Improving End-to-end Sign Language Translation with Adaptive Video Representation Enhanced Transformer

Zidong Liu, Jiasong Wu, Zeyu Shen, Xin Chen, Qianyu Wu, Zhiguo Gui, Lotfi Senhadji, Huazhong Shu

Improving End-to-end Sign Language Translation with Adaptive Video Representation Enhanced Transformer

Zidong Liu, Jiasong Wu, *Member, IEEE*, Zeyu Shen, Xin Chen, Qianyu Wu, Zhiguo Gui, Lotfi Senhadji, *Senior Member, IEEE*, Huazhong Shu, *Senior Member, IEEE*

***Abstract*—The aim of end-to-end sign language translation (SLT) is to interpret continuous sign language (SL) video sequences into coherent natural language sentences without any intermediary annotations, i.e., glosses. However, end-to-end SLT suffers several intractable issues: (i) the temporal correspondence constraint loss problem between SL videos and glosses, and (ii) the weakly supervised sequence labeling problem between long SL videos and sentences. To address these issues, we propose an adaptive video representation enhanced Transformer (AVRET), with three extra modules: adaptive masking (AM), local clip self-attention (LCSA) and adaptive fusion (AF). Specifically, we utilize the first AM module to generate a special mask that adaptively drops out temporally important SL video frame representations to enhance the SL video features. Then, we pass the masked video feature to the Transformer encoder consisting of LCSA and masked self-attention to learn clip-level and continuous video-level feature information. Finally, the output feature of encoder is fused with the temporal feature of AM module via the AF module and use the second AM module to generate more robust feature representations. Besides, we add weakly supervised loss terms to constrain these two AM modules. To promote the Chinese SLT research, we further construct CSL-FocusOn, a Chinese continuous SLT dataset, and share its collection method. It involves many common scenarios, and provides SL sentence annotations and multi-cue images of signers. Our experiments on the CSL-FocusOn, PHOENIX14T, and CSL-Daily datasets show that the proposed method achieves the competitive performance on the end-to-end SLT task without using glosses in training. The code is available at https://github.com/LzDddd/AVRET.**

***Index Terms*—End-to-end sign language translation, adaptive masking, local clip self-attention, adaptive fusion, continuous sign language video dataset, without using glosses.**

This work was supported in part by the National Key Research and Development Program of China (No. 2022YFE0116700), and in part by the National Natural Science Foundation of China under Grants 62171125, 61876037, 31800825, and in part by the innovation project of Jiangsu Province under grants BZ2023042, BY2022564.

Zidong Liu, Jiasong Wu, Zeyu Shen, Xin Chen, Qianyu Wu and Huazhong Shu are with LIST, Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications, Southeast University, Ministry of Education, Nanjing 210096, China; Jiangsu Provincial Joint International Research Laboratory of Medical Information Processing, Southeast University, Nanjing 210096, China; Centre de Recherche en Information Biomédicale Sino-français (CRIBs), Univ-Rennes, INSERM, Southeast University, Rennes F-3502, France, Nanjing 210096, China. (e-mail: {zd_liu, jswu, shenzeyu, xinchen-seu, 230218787, shu.list}@seu.edu.cn). Zidong Liu is the first author, Jiasong Wu is the co-first author and Huazhong Shu is the corresponding author.

Zhiguo Gui is with the State Key Laboratory of Dynamic Testing Technology, North University of China, 030051 Taiyuan, China. (e-mail: guizhiguo@nuc.edu.cn).

Lotfi Senhadji is with Univ-Rennes, INSERM, LTSI-UMR 1099, Rennes F-3502, France; and with CRIBs, Univ-Rennes, INSERM, Southeast University, Rennes F-3502, France. (e-mail: lotfi.senhadji@univ-rennes1.fr).

Fig. 1. Two ways of sign language translation (SLT). The black dotted pipeline denotes two-stage SLT based on continuous sign language recognition (CSLR) and gloss to text (G2T). The black continuous pipeline denotes end-to-end SLT without glosses.

I. Introduction

SIGN language (SL) is the main medium of communication for the deaf. It can convey visual information through the cooperation of multiple organs (e.g., hands, body, lip and facial expressions), and has unique characteristics different from spoken language. However, the complex body movements and linguistic logic make it difficult for normal people to understand this language, which greatly limits the social scope of the deaf. Therefore, it can undoubtedly reduce the communication burden of the deaf in their daily lives if SL can be automatically translated into natural language. As shown in Figure 1, SL translation (SLT) can be implemented not only in end-to-end ways, but also in the way of two-stage: continuous SL recognition (CSLR) and gloss to text (G2T). CSLR can recognize continuous SL videos as gloss sequences, and the G2T can translate the gloss sequences into SL sentence. Furthermore, it can also generate glosses and SL sentences simultaneously via CSLR and SLT multi-task joint learning.

Currently, most of the work in SL research focus on CSLR [1]–[8] and SLT with glosses [9]–[19], few research works concentrate on end-to-end SLT without using glosses [20]–[24]. The main reason is that gloss sequences are consistent with sign gestures in the SL video. It can be used as the intermediate mapping between SL videos and sentences to allow CSLR and SLT models to alleviate the syntactic alignment problem by learning the temporal correspondence between SL videos and gloss sequences, thus achieving better SLT performance through multi-task joint learning or two-stage ways. However, gloss annotations can only be performed

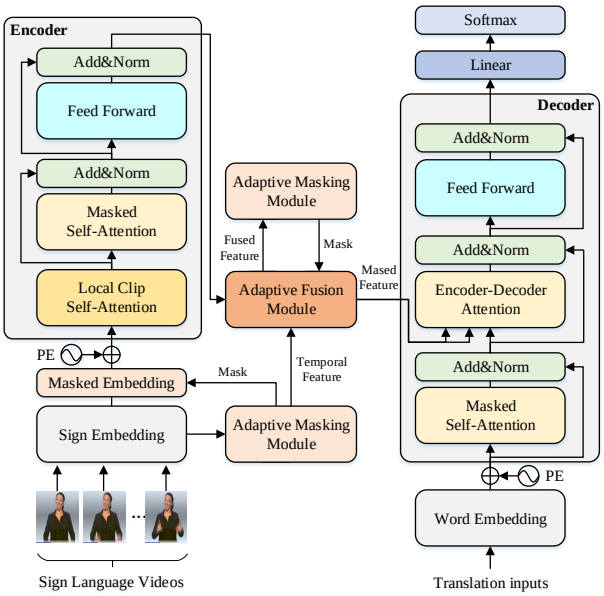


Fig. 2. An overview of our AVRET model. The output of bottom adaptive masking (AM) module consists of mask and temporal feature, where the mask can drop out the video frame representations and the temporal feature is passed to the adaptive fusion (AF) module. The AF module can adaptively fuse the temporal feature of AM module with the output feature of encoder. Then, the fused feature generated by the AF module is masked by the top AM module and the masked feature is passed to the decoder. (PE: Positional Encoding)

by SL experts [20]. In contrast, end-to-end SLT can directly translate continuous SL videos into sentences without using glosses. And, unlike CSLR which focuses on explicitly aligning SL video frames with glosses [5] and enhancing short-term temporal information [8], end-to-end SLT is not sensitive to the alignment of SL video frames and words, which mainly needs to enhance the overall mapping relationship between SL videos and sentences. So, end-to-end SLT has a wider range of applications and can be easily transferred to other SL datasets. However, end-to-end SLT is a very challenging task due to the following reasons: (i) the frame size of SL videos is longer than the gloss sequences length and contains more complex corpus information; (ii) the grammar rules of natural language sentences are different from SL videos and gloss sequences; (iii) it is a weakly supervised problem since continuous SL videos do not have boundary annotations for SL action transitions. Considering the difficulty of obtaining gloss annotations and the strong migration of end-to-end SLT methods, in this paper, we concentrate on the task of end-to-end SLT, aiming to alleviate the above difficulties and improve the translation performance of continuous SL videos without using glosses in training.

To improve the translation quality of SLT, Camgöz et al. [10] first introduced the Transformer network [25] to SLT and showed good performance. Since then, many research works also use the Transformer as the SLT backbone network and improve it in many aspects, including attention mechanism [14], [23], gloss-text joint learning [12], [15], [17], [18], network pre-training [19], [24], and different data inputs (e.g., SL video clips [22], multi-cue images [9], [11], [13], [16], [21]). However, most of the above methods mainly focus on the improvement of SLT network based on glosses, and only a few of them [21]–[24] are end-to-end SLT without glosses. Besides, most methods use a single type of data input, while ignoring the convergence between different data inputs and the generalization problem between long sequences due to the loss of glosses. And, SL emphasizes the cooperation of multiple semantic organs to convey visual information, traditional data enhancement methods (e.g., flipping, scaling, random cropping) and multi-cue (e.g., hands, face, body keypoints) images tend to affect the relative position information and temporal consistency of SL semantic organs. Therefore, how to make better use of SL videos and enhance network generalization is the key to improving end-to-end SLT performance without glosses. Inspired by this, video representation learning (VRL) methods [26]–[29], which have extensive research in video processing, feature and model generalization enhancement, are a worthwhile direction to explore. These methods not only use full-size frames to learn the complete visual information, but also improve the performance on a range of downstream tasks by enhancing the video feature representation. However, most of VRL methods require a huge computational resource to satisfy the data inputs with large batch size. The large frame size of long SL videos also makes training more difficult. Therefore, we try to simplify some effective VRL methods and incorporate them into SLT network in order to improve the translation effect by enhancing its generalization capability.

In this paper, we propose an adaptive video representation enhanced Transformer (AVRET) to learn robust continuous SL video feature representations. The overview of our AVRET is shown in Figure 2. It is equipped with three extra modules: adaptive masking (AM), local clip self-attention (LCSA), and adaptive fusion (AF). To improve the feature robustness of SL videos, we first introduce an AM module based on the Generator of VideoMoCo [28], which can provide a special mask to adaptively drop out the feature representation of video frames. Specifically, we extract the temporal feature of SL videos by the BiLSTM [30] of AM module and drop out the temporally important frame representations based on the mask generated by this feature to enhance the video feature. Since the Transformer encoder does not destroy the dimension of input feature, so the AM module can be equipped not only before the encoder, but also between the encoder and decoder. Note that the AM module does not have corresponding annotations for supervised training. In order to solve the semantic ambiguity problem that may result from increasing the dropout thresholds of video frame representations, we pass the temporal feature of AM module to the Transformer decoder and use it to simulate the Discriminator of VideoMoCo. Then, we stabilize the dropout effect of AM module by establishing a weakly supervised loss constraint for its decoding results using spoken translation sentences as pseudo-labeling. To enhance the local semantics learning ability of the network for SL videos, unlike the common video clip partition (CCP) in [5], [22] and the use of short-term neighboring frames [8], we add an LCSA to the Transformer encoder. It can split the continuous video features at the clip-level and then extend

TABLE I
SUMMARY INFORMATION OF SOME PUBLIC AVAILABLE CONTINUOUS SL VIDEO DATASETS. ('**C**' REPRESENTS CONTINUOUS: THE CORPUS IS COMPOSED OF CONTINUOUS VIDEOS. '**G**' REPRESENTS GLOSS: THE CORPUS IS COMPOSED OF GLOSS-LEVEL ANNOTATIONS. '**T**' REPRESENTS TRANSLATION: THE CORPUS HAS SL TRANSLATION SENTENCES. **VOCAB.**: THE VOCABULARY SIZE OF SL TRANSLATION SENTENCES. '**ASL**', '**DGS**' AND '**KSL**' REPRESENT AMERICAN SL, GERMAN SL, AND KOREAN SL, RESPECTIVELY.)

