IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2021, accepted May 2, 2021, date of publication May 10, 2021, date of current version May 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078638

Boundary-Adaptive Encoder With Attention Method for Chinese Sign Language Recognition

SHILIANG HUANG AND ZHONGFU YE
Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China
National Engineering Laboratory for Speech and Language Information Processing, Hefei 230027, China

Corresponding author: Zhongfu Ye (yezf@ustc.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant WK2350000002.

ABSTRACT The sign language signal has hierarchically related information over short and long distances. Due to the intricate temporal correlation of input sequences, Chinese sign language recognition (SLR) has a modeling challenge. The conventional encoders based on recurrent networks cannot discover and leverage the hierarchical structure of sign language well. In this paper, we propose a novel encoder-decoder method based on boundary adaptive learning for Chinese SLR. The hierarchical structure of sign language signal can be encoded by the boundary-adaptive encoder (BAE) in the proposed method. In order to improve efficiency in modeling long sign language sequences, the window attention model based on location is utilized in the decoding phase, which can generate more effective weight coefficients. Besides, we use sign language subword units to realize both isolated and continuous Chinese SLR in the same sequence learning framework in our method. Theoretical analysis and experimental results demonstrate the effectiveness and superiority of the proposed method.

INDEX TERMS Sign language recognition (SLR), boundary learning, attention, hierarchical structure.

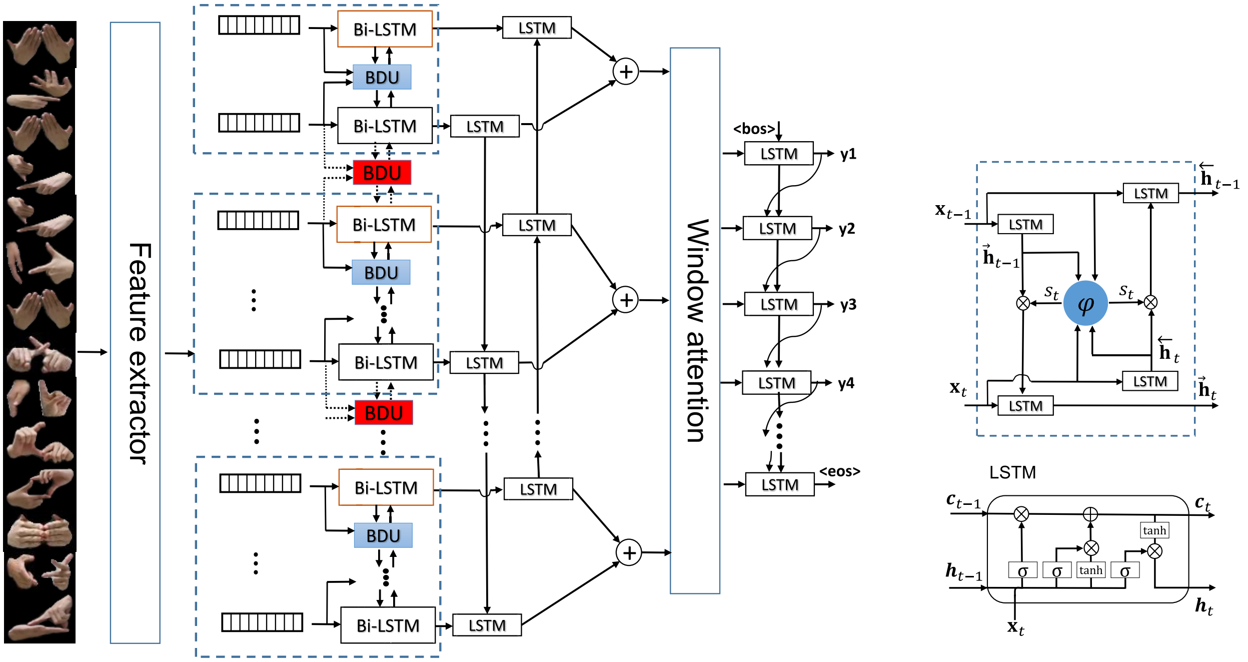
I. INTRODUCTION

Sign language is the most important way to communicate with deaf-mute people and sign language recognition (SLR) is a task dedicated to advancing this communication process with the help of computer technology. Generally, SLR can be divided into isolated SLR and continuous SLR. The recognition targets of the former are isolated sign words, and of the latter are continuous sign sentences. In recent years, many researchers have made some achievements in SLR.

