou Zhao. 2021. Contrastive disentangled meta-learning for signer-independent sign language translation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5065–5073.

Tao Jin, Zhou Zhao, Meng Zhang, and Xingshan Zeng. 2022. MC-SLT: Towards Low-Resource Signer-Adaptive Sign Language Translation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4939–4947.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

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.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, et al. 2019a. STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An Evaluation Toolkit for Simultaneous Translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting Simultaneous Text Translation to End-to-End Simultaneous Speech Translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2019b. Monotonic Multihead Attention. In *International Conference on Learning Representations*.

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.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Does Simultaneous Speech Translation need Simultaneous Models? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 141–153.

Sara Papi, Matteo Negri, and Marco Turchi. 2023. Attention as a Guide for Simultaneous Speech Translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796.

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.

Aoxiong Yin, Zhou Zhao, Jinglin Liu, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. 2021a. SimulSLT: End-to-end simultaneous sign language translation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4118–4127.

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021b. Including Signed Languages in Natural Language Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360.

Pei Yu, Liang Zhang, Biao Fu, and Yidong Chen. 2023. Efficient Sign Language Translation with a Curriculum-based Non-autoregressive Decoder. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5260–5268. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. RealTranS: End-to-End Simultaneous Speech Translation with Convolutional Weighted-Shrinking Transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474.

Biao Zhang, Mathias Müller, and Rico Sennrich. 2022a. SLTUNET: A Simple Unified Model for Sign Language Translation. In *The Eleventh International Conference on Learning Representations*.

Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2022b. Learning adaptive segmentation policy for end-to-end simultaneous translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7862–7874.

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. 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.

Shaolei Zhang and Yang Feng. 2021b. 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 Multi-head Attention for Simultaneous Machine Translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3019–3030.

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.

Shaolei Zhang and Yang Feng. 2022c. Reducing Position Bias in Simultaneous Machine Translation with Length-Aware Framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6775–6788.

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.

Shaolei Zhang, Shoutao Guo, and Yang Feng. 2022c. Wait-info Policy: Balancing Source and Target at Information Level for Simultaneous Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2249–2263.

Rui Zhao, Liang Zhang, Biao Fu, Cong Hu, Jinsong Su, and Yidong Chen. 2024. Conditional Variational Autoencoder for Sign Language Translation with Cross-Modal Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021a. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1325.

Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. 2021b. Spatial-temporal multi-cue network for sign language recognition and translation. *IEEE Transactions on Multimedia*, 24:768–779.

A. Numeric Results

We provide the numeric results for Figure 3(a) in Table 2, and for Figure 3(b) in Table 3. We also include additional metrics for evaluation.

SimulSLT								
k	1	3	5	7				
AL	1905	2826	3435	3885				
BLEU1	45.69	45.89	47.34	47.18				
BLEU2	32.66	33.03	35.20	34.85				
BLEU3	24.94	25.46	27.74	27.47				
BLEU4	20.10	20.70	22.97	22.65				
ROUGE	45.62	46.91	48.36	49.47				
Wait-k								
k	1	3	5	7	9	11		
AL	1181	1858	2457	2977	3394	3683		
BLEU1	43.95	48.06	49.64	50.59	50.66	50.99		
BLEU2	30.85	34.51	36.05	37.17	37.31	37.63		
BLEU3	23.38	26.37	27.77	28.79	28.95	29.34		
BLEU4	18.77	21.33	22.60	23.45	23.67	24.01		
ROUGE	44.35	47.50	48.86	49.77	49.83	49.99		
MU-ST								
δ	0.3	0.5	0.7	0.8	0.85	0.9	0.95	1.0
AL	1370	1494	1650	1837	1998	2231	2674	4027
BLEU1	48.10	48.89	49.39	49.38	49.75	50.02	50.63	50.99
BLEU2	34.58	35.35	35.78	35.90	36.21	36.45	37.02	37.67
BLEU3	26.26	26.96	27.37	27.54	27.76	28.04	28.58	29.49
BLEU4	21.09	21.66	22.02	22.20	22.38	22.62	23.20	24.19
ROUGE	47.82	48.41	48.77	48.84	48.85	49.19	49.57	50.04
CTL								
m	0	2	4	6	8	10	14	18
AL	974	1559	2112	2613	3048	3400	3796	3955
BLEU1	48.46	49.67	49.95	50.65	50.91	50.79	51.16	51.04
BLEU2	34.16	36.14	36.30	37.23	37.31	37.43	37.73	37.71
BLEU3	25.70	27.73	27.96	28.88	28.88	29.15	29.49	29.51
BLEU4	20.43	22.44	22.73	23.57	23.52	23.84	24.16	24.20
ROUGE	46.99	48.66	48.88	49.69	49.73	49.87	50.09	50.07
CTL++								
m	0	2	4	6	8	10	14	18
AL	1033	1592	2137	2629	3061	3406	3800	3960
BLEU1	48.57	49.74	49.54	49.91	49.54	49.47	49.35	49.40
BLEU2	35.28	36.61	36.60	36.64	36.66	36.59	36.46	36.40
BLEU3	27.12	28.67	28.65	28.73	28.80	28.74	28.56	28.48
BLEU4	21.84	23.50	23.55	23.68	23.75	23.73	23.49	23.41
ROUGE	46.94	48.24	48.23	48.26	48.34	48.23	48.16	48.06