Datasets	Language	Attribute				Statistics			Source
		C	G	T	Resolution	#Vocab.	#Videos	#Signers	
WLASL [6]	ASL				-	-	21,083	119	Web
ASLLVD [33]	ASL				-	-	9,800	6	Lab
MS-ASL [34]	ASL				-	-	25,513	222	Web
How2Sign [35]	ASL	✓	✓	✓	1280×720	15,686	35,191	11	Lab
YouTube-ASL [36]	ASL	✓		✓	-	60,000	11,093	>2519	Web
PHOENIX-2014 [37]	DGS	✓	✓		210×260	-	6,841	9	TV
PHOENIX14T [20]	DGS	✓	✓	✓	210×260	2,887	8,257	9	TV
SIGNUM [38]	DGS	✓		✓	780×580	450	12,150	25	Lab
KSL-Guide-Sentence [39]	KSL	✓	✓	✓	1920×1080	319	40,000	20	Lab
VCSL [2]	CSL				1280×720	-	125,000	50	Lab
CCSL [3]	CSL	✓	✓		1280×720	-	25,000	50	Lab
CSL-Daily [12]	CSL	✓	✓	✓	1920×1080	2,343	20,654	10	Lab
CSL-FocusOn (ours)	CSL	✓		✓	224×224	28,325	11,665	6	TV
CSL-FocusOn (subset)	CSL	✓		✓	224×224	21,058	4,200	6	TV

keyframe features from multiple neighboring clips. Next, inter-cross attention (ICA) is set for each clip to enhance its local information. And, the connection between LCSA and masked self-attention allows the encoder to learn local and global information of SL videos. Moreover, since both the temporal feature of the AM module and the output feature of the encoder can be decoded, and these two features are different in terms of temporal and spatial learning. Therefore, unlike [8], [29] only use simple concatenation, we introduce an AF module based on GRF [31] and AFA [32], which can adaptively fuse the two features to generate a robust feature representation. Owing to the more robust feature representation, our method enables the Transformer to learn more spatio-temporal information, and thus improving the end-to-end SLT effect without using gloss annotations in training.

Besides, we note that the most widely used large-scale continuous SL video datasets are mainly annotated in German [20], [37], [38] or English [6], [33]–[35], while Chinese continuous SL video datasets are only CCSL [3] and CSL-Daily [12]. However, the two datasets were collected with SL experts. While this collection form can ensure the quality of data annotations, there are limitations in video content scenarios and subsequent dataset extension. Therefore, in order to explore an automated and efficient data collection method and to promote academic research on Chinese SL (CSL), we construct a continuous CSL video dataset, namely CSL-FocusOn, based on a Chinese news program. The automated data collection method allows us to obtain a large-scale CSL dataset by manually filtering the final SL video segments without SL experts, and also facilitates the subsequent dataset extension. Benefit from the diversity of content in news programs, CSL-FocusOn can cover many common scenarios (e.g., daily life, weather, news, medical care) and special scenarios (e.g. corona virus disease 2019, economy), while providing SL sentences with large vocabulary size and multi-cue images of signers. It contributes to the diversity and completeness of topics in dataset. Besides, video segments can be divided into different subsets based on the segment duration to further explore the effect of long SL video translation. We also compare the experimental results of several end-to-end SLT methods on the CSL-FocusOn dataset and perform a detailed analysis.

The contributions of this paper can be summarized as follows:

1) We propose a simple and effective adaptive video representation enhanced Transformer (AVRET) to alleviate the weakly supervised problem of end-to-end SLT and improve the translation performance. Three extra modules in AVRET can be freely added into the Transformer-based network.
2) We design and share a collection method of CSL video dataset, and use it to construct the first news corpus-based CSL video dataset, namely CSL-FocusOn. It contains rich corpus contents and is easier to extend.
3) Our method is validated on the CSL-FocusOn, CSL-Daily and the benchmark dataset RWTH-PHOENIX-Weather 2014T [20] (PHOENIX14T), which achieves the competitive accuracy performance on the end-to-end SLT task without using glosses in training.

The rest of this paper is organized as follows. In Section II, we review some public available SL datasets and related works on SLT and VRL. In Section III, we present the model architecture of AVRET and the process of CSL-FocusOn data collection and annotation. In Section IV, we provide implementation details and ablation analysis of the model, and finally present the experimental results and compared them with several models on the same task. In Section V, we conclude this paper.

II. Related Work

A. Sign Language Dataset

A summary information of some publicly video-based SL datasets is shown in Table I. Some of these datasets [2], [6], [33], [34] are composed of word-level videos, which can only be used on isolated SLR task. This task can be considered as a SL classification problem, which aims to recognize an individual SL word from short video. To further implement the CSLR and SLT tasks, some datasets [3], [12], [20], [35], [36], [38], [39] provide continuous gloss and translation annotations. Among them, PHOENIX14T [20] is the benchmark dataset for both CSLR and SLT because it contains high-quality gloss and translation annotations, and its corpus content is more realistic and close to the real life compared to other datasets since it is collected from weather forecast TV programs. In contrast, while both the CSL-Daily [12] and How2Sign [35] datasets can provide high-quality translation annotations, they are produced by the laboratory with limited corpus content and the collection form is not easy. Besides, most of the publicly available datasets currently maintain a thousand-level vocabulary for both gloss and translation annotations, and only How2Sign and YouTube-ASL [36] have a vocabulary size of 16k and 60k, respectively. Although low vocabulary size can reduce the difficulty of SLT, it also limits the content of single sentence. Therefore, as a contribution to the CSL study, our CSL-FocusOn was collected from a Chinese news program. It ensures the richness of corpus content while increasing the vocabulary size of translation annotations to 20k. To keep the frame size of video sequences on the CSL-FocusOn close to the average frame size of PHOENIX14T, we make a subset based on the original dataset by filtering videos with the duration of less than 35 seconds and used it for the subsequent end-to-end SLT studies. Some information about CSL-FocusOn and its subset are also shown in Table I.

B. Sign Language Translation

Early research works on SLT mainly used RNN-based attention network architectures [20], [21]. Since the introduction of BiLSTM [30], researchers gradually applied it to SLT [9], [13] because it can well solve the long-term dependency problem of RNN model and further enhance the context learning capability. With the public and wide application of Transformer [25] in natural language processing (NLP), the Transformer network, which relies entirely on attention mechanisms and feed-forward layers, greatly improves the quality and efficiency of various sequence translation tasks. And, more and more research works also apply and improve it to computer vision (CV) tasks and show excellent performance, such as video captioning [29], image captioning [32], oriented object detection in remote sensing imagery [40], online anomaly detection [41], and so on. Therefore, Camgöz et al. [10] first applied the Transformer network to SLT, and achieved a good translation results while verifying the effectiveness of jointly learning. In recent research work, Xie et al. [14] proposed a PiSLTRc model, which improves the Transformer network by content-aware and position-aware temporal convolution and disentangled relative position encoding. Yin et al. [15] proposed a boundary predictor from the perspective of simultaneous SLT with low latency to simulate the correspondence between SL videos and vocabulary. Zhou et al. [13] proposed a spatio-temporal multi-cue network (STMC-T), aiming to perform video-based SLT by using multi-cue images. Chen et al. [19] performed pre-training on multiple datasets using multi-modality transfer learning (MMTLB). Kan et al. [16] represented the semantic organs of SL with a hierarchical spatio-temporal graph neural network (HST-GNN), and use it to learn the semantic information of SL. Fu et al. [17] built a token-level contrastive framework for SLT (ConSLT). Zhou et al. [12] incorporated massive spokens text into SLT training via parallel data and sign back-translation (SignBT) method.

In terms of end-to-end SLT without glosses, Li et al. [22] proposed a TSPNet and set a fixed clip frame size that allows it to learn discriminative SL video features by using the semantic hierarchy between video clips. Yin et al. [23] proposed a gloss attention GASLT, which allows the attention to focus on the video clips that have the same local semantics and helps the model understand SL videos via knowledge transfer. GFSLT-VLP [24] combines contrastive language-image pre-training (CLIP) with masked self-supervised learning to pre-train model, and transfer the prior knowledge from pre-trained model into SLT framework, thus improving the SLT effect.

However, the performance improvement of SLT methods relies mostly on glosses or model pre-training. When it is not involved in training, the generalization and translation performance of network is limited to be improved due to the loss of correspondence gloss constraint and prior knowledge. Moreover, the above methods mainly use all video frames and a single data input method, but rarely involve the study of special frame selection and data enhancement methods. In contrast, we integrate and simplify VRL methods to improve the translation performance by enhancing feature robustness and model generalization. To investigate how different data inputs can be used on SLT, we note that Yan et al. [29] designs a global-local framework (GLR) on the video captioning task that can encode video clips at different ranges to improve linguistic expression. Since SLT also belongs to the video captioning task, we combine their research idea with SL characteristics and migrate it to our method. Therefore, we add a clip-level feature learning module LCSA to the Transformer encoder and endow the encoder with the learning ability of local and global video information.