Isolated SLR, the earliest research task of SLR, draws on many ideas in feature extraction and temporal modeling from action recognition [4]. For example, because of the strong ability in feature extraction, methods based on convolutional neural networks (CNN) or 3D-CNN are widely used in SLR [2], [7], [9]. And a series of methods based on recurrent neural networks (RNN) or long short-term memory (LSTM) for sequence processing are applied in isolated SLR [9], [11], [12], [52]. However, there are still some differences between these two tasks. The first one is sign language has very subtle limb changes. Therefore, there are extensive work on multimodal sign feature descriptors incorporated with depth and skeleton information [14], [11]. More importantly, sign language sequences have stricter temporal relationships compared to action recognition. Specifically, sign language signals have hierarchically related information over short and long distances. Short-distance frames contain underlying information such as sign language shape and trajectory changes. And long-distance temporal relationships contain high-level semantic information of sign language.

Continuous SLR is much more difficult due to its longer sequences and more complex correlation compared to isolated SLR. The conventional continuous SLR methods are based on the isolated SLR methods. Relying on the partitioning algorithms, the continuous sequence is segmented into several parts, which can be identified by isolated SLR methods. Finally, the above recognition results can be combined into a complete sentence using language model. However, these hard-segmentation methods face several enormous challenges. First, due to the ambiguity of the boundary between two sign language words, it is difficult to locate the boundary accurately relying on the partitioning algorithms.

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang.



(a) BAE-based model

(b) BDU and LSTM unit

FIGURE 1. The overall framework of the proposed model for SLR is illustrated in (a), and (b) shows the connection of BDU and bidirectional LSTM in the blue dotted box. The red BDU represents the boundary detection signal $s_t = 0$.

In addition, the hard-segmentation approaches ignore the semantic relationship between sign words because they consider each word independently rather than the sentence comprehensively.

In order to avoid these problems, some soft-segmentation or no-segmentation methods have been proposed in continuous SLR. Typically, the encoder-decoder framework is utilized to implicitly model the word relationship in continuous SLR. It is based on LSTM or RNN, which are very suitable for processing sequence relationship modeling. With the help of structural memory units, the long-term and short-term correlations of input sequences are encoded to semantic vectors to some extent. However, LSTM and other variants can only perform well in modeling sequences of a certain length due to its structural limitation [15]. When the sequence length exceeds this scale, the learning modeling ability of LSTM will be degraded. In addition, the flat recursive networks also have difficulty processing hierarchical sign signals. Specifically, In the process of sign language expression, there are different levels of meanings in the semantic structure. The high-level semantic information of sign language is composed of the low-level semantic information, which is concentrated in a short time between different time periods.

To address the above problems, we design a hierarchical encoder-decoder network based on Boundary-Adaptive Encoder (BAE), which can learn and encode the boundary information of the sign language signal automatically. The proposed framework is illustrated in Fig.1. Inspired by the work of [16], we use two bidirectional LSTM layers to build the BAE. The stacked network, which has BAE and decoder, model the semantic information of different levels of sign language signals effectively. Unlike [16], both future and history information of signs are utilized in our work. Besides, in the conventional encoder-decoder framework [17], it is flawed to rely on only one specific vector containing all input information for decoding. Actually, sign language input information has different effects on decoding at different times. And this problem is exacerbated by longer continuous sign language sequences. Therefore, we incorporate the window attention model into the encoder-decoder. It automatically assigns weights to the encoded vectors during decoding, thereby achieving better performance. Each decoding step corresponds to an input within a specific time position range, whereas inputs at other positions are usually less affected.

In this paper, we focus on both the isolated and continuous SLR tasks. We further divide the sign language words into more fine-grained basic subword units at a semantic level. We use subwords as a bridge to unify the two tasks of isolated and continuous SLR into one task. Then we propose a Chinese word recognition method based on subwords, which is more in line with real scenes. The main contributions of the paper are summarized as follows:

1) More fine-grained basic units for Chinese sign language are proposed. The proposed sign subword units can model the sign language more accurately than the word units. More importantly, isolated and continuous SLR can be unified into the same framework by using subword units.