Table 2: Numeric results on PHOENIX2014T (Figure 3(a))

SimulSLT								
k	1	3	5	7				
AL	1427	2054	2756	3370				
BLEU1	41.19	42.17	42.09	41.52				
BLEU2	27.15	28.45	28.65	28.41				
BLEU3	18.09	19.38	19.59	19.60				
BLEU4	12.30	13.44	13.55	13.88				
ROUGE	40.50	42.63	43.28	43.18				
Wait-k								
k	1	3	5	7	9	11		
AL	1178	1741	2281	2784	3211	3554		
BLEU1	38.95	41.60	42.62	43.08	43.06	43.01		
BLEU2	25.40	27.66	28.72	29.10	29.28	29.25		
BLEU3	16.81	18.53	19.48	19.86	20.11	20.09		
BLEU4	11.54	12.83	13.68	14.05	14.29	14.32		
ROUGE	37.88	40.25	41.13	41.58	41.73	41.76		
MU-ST								
δ	0.3	0.5	0.7	0.8	0.85	0.9	0.95	1.0
AL	960	1338	1780	2081	2265	2527	2926	4259
BLEU1	40.64	41.69	42.67	42.83	43.38	43.50	43.58	43.37
BLEU2	26.75	27.66	28.47	28.76	29.23	29.45	29.49	29.54
BLEU3	17.73	18.50	19.20	19.55	19.89	20.10	20.17	20.29
BLEU4	12.04	12.78	13.37	13.74	14.01	14.26	14.37	14.47
ROUGE	39.06	40.00	40.59	40.86	41.39	41.66	41.77	41.89
CTL								
m	0	2	4	6	8	10	14	18
AL	1040	1465	1882	2288	2673	3023	3562	3927
BLEU1	42.80	43.29	43.32	43.37	43.60	43.56	43.16	43.30
BLEU2	28.51	28.82	29.07	29.34	29.55	29.57	29.37	29.52
BLEU3	19.18	19.34	19.70	20.04	20.16	20.25	20.19	20.29
BLEU4	13.40	13.46	13.88	14.24	14.26	14.38	14.42	14.48
ROUGE	40.64	40.97	41.42	41.76	41.89	41.96	41.88	41.95
CTL++								
m	0	2	4	6	8	10	14	18
AL	1163	1608	2022	2427	2815	3155	3677	3983
BLEU1	43.62	43.38	43.43	43.72	43.72	43.77	43.81	43.72
BLEU2	29.65	29.36	29.54	29.87	29.81	29.79	29.78	29.74
BLEU3	20.15	19.89	20.19	20.48	20.41	20.35	20.34	20.31
BLEU4	14.13	13.94	14.27	14.46	14.42	14.34	14.33	14.30
ROUGE	41.62	41.40	41.41	41.68	41.72	41.63	41.78	41.70

Table 3: Numeric results on CSL-Daily (Figure 3(b))