C. Video Representation Learning

Video representation learning (VRL) methods are mainly based on unsupervised learning to focus on the temporal properties between continuous video frames. Pan et al. [28] proposed a VideoMoCo which can adaptively drop out several temporally important frames from the original video sequence through a Generator ($\mathcal{G}$), and pass the video of dropped frames together with the full frame video to the Discriminator ($\mathcal{D}$) to learn similar feature representations. Huang et al. [42] explored temporal context information by sampling rate order prediction. Jenni et al. [43] analysed and evaluated four different video temporal transformation methods, including

speed, random, periodic and warp. Tao et al. [44] proposed an inter-intra contrastive framework based on self-supervised contrastive learning, which can force the model to learn better more discriminative temporal information by setting intra-positive and intra-negative clip samples within the same video and negative clip samples on different videos. Moreover, some multi-modal based learning methods can also learn more video information by introducing multi-modal data (e.g., optical flow [45], audio [46] and text [47]). In the specific applications of VRL, Yan et al. [29] proposed a GLR framework for video captioning, which can encode video representations at different ranges (e.g., long-range, short-range, and local-keyframe) to improve linguistic expression. Considering the SL visual characteristics and the complexity of original VRL methods, we simplify some methods from VideoMoCo and GLR. Unlike VideoMoCo, we let the model adaptively generate a mask to drop out frame feature representations instead of video frames. And, we use the Transformer decoder to simulate the $\mathcal{D}$ in VideoMoCo. Furthermore, unlike GLR extracts video representations from three ranges and fuses them via concatenation, we split the original video into multiple clips and combine the short-range clips with local-keyframe to increase the neighbor information. And, we can obtain more robust feature representations by additional LCSA and AF module.

III. METHOD

A. Sign Language Transformer and Model Overview

SL Transformer is mainly applied to sequence-to-sequence SLT tasks, and the whole network consists of an encoder-decoder architecture. The encoder can map the SL video sequences $(x_1, ..., x_n)$ with n frames into a continuous sequence feature representation $\mathcal{V} \in \mathbb{R}^{n \times d_m}$, where d_m denotes the feature dimension. After obtaining $\mathcal{V}$, the decoder is able to generate a sentence sequence $\mathcal{S} = \{\omega_u\}_{u=1}^{U}$ with conditional probability $p(\mathcal{S}|\mathcal{V})$ by consuming previously generated words each time in an auto-regressive manner. The most important part of SL Transformer is the stackable masked self-attention (MSA). The standard MSA can be formulated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_m}})V, \quad (1)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), \quad (2)$$

$$MSA(Q, K, V) = Concat(head_1, ..., head_h)W, \quad (3)$$

where query Q, key K, and value $V \in \mathbb{R}^{n \times d_m}$ represent the input matrices of single-head attention $head_i$, $i = (1, 2, ..., h)$. $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_m \times \frac{d_m}{h}}$, and $W \in \mathbb{R}^{d_m \times d_m}$ represents projection parameter matrices. $Concat(\cdot)$ is the function that concatenates h single-head attentions into MSA.

Since our work aims to improve the Transformer network by adding extra modules in order to generate more robust representations of SL videos, we omit the detailed description of the Transformer network and refer to [10], [25] for more specific information.

As shown in Figure 2, our model consists of three extra modules, including adaptive masking (AM) module, local clip self-attention (LCSA), adaptive fusion (AF) module. Firstly, given a SL video sequence $(x_1, ..., x_n)$ with n frames, each frame is concatenated to a continuous video feature after extracting spatial representation by sign embedding. Then, the AM module equipped with BiLSTM is used to capture the temporal information of the video feature, and the generated temporal feature and mask are passed into the AF module and Transformer encoder, respectively. And, the encoder equipped with LCSA and MSA can learn local and global information simultaneously. Next, the spatial feature of encoder and the temporal feature of AM module are adaptively fused by the AF module, and the fused feature will masked by the second AM module. Finally, the masked feature and sentence sequences are used for sequence learning and inference by the Transformer decoder to generate the SL sentence.

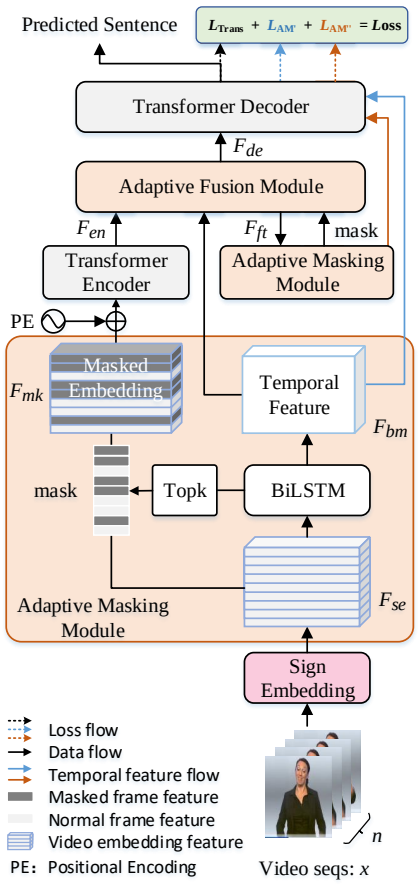


Fig. 3. The architecture of adaptive masking (AM) module. F_{se}, F_{bm}, F_{mk}, F_{en}, F_{ft}, and $F_{de} \in \mathbb{R}^{n \times d_m}$ denote sign embedding, temporal feature, masked embedding, encoder output feature, adaptive fusion feature, and decoder input feature, respectively. Three loss terms are adopted during training: the translation loss $\mathcal{L}_{Trans}$ (black dotted flow) enhances the synchronization effect between translated and predicted sentences, and the AM loss $\mathcal{L}_{AM'}$ (blue dotted flow) and $\mathcal{L}_{AM''}$ (orange dotted flow) can optimize the dropout effect.

B. Adaptive Masking

As shown in Figure 3, the SL video features $F_{se} \in \mathbb{R}^{n \times d_m}$ extracted by sign embedding need to be processed by the adaptive masking (AM) module before being passed to the encoder. The purpose is to generate a continuous temporal feature $F_{bm} \in \mathbb{R}^{n \times d_m}$ while generating a special mask for F_{se}

to drop out the frame feature representations. In this section, we first introduce the temporal feature extraction layer of the AM module and then elaborate the details of acquiring important frame sequences based on the temporal feature, which can adaptively drop out temporally important frame feature representations.

Since VideoMoCo requires two different convolutional neural networks (CNN) for two-stage feature extraction, if we introduce it to the Transformer model, it will greatly increase model parameters. Therefore, we replace the sign embedding layer with the pre-trained SL video feature representation F_{se}. For the temporal feature extraction layer, we note that BiLSTM has been effective in learning temporal semantic information and long-term dependencies of video sequences, and the continuous temporal feature extracted by it do not affect the feature dimension of original input. Therefore, the index numbers of masked video frames can be obtained from the probability distribution generated by it. Specifically, we first use the BiLSTM to generate continuous temporal feature for sign embedding. To generate the corresponding probability distribution for each video frame representation, the dimension of temporal feature needs to be mapped to one-dimension by a linear layer:

$$f_l = linear(BiLSTM(F_{se})), \tag{4}$$

where $f_l \in \mathbb{R}^{n\times 1}$ denotes the output feature of linear mapping. Then, we use the softmax function to obtain the probability distribution $d_l \in \mathbb{R}^n$:

$$d_l = softmax(f_l). \tag{5}$$

Finally, the original mask is updated according to the index numbers where the top k largest values in d_l so that k frame representations can be dropped to generate the mask embedding $F_{mk} \in \mathbb{R}^{n\times d_m}$:

$$I_k = topk(d_l, k), \tag{6}$$

$$F_{mk} = mask(I_k, F_{se}), \tag{7}$$

where $topk(input, k)$ is used to compute the top k largest values of the input matrix and the corresponding index numbers. $I_k \in \mathbb{R}^k$ denotes a tensor containing k index numbers. $mask(index, input)$ denotes the mask of SL video inputs. Except that the value of k needs to be set manually in advance, the updated mask can be adjusted adaptively during training to make it reach a relatively stable state.

Furthermore, there are also research works [9], [13] using BiLSTM as the backbone network for decoding and inference of SLT. Therefore, the temporal feature generated by BiLSTM is decodable. So we add a normalization layer after the BiLSTM layer and use its output as the input of AF module and decoder. This process can be formulated as:

$$F_{bm} = LayerNorm(BiLSTM(F_{se})), \tag{8}$$

where $F_{bm} \in \mathbb{R}^{n\times d_m}$ denote the temporal feature of BiLSTM.

Note that the output feature of Transformer encoder also does not destroy the feature dimension of the original input. Therefore, the AM module can be equipped between the encoder and the decoder to drop out some frame feature representations before the feature is passed to the decoder.

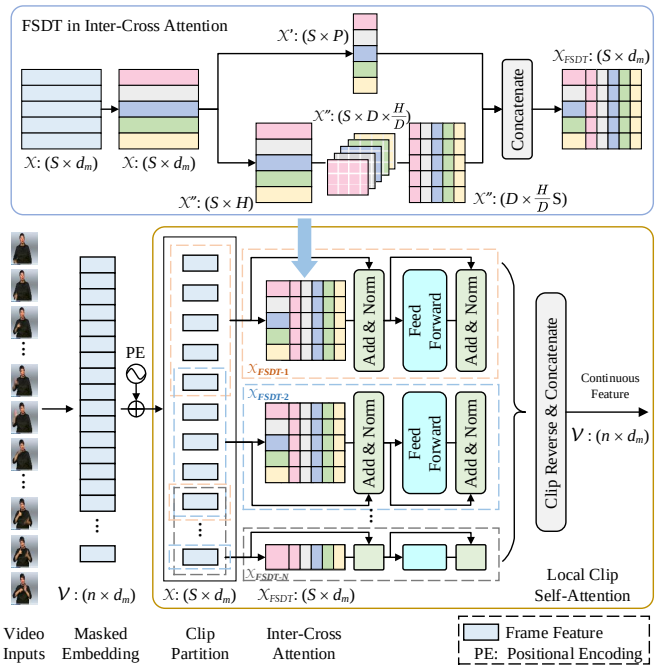


Fig. 4. The architecture of our local clip self-attention (LCSA) module. The mask embedding $\mathcal{V}$ is split into N clip features by clip partition. Each clip feature $\mathcal{X}$ contains S frames and passes into inter-cross attention module to generate a local clip enhanced feature via the feature split and dimensional transformation (FSDT) mechanism. Finally, all clip features are restored to continuous feature for subsequent global learning via the clip reverse & concatenate function.

C. Local Clip Self-Attention

As shown in Figure 4, we introduce a local clip self-attention (LCSA) to learn local semantic information, aiming to enhance the SL video features at clip-level. In this section, we first introduce a clip partition (CP) method, which can split the original continuous video feature into multiple clips. Then, we elaborate the details of inter-cross attention (ICA), which enables local sparse feature interactions within each clip to learn the interrelationships between frames. Finally, the connection of LCSA with MSA allows the model to benefit from local and global information to obtain more discriminative feature representations.