2) We propose a hierarchical BAE with two bidirectional LSTM layers, which can learn the temporal boundaries

information of sign signals from history and future and encode the sign language signals at different levels.

3) We incorporate the window attention model into the sign language sequence learning network to effectively improve decoding. Under the encoder-decoder framework composed of LSTM units, the attention mechanism promotes the efficiency of long sequence modeling.

II. RELATED WORK

In this section, we briefly review three related topics: 1) isolated SLR, 2) continuous SLR, and 3) sequence to sequence learning with attention model.

A. ISOLATED SLR

Isolated SLR is the basic task of SLR. In recent years, many models have been introduced into this filed by researchers. The Hidden Markov Models (HMMs) and their variants are exploited popularly in isolated SLR [18], [19]. For some small-scale vocabulary, there are other methods, such as dynamic time warping (DTW) [20] and conditional random field (CRF) [21], support vector machine (SVM) [22], random forest (RF) [23], are also applicable to isolated SLR. With the development of deep learning technology, methods based neural networks have been favored by SLR. In recent years, replacing traditional manual features with CNN features has become the mainstream in SLR. For example, a two-stream CNN-based sign language feature extraction method is explored, in which one CNN extracts hand features and another one extracts upper body features [7]. In order to effectively integrate motion information into sign language feature, a 3D-CNN architecture has been introduced to capture the distinguishing feature along the spatial and temporal dimensions [7]. Similarly, a method based low-resolution and high-resolution 3D-CNN subnets was developed for gesture recognition, which significantly improved the classification accuracy [7]. Inspired by some temporal tasks [24], [6], LSTM is usually used to properly model the temporal relationship of sign language features. Different LSTM-based network structures have been established to identify sign language sequences after extracting sign language features, and the methods perform well on large-scale vocabulary [25], [26].

B. CONTINUOUS SLR

Continuous sign language sentences are longer and more complicated than isolated sign language words. Many methods first split the sign sentences into many short parts. For example, a DTW-HMM model using a threshold matrix for coarse segmentation and DTW for fine segmentation has achieved online continuous SLR [27]. At the same time, many methods for segmenting sign language sequences using translational motion have also been proposed [28], [29], [50]. In addition, sign action spotting [30] and alignment analysis [31], have been studied extensively for learning the complicated temporal information from continuous sign language. Generally, locating the temporal boundaries of sign language requires frame-level labels. However, continuous SLR can be considered as a weakly supervised task with class labels. Therefore, deep CNN has been integrated into the HMM framework to iteratively adjust the segmentation position to obtain more accurate alignment [32], [33]. On the other hand, inspired by the achievements of speech recognition, connectionist temporal classification (CTC) has been popularly used to learn continuous sign language sequences [3], [31], [25]. For example, in [31], CTC is first employed as the objective function for sign feature alignment proposal, and then RNN or LSTM is used to tune and optimize the SLR model.

Most of these methods require the same order of gloss between the visual content and the sign language content, whereas some end-to-end continuous SLR methods do not need to meet this limitation. Similar to our earlier exploration of isolated SLR [25], a hierarchical encoder-decoder framework was constructed in [34]. It simultaneously models the visual information and the semantic information of the sign language sequence, and avoids segmentation of the sequence. Furthermore, the deep fusion model was proposed to encode adaptively the RGB and the skeletal information of the sign language synchronously or asynchronously [10]. And the result is optimized by the decoding model. In addition, in order to effectively learn multiple levels of semantic information in sign language data, a structured feature network (SF-Net) was proposed in [13] to extract features in a structured manner. The sign information in frame level, gloss level, and sentence level has been encoded gradually into feature representations. The proposed SF-Net can perform end-to-end training without resorting to other models or pre-training.