Splitting a complete video into multiple clips via non-overlapping sliding window as data input is very common in many studies [5], [8], [22], [29]. However, it is difficult to find a suitable window size for splitting video clips. This is mainly due to the fact that a single SL video usually contains multiple complex sign gestures and the frame size in a single clip needs to be set according to SL characteristics. Besides, the end-to-end SLT emphasizes overall semantic information and too sparse clips will destroy the semantic coherence. Therefore, in this work, our CP method differs from the common CP (CCP) [5], [8], [22] and GLR [29] in that we fuse several keyframes from neighbor clips with the last frame of short-range clips and replace long-range clips with original continuous video. Specifically, given the video representation F_{mk} from sign embedding, we split it into N clips with

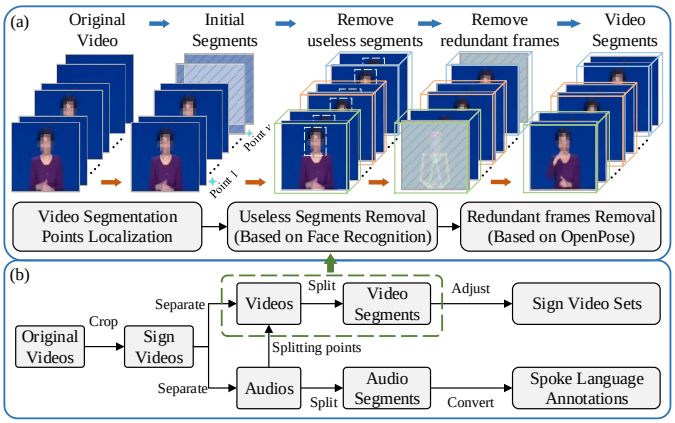


Fig. 5. An overview of CSL-FocusOn data collection and annotation. (a) Detailed video segmentation process. (b) The key process of dataset collection.

sliding window size w and stride size s. Then, we extend m keyframe feature representations for each clip. The first and last clips are extended backward and forward, respectively, and the extension frame size is m and the stride size is t. The clips in the middle are extended to both sides with extension frame size $m/2$ and stride size t. So, the frame size of each clip is $S = w$, which contains m extended frames. By supplementing each clip with sparse keyframes from neighbor clips, it not only increases neighbor information, but also alleviates the semantic ambiguity problems caused by inappropriate clip size and boundary.

The standard MSA allows spatial mixing of entire spatial locations of the input sequence to learn global contextual information. However, there is a lack of information interaction between sequence features. To solve this issue, we introduce an ICA based on MSA to split the full-size feature into two sparse-size features by simply decomposing the feature axis, and then generates an information-interactive enhanced feature after dimensional transformation and concatenation. As shown in Figure 4, ICA is composed of feature split and dimensional transformation (FSDT) mechanism and MSA. Specifically, we consider the last dimension d_m of each clip feature $\mathcal{X} \in \mathbb{R}^{S\times d_m}$ as the feature axis, where S denotes the clip frame size. Next, we split $\mathcal{X}$ into $\mathcal{X}' \in \mathbb{R}^{S\times P}$ and $\mathcal{X}'' \in \mathbb{R}^{S\times H}$ on the feature axis d_m, and then the dimension of $\mathcal{X}''$ is decomposed to $\mathcal{X}'' \in \mathbb{R}^{S\times D\times \frac{H}{D}}$ on the feature axis H, where $d_m = (P + H), D = S$. D denotes sparse feature size. Then, the dimension of $\mathcal{X}''$ is transformed to $\mathcal{X}'' \in \mathbb{R}^{D\times \frac{H}{D}S}$. Finally, we use the concatenation function for $\mathcal{X}'$ and $\mathcal{X}''$ to get the final $\mathcal{X}_{FSDT} \in \mathbb{R}^{S\times d_m}$, where $d_m = (P + \frac{H}{D}S)$. The FSDT process from $\mathcal{X}$ to $\mathcal{X}_{FSDT}$ can be formulated as:

$$\mathcal{X}(S, d_m) \rightarrow \mathcal{X}'(S, P)\ \&\ \mathcal{X}''(S, H), \tag{9}$$

$$\mathcal{X}''(S, H) \rightarrow \mathcal{X}''(S, D, \frac{H}{D}) \rightarrow \mathcal{X}''(D, \frac{H}{D} \times S), \tag{10}$$

$$\mathcal{X}_{FSDT}(S, P + \frac{H}{D} \times S) = Concat(\mathcal{X}', \mathcal{X}''), \tag{11}$$

$$\mathcal{X}_{FSDT}(S, d_m) = \mathcal{X}_{FSDT}(S, P + \frac{H}{D} \times S). \tag{12}$$

Next, we pass the transformed clip feature $\mathcal{X}''$ into the MSA. By applying ICA to each clip, we implement local feature enhancement within each clip.

Furthermore, after reversing and concatenating the clip features, we further combine LCSA and MSA to form the Transformer encoder in order to learn local and global information. The whole process can be formulated as:

$$F_{en} = MSA(CRC(LCSA(CP(PE(F_{mk}))))), \tag{13}$$

where $CRC(\cdot)$ denotes clip reverse and concatenate function, and $F_{en} \in \mathbb{R}^{n\times d_m}$ denotes the encoder output feature.

D. Adaptive Fusion

To make the model learn the temporal semantic information and long-term dependencies of SL video sequences, Yin et al. [9] combined the BiLSTM with Transformer to achieve better translation results. However, using some simple methods for these two networks, such as network connection, feature concatenation [29], matrix-vector addition [8], etc., does not lead to better SLT improvements without using glosses in training. Furthermore, considering that both the temporal feature of AM module and the output feature of encoder can be decoded, and that the two features are different in terms of temporal and spatial learning. Therefore, for the model to better learn the knowledge of temporal and spatial information simultaneously, a novel idea is to let the model measuring the important feature information of both networks by itself and fuse them together. So, we introduce an adaptive fusion (AF) module based on GRF [31] and AFA [32]. Considering the gate mechanism, computational complexity and extensibility of GRF, we simplify its memory mechanism and modify it to suit the needs of spatio-temporal feature fusion.

As shown in Figure 3, we first pass the F_{mk} generated by the AM module to the encoder and generate the enhanced spatio-temporal features F_{en} through its LCSA and MSA mechanism. Then, F_{en} is passed into the AF module for adaptive fusion together with the temporal feature F_{bm}. Moreover, unlike the adaptive gate in [32], we generate corresponding adaptive weights λ_1, λ_2 for each feature to enhance its important information. The whole process can be formulated as:

$$\lambda_1, \lambda_2 = \sigma(F_{en}W_{en} + F_{bm}W_{bm}), \tag{14}$$

$$F_{af} = \lambda_1 \odot F_{en} + \lambda_2 \odot F_{bm}, \tag{15}$$

where $W_{en}, W_{bm} \in \mathbb{R}^{d_m\times 2}$ are learnable parameters. $\odot$ and σ denote the element-wise multiplication and the sigmoid function, respectively. The $\lambda_1, \lambda_2 \in [0, 1]$ weight the expected importance of F_{en} and F_{bm} for each frame feature representation, respectively. And $F_{af} \in \mathbb{R}^{n\times d_m}$ denotes the output feature of AF module. Finally, to further learn the important knowledge in F_{bm} and F_{en}, we perform a matrix-vector addition with the F_{af} and add a normalization layer, and then pass it to the second AM module AM'' and using its masked feature $F_{de} \in \mathbb{R}^{n\times d_m}$ as the input data of Transformer Decoder:

$$F_{ft} = LayerNorm(F_{en} \oplus F_{bm} \oplus F_{af}), \tag{16}$$

$$F_{de} = AM''(F_{ft}), \tag{17}$$

where $\oplus$ denotes the matrix-vector addition. F_{ft} denotes the fused feature by AF module.

E. Inference and Joint Loss Design

Given a sentence $\mathcal{S} = \{\omega_u\}_{u=1}^{U}$ that is a sequence of U words ω_u, its word embedding after positional encoding can be formulated as:

$$\widehat{\omega}_u = WE(\omega_u) + PE(u), \tag{18}$$

where $WE(\cdot)$ denotes the word embedding layer. $\widehat{\omega}_u \in \mathbb{R}^{U\times d_m}$ denotes the embedded representation for the word $\omega_u \in \mathbb{R}^{1\times U}$ with positional encoding. The Transformer-based SLT model is an auto-regressive encoder-decoder model. Before decoding, we first add the special sentence-initial token, "$< bos >$", to the target SL sentence $\mathcal{S}$. Then, we pass the $\mathcal{Y} = \{\widehat{\omega}_u\}_{u=1}^{U}$ into the masked self-attention layer in the decoder. The Transformer decoder learns to generate one word at a step during inference by consuming previously generated words each time in an auto-regressive manner, until it generates a special sentence-final token, "$< eos >$". In inference, the original sentence level conditional probability $p(\mathcal{S}|\mathcal{V})$ will be decomposed into ordered word level conditional probability. It can be formulated as:

$$h_u = Decoder(\widehat{\omega}_{u-1}|\widehat{\omega}_{1:u-2}, F_{de}), \tag{19}$$

$$p(\mathcal{S}|\mathcal{V}) = \prod_{u=1}^{U} p(\omega_u|h_u), \tag{20}$$

which are used to calculate the cross-entropy loss as:

$$\mathcal{L} = 1 - \prod_{u=1}^{U}\sum_{m=1}^{M} p(\widehat{\omega}_u^m)p(\omega_u^m|h_u), \tag{21}$$

where $p(\widehat{\omega}_u^m)$ denotes the ground truth probability of word w_u^m at decoding step u and M is the target sentence vocabulary size [10]. Therefore, the translation loss at training can be calculated by Equation 21 and represented as $\mathcal{L}_{Trans}$.

Since the AM module does not have corresponding mask annotations for supervised training, there is a risk of unstable dropout effect when increasing the dropout threshold of video frame representations, which may cause the greater semantic ambiguity problem. Therefore, we add a weakly supervised loss term $\mathcal{L}_{AM}$ to the AM module via Equation 21. The basic principle is that we consider the temporal feature of AM module as a decodable feature representation and pass it to the Transformer decoder for decoding. We use the decoder to simulate the Discriminator of VideoMoCo. Note that although both $\mathcal{L}_{AM}$ and $\mathcal{L}_{Trans}$ use the same decoder, $\mathcal{L}_{AM}$ is only used for weakly supervised training of the AM module and is not involved in the inference of target sentences. Besides, since we add two AM modules to network, we need to set two loss terms for them, i.e., $\mathcal{L}_{AM'}$ and $\mathcal{L}_{AM''}$. $\mathcal{L}_{AM'}$ denotes the loss term of the AM module placed before the encoder and $\mathcal{L}_{AM''}$ denotes the loss term of the AM module placed between the encoder and decoder.

Like [10], we train AVRET by minimizing the joint loss term $\mathcal{L}_T$, which is a weighted sum of the translation loss $\mathcal{L}_{Trans}$ and the AM loss $\mathcal{L}_{AM'}$ and $\mathcal{L}_{AM''}$ as:

$$\mathcal{L}_T = \lambda_{Trans}\mathcal{L}_{Trans} + \lambda_{AM'}\mathcal{L}_{AM'} + \lambda_{AM''}\mathcal{L}_{AM''}, \tag{22}$$

where λ_{Trans} is a constant hyper-parameter and both $\lambda_{AM'}$ and $\lambda_{AM''}$ are hyper-parameters used to measure the importance of their corresponding loss term during training.

In summary, based on the SL video features enhanced method of the AM and LCSA module and the robust feature fusion method of AF module, we finally build the adaptive video representation enhanced Transformer.

Fig. 6. The right image is the example of original video frame. The signer is shown in the bottom left of the original video frame and the image on the left is the result of resizing it to 224×224.

F. The Proposed CSL-FocusOn Dataset

FocusOn news program contains a large amount of different corpus content and has very rich SL resources. However, it is difficult to split complete SL video segments due to the lack of fine-grained segment-caption alignment and corresponding timestamps. Therefore, in order to reduce the alignment error between SL video segments and sentences, we explore an automated data collection method and construct a news corpus-based continuous SL video dataset CSL-FocusOn. In this section, we will introduce the detailed collection process of CSL-FocusOn.

1) Data Collection and Annotation: As shown in Figure 5-(b), we share the key processes of automatic data collection and annotation. As shown in Figure 6, the rectangle position of the signer (bottom left) is fixed in the original video. We first crop the signer from the original video according to the predefined rectangle coordinates and resize it to 224×224. Then, the cropped SL video is separated from its audio. After that, we split the audio by using the voice pause points of audio host as the splitting points, and then convert audio splitting points to the video segmentation points to initially split video segments. As shown in Figure 5-(a), we design a video segmentation method to further reduce the splitting and alignment errors. Firstly, we use the face recognition model [50] to determine whether there is a signer in the segments so as to remove useless segments, and then use OpenPose [51] to locate the start and end frame positions according to the arms keypoint coordinates rule of the start and end sign gestures so as to remove redundant frames. Finally, the video segments are adjusted into video sets, while the audio segments are converted into spoken annotation sets by using the audio-to-text tool of the iFLYTEK open platform. The corpus covers 4,200 videos with a total duration of 23.11 hours, 6 different signers and contains 21,058 different words in SL sentences.

TABLE II
DETAILED AVRET FRAMEWORK AND FEATURE SIZES OF EACH KEY MODULE. B, n, U, M DENOTES BATCH SIZE, MAX FRAME SIZE IN THE BATCH, MAX WORD NUMBER IN THE BATCH, AND WORD NUMBER OF TARGET SENTENCE, RESPECTIVELY.

Module	Output Size
SL Video Inputs	$B \times n \times 224 \times 224 \times 3$
En-b0 [48] + VMC [28]	$B \times n \times 1024$
CNN-LSTM-HMMs [4]	$B \times n \times 1024$
S3D [49]	$B \times n/4 \times 1024$
Conv1D-BN1D-ReLU	-
MaxPooling1D	-
Conv1D-BN1D-ReLU	-
Linear	$B \times n \times 512$
Adaptive Masking (AM')	$B \times n \times 512$
Transformer-Encoder-LCSA	$B \times n \times 512$
Transformer-Encoder-MSA	$B \times n \times 512$
Adaptive Fusion	$B \times n \times 512$
Adaptive Masking (AM")	$B \times n \times 512$
Translation Inputs	$B \times U$
Word Embedding	$B \times U \times 512$
Transformer-Decoder-MSA	$B \times U \times 512$
Encoder-Decoder-MSA	$B \times U \times 512$
Fully Connection	$B \times U \times M$

The SL sentence annotations have been automatically transcribed and manually verified. Furthermore, we also generate corresponding multi-view images (e.g., hands, face, and body keypoints) from the original videos via OpenPose.

2) Privacy Considerations: Since the CSL-FocusOn dataset was collected from a public news program, we cannot make it public without obtaining permission, considering the content of corpus and the privacy issues of signers. We can share some information such as video download websites, automatic dataset collection method, etc. Therefore, this dataset is mainly used to perform academic research and to evaluate the effectiveness of our method.

IV. EXPERIMENT

A. Dataset and Metrics

1) Datasets: We evaluate our method on three datasets, including CSL-FocusOn, PHOENIX14T [20] and CSL-Daily [12]. These datasets all contain annotations of SL translation sentences.

PHOENIX14T is a large vocabulary and continuous German SL corpus, which is the primary benchmark for CSLR and SLT in recent years. It is collected from a German weather forecast broadcast PHOENIX, with 9 different signers. The corpus contains 8257 sample pairs, of which the train set, development set, and test set containing 7096, 519, and 642 samples, respectively. The gloss annotations have 1066 different sign glosses and the vocabulary size is 2887 for German translation sentences.

CSL-Daily is a new large-scale Chinese SL corpus that covers a wide range of scenarios, including daily life, medical care, weather, and so on. The corpus content is mainly collected from some Chinese SL textbooks, test materials and Chinese corpora [12]. It contains 2000 sign glosses, 2343 vocabulary size, and 20654 sample pairs, of which the train set, development set, and test set containing 18401, 1077, and 1176 samples, respectively.

2) Metrics: To measure the SLT performance of our method, we adopt the BLEU [52] (n-grams ranging from 1 to 4) and ROUGE [53] scores, which are commonly used in machine translation and SLT. Considering BLEU-4 can better measure the integrity of generated sentences, we use it as the performance metric in ablation study.

3) Task Details: In this paper, we mainly concentrate on the end-to-end SLT without using glosses in training. Note that we do not pre-train the AVRET network on the CSLR task or other datasets to pre-train SL video features. We use pre-trained visual features instead of sign embedding for subsequent SLT task. For fair comparison with baseline models SLTT-S2T [10] and BN-TIN-Transf [12], we use CNN-LSTM-HMMs [4] and S3D [49] to extract gloss-based SL video features on PHOENIX14T and CSL-Daily, respectively. To evaluate the SLT effect of gloss-free features, the gloss-free features of all experimental datasets are extracted by Efficientnet-b0 [48] and VideoMoCo [28] (En-b0+VMC). Specifically, we use Efficientnet-b0 instead of the Encoder in VideoMoCo and pre-train the Discriminator by VideoMoCo framework. Then, the Efficientnet-b0 fine-tuned by [24] is used to extract SL video features. Table II shows the output feature size of different feature extraction networks.

B. Implementation Details

1) Network Details: Table II shows the detailed AVRET framework and feature sizes of each key module. We used different networks to extract SL video features. The temporal conv blocks that replace sign embedding follow [12] when using S3D, which consist of two Conv1D-BN1D-ReLU-MaxPooling1D layers with 1 stride size and 3 kernel size, and the output size is $(B \times n/4 \times 512)$ [19]. The Transformer encoder and decoder both have 512 hidden sizes, 2048 feed forward sizes, 8 heads, 3 layers, and 0.1 dropout rate. For the LCSA module, the clip sliding window size w is 16, stride size s is 3, extension frame size m is 4, extension stride size t is 13. (S, P, H, D) is set to (16, 128, 384, 24) and d_m is 512. For the AM module, we set the value ranging of frame dropout threshold $\{k', k''\}$ from 0 to 10. The hidden state size of BiLSTM in AM module is 512. All module components of the network are implemented in PyTorch.

2) Training: In all experiments, we set the batch size to 32 and set adamW [54] optimizer with an initial learning rate of $10^{-3}(\beta_1 = 0.9, \beta_2 = 0.998)$ and a weight decay of 10^{-3} [10]. We employ the plateau learning rate scheduling to decrease the learning rate, where the decrease factor is 0.7 and the minimum learning rate is 10^{-7} [10]. Besides, our two AM modules in AVRET are only used during training. For the model pre-training, we perform respective visual-language pre-training (VLP) [24] tasks on our AVRET network on the training sets of three SL datasets. And inspired by VLP, we also conduct visual mask pre-training (VMP) tasks, aiming to

TABLE III
EVALUATION OF DIFFERENT VIDEO FRAME REPRESENTATIONS DROPOUT METHODS ON PHOENIX14T AND CSL-FOCUSON. 'AM' DENOTES ADAPTIVE MASKING, AND 'RD' DENOTES RANDOM DROPOUT.

Method	PHOENIX14T		CSL-FocusOn	
AM $\{k', k''\}$	Dev	Test	Dev	Test
{0, 0}	22.64	22.95	6.68	7.06
{0, 2}	23.24	23.06	7.43	7.25
{0, 4}	23.51	23.20	7.84	7.53
{0, 6}	22.14	22.42	8.14	7.81
{0, 10}	20.54	20.82	6.11	5.82
{2, 0}	23.18	23.55	7.71	7.18
{2, 2}	23.81	24.22	7.94	7.51
{2, 4}	**24.31**	**24.84**	8.24	8.02
{2, 6}	22.73	22.98	**8.81**	**8.37**
{4, 0}	23.22	23.65	7.26	7.49
{4, 2}	23.41	23.12	7.65	7.12
{4, 4}	21.18	21.57	6.55	6.17
RD $\{r', r''\}$	Dev	Test	Dev	Test
{0, 4}	22.34	22.62	6.77	6.93
{2, 0}	22.01	21.85	7.51	7.27
{2, 4}	22.58	22.81	6.34	6.02
{4, 2}	23.31	22.75	6.03	5.88

TABLE IV
EVALUATION OF DIFFERENT ATTENTION MODULE IN AVRET ENCODER ON PHOENIX14T AND CSL-FOCUSON. 'MSA' DENOTES MASKED SELF-ATTENTION, 'ICA' DENOTES INTER-CROSS ATTENTION, 'CP' DENOTES OUR CLIP PARTITION, 'CCP' DENOTES COMMON CLIP PARTITION, AND 'LCSA' DENOTES LOCAL CLIP SELF-ATTENTION.

Attention in Encoder	PHOENIX14T		CSL-FocusOn	
	Dev	Test	Dev	Test
w/ MSA	22.41	22.83	6.91	7.18
MSA w/ ICA	21.63	22.14	6.16	5.70
CCP+MSA	22.72	23.15	7.14	6.58
CP+MSA	23.11	23.65	7.42	6.94
CP+LCSA	23.54	24.17	7.94	7.45
CP+MSA+MSA	23.74	24.32	8.26	7.73
CP+LCSA+MSA	**24.31**	**24.84**	**8.81**	**8.37**

TABLE V
EVALUATION OF DIFFERENT FEATURE FUSION METHOD FOR ADAPTIVE FUSION MODULE ON PHOENIX14T AND CSL-FOCUSON. (F_* DENOTES THE DIFFERENT FEATURES TO BE FUSED. '**C**' REPRESENTS CONCATENATION(·): SIMPLE FEATURE CONCATENATION OPERATION.)