C. SEQUENCE TO SEQUENCE LEARNING WITH ATTENTION

Because of the ability to remember historical information, sequence learning methods based on RNN or LSTM have been successfully adopted in many fields, including machine translation [17], video description [6], and image captioning and inpainting [35], [53]. Typically, authors of [17] proposed an encoder-decoder neural network model consisting of two RNNs for machine translation. This model uses an encoder network to map the input sequence to a fixed-size representation vector, and then uses the decoder network to convert the mapped vector to the target output sequence. However, it would be difficult to train the standard RNNs due to the problem of ladder dispersion of resulting long term dependencies [36]. Authors of [37] proposed an encoder-decoder neural network model based on LSTM. In the field of video description, [6] proposed an s2vt structure with a double-layer LSTM to convert a video sequence into a text sequence. Each LSTM learns the inputs both in encoding and decoding phases and shares weights between encoding and decoding. In this structure, the encoding and decoding phases differ only in time. In order to incorporate short-term and long-term time transitions into the encoding, a hierarchical deep network structure HRNE has been proposed for video captioning [8]. HRNE reduces the length of the input information stream and combines multiple consecutive inputs at a higher level, which can effectively use the temporal structure in a longer range. Inspired by the work, [10] applied the improved HRNE to sign language translation. The above methods need to specify the encoding lengths of the encoder, whereas our method is to learn the boundaries of the input sequence automatically.

Recently, to address the different importance of input information, the attention model has been developed in many fields. Some work applied the attention model to reweight spatial information, including object detection [38], fine-grained image classification [39], and image captioning and inpainting [40], [51]. The attention vector is modeled as a sequence of regions in the image, and the next region of interest is predicted by a RNN model based on the location and visual characteristics of the current region of interest. In addition, some work applied the attention model to address the different importance in time sequence [41], [42]. Our method is similar to [41]. To be specific, in the proposed encoder-decoder network, the attention model is utilized to assign different weights to the input encoded vector at different decoding moments. In addition, hierarchical attention models have also been applied for action recognition [43], SLR [5] and document classification [44].

III. PROPOSED METHOD

In this section, a novel method for both isolated SLR and continuous SLR is proposed. In the proposed network, the boundaries of input sign are detected and encoded by BAE, and window attention is applied in the encoder-decoder. Fig.1 depicts the overall framework and the details of the proposed method are introduced by follows.

A. CHINESE SIGN LANGUAGE SUBWORD

Generally, sign language words are basic units of recognition for both isolated and continuous SLR. However, the vocabulary of sign language is huge. For example, there are 5000 sign language words commonly used in Chinese sign language. Unfortunately, the huge vocabulary output space leads to the difficulty when there are not enough samples. Therefore, we define more fine-grained units sign subwords as basic units in both isolated and continuous SLR. Similar to our preliminary work [11], we manually define a Chinese sign word as a sequence of one or several sign subwords semantictly.

Chinese sign language words are composed of several relatively independent parts with different hand shapes and trajectories. We define them sign subwords. For example, the word ''kindergarten'' means ''the house of kids'' in Chinese. So the Chinese sign language word ''kindergarten'' is expressed by two subwords ''kid'' and ''house''. Similarly, the Chinese word ''station'' means ''the house of cars'', and the sign word ''station'' is composed of subwords ''car'' and ''house''. In fact, the subword ''house'' in ''kindergarten'' and ''station'' are identical. Accordingly, we can define three subwords: ''kid,'' ''car,'' and ''house.'' The ''kindergarten'' can be denoted by ''kid–house'' and the ''station'' can be denoted by ''car-house''. Fig.2 shows the instances of subwords.

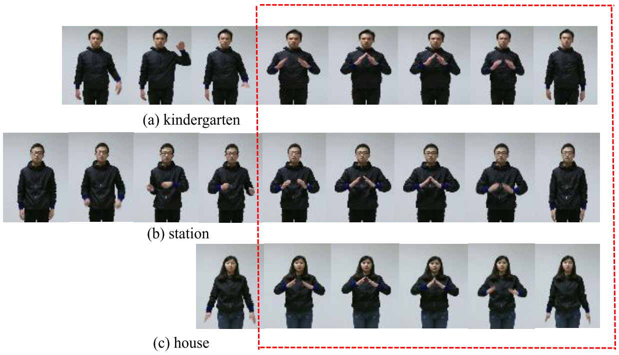


FIGURE 2. The Chinese sign language word (a) ''kindergarten'', (b) ''station'', and (c) ''house''. The actions in the red box of the three words are the same. The displayed frames are sampled from videos at intervals.

In our method, the fine-grained sign subword units are defined in the entire sign word domain. Consequently, the aim of isolated Chinese SLR becomes to model a short subword sequence and the aim of continuous Chinese SLR becomes to model a long subword sequence. In other words, we unify them into the same sequence learning framework. It's worth mentioning that we do not define how the subwords actually perform or what kind of shapes and trajectory they have. Instead, we define subwords at the semantic level.

B. BAE NETWORK

The overall network of our proposed BAE, which has two bidirectional-LSTM layers, is shown in Fig.1 (a). The first layer uses boundary detection units (BDUs) to learn the temporal boundaries of the input sign sequence. There is a BDU between each adjacent bidirectional LSTM units in the first layer to judge whether to segment the video. Then the second bidirectional LSTM layer encodes the outputs of the BDU layer. The upper of Fig.1 (b) shows the structure of one fragment which has two bidirectional LSTMs and one BDU. Crucially, it uses the memory capability of LSTM to perform automatic boundary detection on the connection state of the input sequence. Specifically, at time t, the core of the LTSM $\mathbf{c}_t$ saves the historical information entered at the previous time so that it learns related information with a certain time range. The bottom of Fig.1 (b) depicts the internal structure of LSTM. The input, forget, and output gates of LSTM combine the input $\mathbf{x}_t$ and the hidden state $\mathbf{h}_{t-1}$ at the previous time to update the current state value. The bidirectional LSTM ensures that information in both historical and future directions can be interacted. At each input moment, we use the binary boundary detection signal $s_t \in \{0, 1\}$ to choose to transfer the hidden state value and core value of the current LSTM to the next moment, or interrupt the transmission to reset. The non-artificially preset learnable BDU processes independently the sign language input feature sequence into a plurality of continuous variable-length hierarchical expression parts. It facilitates organizing the long-term and short-term time-series information of the sign language signal in the model.

Suppose a length of N input sign language feature sequence $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ and a length of M target output sequence of the entire video $(\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M)$, whose each element is a one-hot encoded subword vector. In the first Bi-LSTM layer of proposed network, the connected BDU can make corresponding changes according to the current input information and the hidden states of the LSTMs. Compared to other structures, it can adaptively learn the boundary change information of the input sign language signal. When the sign signal is estimated to undergo a large state change, the BDU can modify the connection state on the time stream, reset the two LSTM unit states, and at the same time, the hidden states of the LSTMs at the end of the time period is output as a segment. And then different blocks of similar sign language feature frames represent the sign language signals hierarchically. This structure guarantees that the subsequent boundaries of the input sign language data are not affected by the previous data. Moreover, it alleviates the shortcoming of the limited capacity of LSTM for processing long sequences, which can enable more flexible encoding.

We use [] to represent the concatenate operation and $\sigma()$ to represent the sigmoid function. Then the boundary detection signal s_t is calculated from a step function φ by:

$$s_t = \varphi\left(\mathbf{v}_s^T \cdot \left(\mathbf{W}_s\left[\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}^1, \mathbf{x}_{t-1}, \overleftarrow{\mathbf{h}}_t^1\right] + \mathbf{b}_s\right)\right) \tag{1}$$

$$\varphi(x) = \begin{cases} 1, & if\, \sigma(x) < 0.5 \\ 0, & otherwise, \end{cases} \tag{2}$$

where $\mathbf{x}_t, \mathbf{x}_{t-1}$ are the inputs and $\overrightarrow{\mathbf{h}}_{t-1}^1, \overleftarrow{\mathbf{h}}_t^1$ are the hidden states of forward LSTM at time $t-1$ and backward LSTM at time t in first layer respectively, $\mathbf{v}_s$ is a learnable vector, $\mathbf{W}_s, \mathbf{b}_s$ are learned weight parameters and bias. When $s_t = 0$, the last hidden states of the forward and backward LSTM of the segment are fed into the forward and backward LSTM of the second layer, respectively(seen in Fig.1.(b)). According to the boundary detection signal s_t, $\overrightarrow{\mathbf{h}}_{t-1}^1$, $\overleftarrow{\mathbf{h}}_t^1$ and $\overrightarrow{\mathbf{c}}_{t-1}^1$, $\overleftarrow{\mathbf{c}}_t^1$ are updated in first LSTM layer as follows:

$$\begin{aligned} \overrightarrow{\mathbf{h}}_{t-1}^1 &\leftarrow \overrightarrow{\mathbf{h}}_{t-1}^1 \cdot s_t, \quad \overrightarrow{\mathbf{c}}_{t-1}^1 \leftarrow \overrightarrow{\mathbf{c}}_{t-1}^1 \cdot s_t, \\ \overleftarrow{\mathbf{h}}_t^1 &\leftarrow \overleftarrow{\mathbf{h}}_t^1 \cdot s_t, \quad \overleftarrow{\mathbf{c}}_t^1 \leftarrow \overleftarrow{\mathbf{c}}_t^1 \cdot s_t. \end{aligned} \tag{3}$$