Fusion method	PHOENIX14T		CSL-FocusOn	
	Dev	Test	Dev	Test
w/ F_{en}	21.64	22.36	6.58	6.13
w/ F_{af}	22.87	23.26	7.15	6.70
F_{en} + F_{bm}	23.54	23.91	8.17	7.33
F_{en} + F_{bm} + F_{af}	**24.31**	**24.84**	**8.81**	**8.37**
C(F_{en}, F_{bm})	23.21	22.88	7.88	7.31
C(F_{en}, F_{bm}, F_{af})	23.75	24.20	8.32	7.82
GRF [31]	23.94	24.58	8.51	8.07

perform pre-trained SLT tasks by using randomly masked SL videos as input to AVRET. The mask frame size is set to 5. Note that we only perform model pre-training when using the gloss-free features of the three SL datasets to compare with the gloss-free end-to-end SLT methods. All SLT experiments are run on RTX 3090 GPU.

3) Decoding: In our experiments, we apply the greedy search to decode SL sentences during the training and validation. For the inference step, we use the beam search strategy and the length penalty [55] to decode the test set.

C. Ablation Study

To evaluate the effectiveness of our method, the end-to-end SLT experiments in this subsection are conducted on PHOENIX14T and CSL-FocusOn, using the evaluation metric of BLEU-4. Except for Section IV-C6, the experimental results of PHOENIX14T and CSL-FocusOn in ablation analysis are obtained based on gloss-based and gloss-free features, respectively, and are not pre-train the network by VMP and VLP.

1) Analysis of Adaptive Masking: In Table III, we analyse the effectiveness of different video frame representations dropout methods on PHOENIX14T and CSL-FocusOn, including two AM modules in AVRET and random dropout. The experiment variable is the AM dropout threshold $\{k', k''\}$ and the random dropout threshold $\{r', r''\}$, where k' and r' denote the dropout threshold before feature is passed to the encoder, k'' and r'' denote the dropout threshold before feature is passed to the decoder. In terms of the range of $\{k', k''\}$, we set it from 0 to 10 depending on the average frame size contained in all videos on both datasets, where 0 means that the AM module is not used. We can see that the BLEU-4 score of PHOENIX14T maintains an increasing trend when the value of $\{k', k''\}$ increases from 0 to 4 and achieves the best SLT results at $\{2, 4\}$. However, when we further increase the value of k', the BLEU-4 score starts to decrease significantly. And when $\{k', k''\}$ is set to$\{0, 10\}$ and $\{4, 4\}$, the SLT effect is poor and is directly weaker than $\{0, 0\}$. The experimental results show that both AM modules are crucial to the choice of the dropout threshold. A suitable value of $\{k', k''\}$ can effectively improve the performance of SLT, while too large value can drop out too many important information, which can seriously affect the final SLT effect.

Moreover, we also conduct the evaluation of AM modules on CSL-FocusOn. From the experimental results, the trend of the value of $\{k', k''\}$ on SLT is consistent with PHOENIX14T. However, CSL-FocusOn achieves the highest BLEU-4 score at $\{2, 6\}$. We also find that the best $\{k', k''\}$ on CSL-Daily is $\{2, 2\}$. Our analysis of these datasets shows that the reason for this phenomenon may be related to the frame size of different datasets. On the one hand, the average frame size for a single video of CSL-FocusOn is higher than PHOENIX14T and CSL-Daily. On the other hand, CSL-FocusOn have 30 frames per second and one sign gesture usually contains 15 to 20 frames, while PHOENIX14T stays at 12 to 16 frames and CSL-Daily stays at 6 to 12 frames. It also indicates that dropout thresholds are affected by average frames size and

TABLE VI
EVALUATION OF DIFFERENT LOSS WEIGHT OF LOSS FUNCTION ON PHOENIX14T AND CSL-FOCUSON.

Loss Weight			PHOENIX14T		CSL-FocusOn	
$\lambda_{AM'}$	$\lambda_{AM''}$	λ_{Trans}	Dev	Test	Dev	Test
0	0	1.0	22.64	22.95	6.68	7.06
1.0	1.0	1.0	23.85	24.01	8.47	7.84
1.0	2.0	1.0	24.07	24.54	8.64	8.13
1.0	3.0	1.0	23.34	23.92	7.96	7.32
2.0	1.0	1.0	**24.31**	**24.84**	**8.81**	**8.37**
2.0	2.0	1.0	23.65	24.18	8.53	8.11
2.0	3.0	1.0	24.03	23.61	8.31	7.82
3.0	1.0	1.0	23.28	23.44	8.04	7.62
3.0	2.0	1.0	22.81	23.04	7.21	7.45
3.0	3.0	1.0	22.63	23.14	7.54	7.22
5.0	1.0	1.0	22.76	22.91	7.72	7.51
10.0	2.0	1.0	22.14	21.85	7.03	7.14
2.0	1.0	3.0	22.82	23.17	8.10	7.81
2.0	1.0	5.0	22.24	23.41	7.46	7.25
2.0	1.0	10.0	21.53	21.88	6.87	6.62

TABLE VII
EVALUATION OF DIFFERENT OPTIMIZER ON PHOENIX14T AND CSL-FOCUSON.

Optimizer	PHOENIX14T		CSL-FocusOn	
	Dev	Test	Dev	Test
Adadelta [56]	24.03	23.83	8.43	8.03
Adam [57]	24.17	24.05	8.62	8.28
AdamW [54]	**24.31**	**24.84**	**8.81**	**8.37**

TABLE VIII
EVALUATION OF DIFFERENT SL VIDEO FEATURES AND PRE-TRAINING ON PHOENIX14T AND CSL-FOCUSON. (EN-B0+VMC DENOTES EFFICIENTNET-B0 AND VIDEOMOCO)

Pre-trained Feature	Pre-training		PHOENIX14T		CSL-FocusOn	
	VMP	VLP	Dev	Test	Dev	Test
En-b0+VMC	✗	✗	19.12	19.41	8.81	8.37
	✓	✗	21.57	21.25	9.13	9.28
	✗	✓	22.49	22.17	**9.64**	**9.81**
TSPNet [22]	✗	✓	17.34	17.62	8.05	7.95
SLTT-S2T [10]	✗	✗	**24.31**	**24.84**	-	-

need to be adjusted according to different datasets.

We also compared the effect of random dropout on both datasets. The dropout value that performs better in AM is selected. It can be found that the effect of random dropout is relatively weak and unstable. This is because the random dropout method is unlearnable and predefined dropout index. It may break the local semantic consistency.

2) Analysis of Local Clip Self-Attention: In our proposed LCSA, we introduce a clip partition (CP) method and inter-cross attention (ICA) for clip-level inputs to enhance the information interaction within each clip. To verify the effectiveness of clip-level inputs and the LCSA, different combinations of attention modules in AVRET encoder are evaluated in Table IV. For notation, the baseline model is to use masked self-attention (MSA) on the original inputs. **MSA w/ ICA** means MSA with ICA. **CCP+MSA** means use the common clip partition (CCP) method and MSA on clip-level inputs. **CP+MSA** and **CP+LCSA** mean to the use of MSA and LCSA respectively for each clip on clip-level inputs, and the clip-level features are concatenated into continuous features and then used as the output of the encoder. **CP+MSA+MSA** and **CP+LCSA+MSA** mean to adding the global information learning by connecting MSA after the original attention combinations.

As shown in the Table IV, the MSA equipped with ICA has a certain performance loss when using the original inputs. This is because when processing long SL videos with multiple sign gestures, ICA crosses features together and affects the discriminative information between different gestures. Furthermore, after processing the original inputs into clip-level, the combinations with different attentions show different effects. When only clip-level inputs and CCP are performed, the improvement of MSA is relatively weaker, while LCSA has a better performance and brings a larger improvement compared to MSA w/ ICA. When clip-level attention combinations are additionally connected to the MSA, the performance of SLT is further improved because the encoder increases the learning capability of continuous spatio-temporal features and global information. Overall, our LCSA performs better than other methods and has better performance when local and global are combined.

3) Analysis of Adaptive Fusion: To verify the effectiveness of AF module, we conduct comparison experiments of different feature fusion method. For notation in Table V, F_{en} denotes the output feature of Transformer encoder. F_{bm} denotes the temporal feature of BiLSTM in AM module. F_{af} denotes the adaptive fusion feature of F_{en} and F_{bm}. On the test set of PHOENIX14T, we can see that the second and third fusion methods bring improvements of +0.90 BLEU and +1.55 BLEU, respectively. It also indicates that incorporating F_{bm} in F_{en} can effectively improve SLT performance. However, the BLEU score that using F_{af} is lower than the score of the addition of F_{en} and F_{bm}. It is mainly because adaptive fusion is an approach where knowledge enhancement and weakening coexist. During training, some important knowledge of F_{en} and F_{bm} may be weakened, which affects the final SLT effect. Therefore, we fuse the three feature representations of F_{en}, F_{bm} and F_{af}, so that the fused features of F_{en} and F_{bm} can be enhanced by F_{af} while mitigating the weakening effect of F_{af}. Thus, further performance improvement is achieved. Moreover, this fusion approach is still valid on CSL-FocusOn and achieves a +2.24 BLEU improvement on the test set. We also perform experiments on simple feature concatenation methods and GRF. As the experimental results in the bottom of Table V show, our fusion method works better. It is interesting to note that our AF module is simplified and adapted from GRF, but performs slightly better than it. The reason may be that the final feature fusion operation of GRF is achieved by concatenation, and the experimental results of

TABLE IX
COMPARISON OF EVALUATION RESULTS FOR END-TO-END SLT ON PHOENIX14T AND CSL-DAILY. '*' DENOTES THAT THE METHOD USES GLOSS ANNOTATIONS IN TRAINING

Method	PHOENIX14T									
	Dev					Test				
Gloss-free	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
RNN+Luong [20]	32.60	31.58	18.98	13.22	10.00	30.70	29.86	17.52	11.96	9.00
RNN+Bahdanau [20]	31.80	31.87	19.11	13.16	9.94	31.80	32.24	19.03	12.83	9.58
Multitask-T [21]	-	-	-	-	-	36.28	37.22	23.88	17.08	13.25
TSPNet-Joint [22]	-	-	-	-	-	34.96	36.10	23.12	16.88	13.41
SimulSLT [15]	36.38	36.21	23.88	17.41	13.57	35.88	37.01	24.70	17.98	14.10
GASLT [23]	-	-	-	-	-	39.86	39.07	26.74	21.86	15.74
STMC-T [13]	39.76	40.73	29.42	22.61	18.21	39.82	41.05	29.92	23.01	18.47
Multi-channel [11]	44.59	-	-	-	19.51	43.57	-	-	-	18.51
GFSLT-VLP [24]	43.72	44.08	33.56	26.74	22.12	42.49	43.71	33.18	26.11	21.44
AVRET-VLP (ours)	**47.62**	**46.98**	**34.97**	**27.48**	**22.49**	**46.61**	**46.80**	**34.73**	**27.22**	**22.17**
Gloss-based	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
SLTT-S2T [10]	-	45.54	32.60	25.30	20.69	-	45.34	32.31	24.83	20.17
PiSLTRc-S2T [14]	47.89	46.51	33.78	26.78	21.48	48.13	46.22	33.56	26.04	21.29
MMTLB [19]	45.84	47.31	33.64	25.83	20.76	45.93	47.40	34.30	26.47	21.44
HST-GNN [16]	-	46.10	33.40	27.50	22.60	-	45.20	34.70	27.10	22.30
XmDA [18]	48.05	-	-	-	22.90	47.33	46.84	34.69	27.50	22.79
ConSLT* [17]	-	-	-	-	-	-	48.73	36.53	29.03	24.00
SignBT* [12]	**50.29**	**51.11**	**37.90**	**29.80**	**24.45**	49.54	50.80	37.75	29.72	24.32
AVRET (ours)	50.24	50.32	37.83	28.97	24.31	**50.86**	**51.38**	**38.45**	**30.39**	**24.84**
Method	CSL-Daily									
	Dev					Test				
Gloss-free	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
GASLT [23]	-	-	-	-	-	20.35	19.90	9.94	5.98	4.07
GFSLT-VLP [24]	36.70	39.20	25.02	16.35	11.07	36.44	39.37	24.93	16.26	11.00
AVRET (ours)	36.11	38.40	24.28	15.83	10.86	36.64	39.73	25.27	16.41	11.28
AVRET-VLP (ours)	**37.22**	**40.22**	**26.33**	**17.27**	**12.48**	**37.63**	**40.43**	**26.52**	**17.61**	**12.63**
Gloss-based	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
BN-TIN-Transf [12]	37.29	40.66	26.56	18.06	12.73	37.67	40.74	26.96	18.48	13.19
AVRET (ours)	46.34	47.31	34.67	23.80	17.84	47.26	48.78	35.11	24.55	18.22
SignBT* [12]	49.49	51.46	37.23	27.51	20.80	49.31	51.42	37.26	27.76	21.34
XmDA* [18]	49.36	-	-	-	21.69	49.34	50.92	38.21	28.31	21.58
MMTLB* [19]	**53.38**	**53.81**	**40.84**	**31.29**	**24.42**	**53.25**	**53.31**	**40.41**	**30.87**	**23.92**

C(F_{en}, F_{bm}, F_{af}) in Table V also confirm that the effect of concatenation method is weaker than matrix-vector addition.

4) Analysis of Joint Loss Design: In Section III-E, we constrain two AM modules by introducing weakly supervised constrained loss term $\mathcal{L}_{AM'}$ and $\mathcal{L}_{AM''}$. Therefore, we further conduct comparison experiments to explore and analyse the effectiveness of $\mathcal{L}_{AM'}$ and $\mathcal{L}_{AM''}$ with different loss weights in training. As shown in Table VI, we set the value of loss weight λ_{Trans} to a constant value of 1.0, which corresponds to λ_T in SLTT [10]. This is because $\mathcal{L}_{AM'}$ and $\mathcal{L}_{AM''}$ are not involved in the inference of the target sentence, and we must use $\mathcal{L}_{Trans}$ to supervise the inference of the target sentence to constrain it. When $\lambda_{AM'}$ or $\lambda_{AM''}$ is 0 indicates that the corresponding loss function is not used and the constraint effect cannot be worked, so the translation effect on PHOENIX14T is not ideal. When both $\lambda_{AM'}$ and $\lambda_{AM''}$ are set to 1.0, the translation performance of the test set is significantly improved. However, when the value of both loss weights are increased, the translation effect acquires further improvement at the beginning, but decreases gradually afterwards. It demonstrates that both loss weights need a suitable value in order to achieve the best training effect. Besides, as we increase λ_{Trans}, the translation effect also decreases gradually. And our experiments on CSL-FocusOn also verified the effectiveness of $\mathcal{L}_{AM'}$ and $\mathcal{L}_{AM''}$. Therefore, in terms of the experimental results on both datasets, the use of weakly supervised constraints with appropriate loss weight for the AM modules is crucial for our method.

5) Analysis of different optimizers: While the Adam [57] optimizer is widely used in CSLR and SLT tasks, in some

TABLE X
COMPARISON OF EVALUATION RESULTS FOR END-TO-END SLT ON CSL-FOCUSON.

Method	Dev					Test				
Gloss-free	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	BLEU-1	BLEU-2	BLEU-3	BLEU-4
RNN+Luong [20]	29.12	28.87	16.57	10.52	6.15	28.03	28.37	16.17	9.81	5.84
RNN+Bahdanau [20]	28.85	29.15	16.82	10.64	6.42	28.41	28.33	16.25	10.06	6.28
SLTT-S2T [10]	30.58	30.23	17.91	11.51	7.33	30.05	29.87	17.45	11.18	7.11
Multi-channel [11]	29.81	29.53	17.17	11.09	7.64	29.88	30.02	17.73	11.39	7.42
GASLT [23]	30.91	31.11	18.52	12.19	8.23	30.38	30.32	18.03	11.52	7.84
TSPNet-Joint [22]	31.14	30.90	18.31	12.11	8.05	30.17	30.52	18.25	11.78	7.95
AVRET (ours)	31.28	31.14	18.75	12.42	8.81	31.08	31.35	18.67	12.24	8.37
GFSLT-VLP [24]	31.21	32.41	19.21	13.47	9.34	30.88	32.18	19.03	13.31	9.12
AVRET-VLP (ours)	**32.18**	**32.74**	**19.47**	**13.65**	**9.64**	**33.02**	**33.15**	**19.88**	**14.12**	**9.81**

recent unsupervised learning research, we note that [27] achieves excellent training results using the AdamW [54] optimizer. Therefore, we further explored the effectiveness of different optimizers in our approach. The experimental results are shown in Table VII, where different optimizers can deliver different performance improvements. On PHOENIX14T, the AdamW optimizer shows the relatively best translation performance on both development and test sets and makes the BLEU score of the test set exceed the development set. On CSL-FocusOn, AdamW also outperforms the other optimizers. Adadelta [56] is not as effective as other optimizers. It shows that AdamW is more suitable for our method than other optimizers.

6) Analysis of SL video features and pre-training: To evaluate the effectiveness of gloss-based and gloss-free SL video features as well as model pre-training, we conduct end-to-end SLT experiments by using different SL video features and pre-training methods, and the experimental results are shown in Table VIII. For the gloss-free features, we perform self-supervised pre-training via Efficientnet-b0 [48] and VideoMoCo [28] (En-b0+VMC), and fine-tuning on [24] to extract the SL features of PHOENIX14T and CSL-FocusOn, respectively, and then use VMP and VLP for model pre-training on AVRET. As shown in Table VIII, VLP can bring better performance improvement on both datasets. This is mainly because the key to both AVRET and VMP is to mask visual features, while VLP is to mask SL sentences. Therefore, the performance improvement of VMP relative to VLP is limited. For comparison with GASLT [23], we also use TSPNet [22] to extract SL video features. The results show that our method performs better than GASLT. Besides, to compare with the baseline model SLTT-S2T [10] on PHOENIX14T, we use gloss-based features for SLT and show a significant performance gain. This also verifies that glosses can not only improve the SLT effect during training, but also generate more discriminative SL features.

7) Limitation: The selected dropout threshold of AM module is related to the average frame size in the different datasets, and the size and boundaries of SL video clips also need to be set manually. If the appropriate dropout threshold, window size and stride cannot be set, it will affect the final SLT performance.

D. Comparisons with State-of-the-art Methods

1) Evaluation on PHOENIX14T and CSL-Daily: As shown in Table IX, we report the translation results of AVRET with existing models on the end-to-end SLT task to demonstrate the effectiveness of our method. For fair comparison, the translation results of each dataset are divided into two groups gloss-free and gloss-based according to the features used. Note that SignBT [12], ConSLT [17], XmDA [18], and MMTLB [19] use gloss annotations to participate in SLT training.

As shown in gloss-free results on PHOENIX14T, GFSLT-VLP [24] has a very significant performance improvement on Transformer-based SLT network by masked self-supervised pre-training with visual language supervision learning. So, we apply it to AVRET for pre-training and feature fine-tuning. As the results shown in Table IX, our AVRET-VLP achieves the best performance. For the gloss-based results, AVRET achieves better performance on end-to-end SLT without using glosses in training and surpasses the SLTT-S2T with BLEU-4 scores of +3.62 and +4.67 on development and test sets, respectively. Compared with the current state-of-the-art SLT methods, AVRET achieves competitive performance. It not only outperforms HST-GNN and XmDA on both sets, but also surpasses SignBT on the test set. However, we are slightly below SignBT in the development set.

In Table IX, we also compare AVRET and AVRET-VLP with some SLT methods on CSL-Daily. For the gloss-free results, AVRET is weaker than GFSLT-VLP on the development set, while it performs better on the test set. And, benefit from the use of VLP in AVRET, we achieve even more significant performance by about +1.35 on test set. For the gloss-based results, BN-TIN-Transf [12] is the baseline model that does not use sign back-translation during training. Our method is weaker than SignBT and MMTLB, although it has a larger improvement and superiority than BN-TIN-Transf. The main reason is that the performance improvement of these methods relies mainly on gloss and multi-modality pre-training, while our model starts from the most basic SLT, which is more effective when no additional annotations are involved in the training.

2) Evaluation on CSL-FocusOn: In Table X, we compare AVRET and AVRET-VLP with some SLT methods on CSL-FocusOn. The experimental results show that AVRET-VLP can achieve better SLT performance than GFSLT-VLP. However, although our method still achieves the better BLEU score, the performance improvement of the method on PHOENIX14T and CSL-Daily is more significant than on CSL-FocusOn, which is attributed to three factors. Firstly, we cannot use CNN-LSTM-HMMs to extract frame-wise SL video features since CSL does not have datasets like in [58]–[60] that can provide the same language with extra supervision. The experimental results in [10] also show that pre-trained CNN-LSTM-HMMs features are more effective than Efficientnet [48], and it can effectively improve the performance of CSLR and SLT. Secondly, although CSL-FocusOn contains fewer videos than PHOENIX14T, the average frame size and vocabulary size of a single video exceeds PHOENIX14T, and the vocabulary size even exceeds 20k. It may be due to the special scenarios of news and the irregular spoken Chinese expressions. Finally, Chinese sentences are separated differently from English or German, and the news scenarios also greatly increase the difficulty of the Chinese words separation tool. Therefore, our method is more effective in dealing with PHOENIX14T which has high-quality annotations.

V. Conclusion

In this paper, we propose an adaptive video representation enhanced Transformer (AVRET) to improve the end-to-end SLT performances without using glosses in training. To solve the weakly supervised problem of end-to-end SLT, we introduce an adaptive masking module that can drop out different video frame representations adaptively to enhance the input samples. It can be equipped not only before the Transformer encoder, but also between the encoder and the decoder. Particularly, we use SL sentences to impose the weakly supervised loss constraints on it. Furthermore, we add a local clip self-attention module in the encoder to learn clip-level video information. By connecting it with masked self-attention, it also endows the encoder to learn local and global video information. To fuse the two spatio-temporal features generated by the AM module and encoder, we introduce an adaptive fusion module that adaptively enhances the important information in both features to produce more robust feature representation. Moreover, we construct a new Chinese continuous SL video dataset based on a news program, namely CSL-FocusOn, and share its construction method. In summary, our method is very simple and flexible and can be easily applied to Transformer-based models. Extensive experiments on the three datasets also demonstrate the superiority of our method. We hope that our work will promote SLT research.

References

[1] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7361–7369.

[2] J. Huang, W. Zhou, H. Li, and W. Li, "Attention-based 3d-cnns for large-vocabulary sign language recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2822–2832, 2018.

[3] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[4] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306–2320, 2019.

[5] C. Wei, J. Zhao, W. Zhou, and H. Li, "Semantic boundary detection with reinforcement learning for continuous sign language recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1138–1149, 2020.

[6] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1459–1469.

[7] Y. Min, A. Hao, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 542–11 551.

[8] W. Yin, Y. Hou, Z. Guo, and K. Liu, "Spatial temporal enhanced network for continuous sign language recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[9] K. Yin and J. Read, "Better sign language translation with stmc-transformer," in *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020, pp. 5975–5989.

[10] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "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 (CVPR)*, 2020, pp. 10 023–10 033.

[11] N. C. Camgöz, O. Koller, S. Hadfield, and R. Bowden, "Multi-channel transformers for multi-articulatory sign language translation," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 301–319.

[12] H. Zhou, W. Zhou, W. Qi, J. Pu, and H. Li, "Improving sign language translation with monolingual data by sign back-translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1316–1325.

[13] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition and translation," *IEEE Transactions on Multimedia*, vol. 24, pp. 768–779, 2022.

[14] P. Xie, M. Zhao, and X. Hu, "Pisltrc: Position-informed sign language transformer with content-aware convolution," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[15] A. Yin, Z. Zhao, J. Liu, W. Jin, M. Zhang, X. Zeng, and X. He, "Simulslt: End-to-end simultaneous sign language translation," in *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*, 2021, pp. 4118–4127.

[16] J. Kan, K. Hu, M. Hagenbuchner, A. C. Tsoi, M. Bennamoun, and Z. Wang, "Sign language translation with hierarchical spatio-temporal graph neural network," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 3367–3376.

[17] B. Fu, P. Ye, L. Zhang, P. Yu, C. Hu, Y. Chen, and X. Shi, "Conslt: A token-level contrastive framework for sign language translation," *arXiv: 2204.04916*, 2022.

[18] J. Ye, W. Jiao, X. Wang, Z. Tu, and H. Xiong, "Cross-modality data augmentation for end-to-end sign language translation," *arXiv:2305.11096*, 2023.

[19] Y. Chen, F. Wei, X. Sun, Z. Wu, and S. Lin, "A simple multi-modality transfer learning baseline for sign language translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5120–5130.

[20] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7784–7793.

[21] A. Orbay and L. Akarun, "Neural sign language translation by learning tokenization," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 222–228.

[22] D. Li, C. Xu, X. Yu, K. Zhang, B. Swift, H. Suominen, and H. Li, "Tspnet: Hierarchical feature learning via temporal semantic pyramid for sign language translation," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020, pp. 12 034–12 045.

[23] A. Yin, T. Zhong, L. Tang, W. Jin, T. Jin, and Z. Zhao, "Gloss attention for gloss-free sign language translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 2551–2562.

[24] B. Zhou, Z. Chen, A. Clapés, J. Wan, Y. Liang, S. Escalera, Z. Lei, and D. Zhang, "Gloss-free sign language translation: Improving from visual-language pretraining," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[26] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[27] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," *arXiv: 2112.09133*, 2021.

[28] T. Pan, Y. Song, T. Yang, W. Jiang, and W. Liu, "Videomoco: Contrastive video representation learning with temporally adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 205–11 214.

[29] L. Yan, S. Ma, Q. Wang, Y. Chen, X. Zhang, A. Savakis, and D. Liu, "Video captioning using global-local representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 6642–6656, 2022.

[30] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[31] C. Fan, J. Yi, J. Tao, Z. Tian, B. Liu, and Z. Wen, "Gated recurrent fusion with joint training framework for robust end-to-end speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 198–209, 2020.

[32] W. Jiang, W. Zhou, and H. Hu, "Double-stream position learning transformer network for image captioning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7706–7718, 2022.

[33] C. Neidle, A. Thangali, and S. Sclaroff, "Challenges in development of the american sign language lexicon video dataset (asllvd) corpus," in *Proceedings 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, Language Resources and Evaluation Conference (LREC) 2012*, 2012.

[34] H. R. V. Joze and O. Koller, "Ms-asl: A large-scale data set and benchmark for understanding american sign language," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[35] A. Duarte, S. Palaskar, L. Ventura, D. Ghadiyaram, K. DeHaan, F. Metze, J. Torres, and X. Giro-i Nieto, "How2sign: a large-scale multimodal dataset for continuous american sign language," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2735–2744.

[36] D. Uthus, G. Tanzer, and M. Georg, "Youtube-asl: A large-scale, open-domain american sign language-english parallel corpus," *arXiv preprint arXiv:2306.15162*, 2023.

[37] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, "Extensions of the sign language recognition and translation corpus rwth-phoenix-weather," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, 2014, pp. 1911–1916.

[38] A. C. Duarte, "Cross-modal neural sign language translation," in *Proceedings of the 27th ACM international conference on Multimedia (ACM MM)*, 2019, pp. 1650–1654.

[39] S. Ham, K. Park, Y. Jang, Y. Oh, S. Yun, S. Yoon, C. J. Kim, H.-M. Park, and I. S. Kweon, "Ksl-guide: A large-scale korean sign language dataset including interrogative sentences for guiding the deaf and hard-of-hearing," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2021, pp. 1–8.

[40] C. Zhang, J. Su, Y. Ju, K.-M. Lam, and Q. Wang, "Efficient inductive vision transformer for oriented object detection in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–20, 2023.

[41] T. Liu, C. Zhang, K.-M. Lam, and J. Kong, "Decouple and resolve: Transformer-based models for online anomaly detection from weakly labeled videos," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 15–28, 2023.

[42] J. Huang, Y. Huang, Q. Wang, W. Yang, and H. Meng, "Self-supervised representation learning for videos by segmenting via sampling rate order prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3475–3489, 2021.

[43] S. Jenni, G. Meishvili, and P. Favaro, "Video representation learning by recognizing temporal transformations," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 425–442.

[44] L. Tao, X. Wang, and T. Yamasaki, "An improved inter-intra contrastive learning framework on self-supervised video representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5266–5280, 2022.

[45] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas, "Geometry guided convolutional neural networks for self-supervised video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5589–5597.

[46] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[47] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7464–7473.

[48] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[49] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *European conference on computer vision (ECCV)*, 2018, pp. 305–321.

[50] M. Khan, S. Chakraborty, R. Astya, and S. Khepra, "Face detection and recognition using opencv," in *International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2019, pp. 116–119.

[51] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[52] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.

[53] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Workshop on Text Summarization Branches Out*, 2004, pp. 74–81.

[54] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[55] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv: 1609.08144*, 2016.

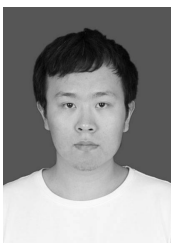
[56] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv: 1212.5701*, 2012.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[58] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2015, pp. 85–91.

[59] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3793–3802.

[60] O. Koller, H. Ney, and R. Bowden, "Read my lips: Continuous signer independent weakly supervised viseme recognition," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 281–296.



Zidong Liu received the B.S. degree from Qingdao University, Qingdao, China, in 2019. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include video processing, sign language translation, and pattern recognition.

Jiasong Wu received the Ph.D. degree in Biomedical Engineering from Southeast University, Nanjing, China, in 2012. He is a Associate Professor with the Department of Computer Science and Engineering, Southeast University, Nanjing, China. He is involved in deep learning, image captioning, audio processing, and pattern recognition.

Zeyu Shen received the M.S. degree from Southeast University, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include image analysis, pattern recognition and radiology report generation.

Xin Chen received the M.S. degree from Shandong Normal University, Shandong, China, in 2021. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include graph neural network, pattern recognition, and signal processing.

Qianyu Wu received the M.S. degree from Nanjing Tech University, Nanjing, China, in 2021. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include video analysis, image restoration, and pattern recognition.

Zhiguo Gui received the Ph.D. degree in Signal and Information Processing from North University of China, Taiyuan, China, in 2004. He is a Professor with the North University of China. He is involved in signal and information processing, image processing and recognition, medical image reconstruction, medical image processing, and pattern recognition.

Lotfi Senhadji received the M.S. and Ph.D. degrees in Signal Processing and Communication from the University of Rennes, France, in 1989 and 1993, respectively. He is a Professor with the University of Rennes (France) and the Head of French-Chinese Biomedical Information Research Center. He is involved in signal processing, time-frequency domain analysis, and pattern recognition.

Huazhong Shu received the B.S. degree in Applied Mathematics from Wuhan University, China, in 1987, and a Ph.D. degree in numerical analysis from the University of Rennes, France, in 1992. He is a Professor with the Department of Computer Science and Engineering, Southeast University, Nanjing, China. He is involved in the applied mathematics, signal processing, pattern recognition, and medical image processing.