The adjacent input signals are jointly encoded when $s_t = 1$. When $s_t = 0$, the LSTM states are reset, the adjacent signals are disconnected from the encoding, and the segment encoding is restarted. The calculation of LSTM is as follows:

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(\mathbf{W}_i\left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_i\right) \\ \mathbf{f}_t &= \sigma\left(\mathbf{W}_f\left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_f\right) \\ \mathbf{o}_t &= = \sigma\left(\mathbf{W}_o\left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_o\right) \\ \mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * tanh\left(\mathbf{W}_c\left[\mathbf{x}_t, \mathbf{h}_{t-1}\right] + \mathbf{b}_c\right) \\ \mathbf{h}_t &= \mathbf{o}_t * tanh\left(\mathbf{c}_t\right), \end{aligned} \tag{4}$$

where $\mathbf{W}_*$ are learned weights and $\mathbf{b}_*$ are learned biases.

The first LSTM layer (BDU layer) will produce a set of variable-length outputs. These outputs can be viewed as the total expression of the content of the sign segments. And they are fed into another bidirectional LSTM layer to build a hierarchical representation of the sign language signal. Each output of the LSTM layer represents the superposition of the contents of sign segments by multiple LSTM layers. Then the encoding semantic vector $\mathbf{h}_i^e$ of the BAE output for the *i-th* sign segment is the concatenation vector of second LSTM layer outputs:

$$\mathbf{h}_i^e = \left[\overrightarrow{\mathbf{h}}_i^2, \overleftarrow{\mathbf{h}}_i^2\right], \tag{5}$$

where $\overrightarrow{\mathbf{h}}_i^2$, $\overleftarrow{\mathbf{h}}_i^2$ are the hidden states of forward and backward LSTMs respectively. The stacked LSTM architecture also adds more nonlinearity. In this case, the underlying sign information is encoded into blocks of a certain length via the bottom BDU layer. The higher-level LSTM layer is responsible for composing the encoded information to obtain the final video representation. In summary, the encoder completely encodes the hierarchical structure of the input sign language features.

Because the temporal connection control signal s_t in BDU is a non-differentiable binary variable, the network cannot be trained by the traditional gradient back-propagation algorithm. Therefore, we design a new BAE network learning method based on random function. Specifically, the boundary detection signal s_t is treated as a random neuron during the network training phase. During training, the detection signal s_t is calculated as follows:

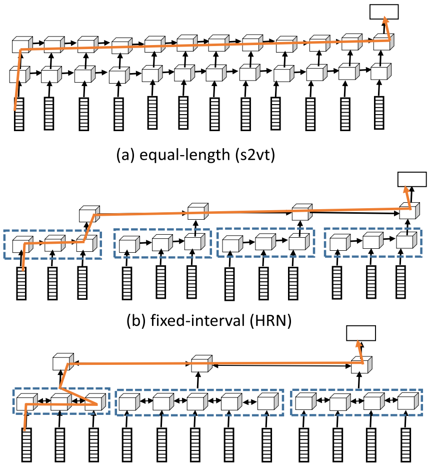
$$s_t = \begin{cases} 1, & if\ \sigma\left(\mathbf{v}_s^T \cdot \left(\mathbf{W}_s\left[\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}^1, \mathbf{x}_{t-1}, \overleftarrow{\mathbf{h}}_t^1\right] + \mathbf{b}_s\right)\right) < z \\ 0, & otherwise \end{cases} \tag{6}$$

where z is designed to be randomly sampled from a uniform distribution $U[0, 1]$, which guarantees s_t to be stochastic and its probability of being 0 or 1 is proportional to the value of the sigmoid output. During gradient back-propagation, the gradient of s_t is almost all zero, the traditional back-propagation algorithm is not suitable. Thus we use the approximate algorithm to calculate the gradient of step function $\varphi(x)$ as follows:

$$\frac{\partial \varphi(x)}{\partial x} = \sigma(x)\left(\sigma(x) - 1\right). \tag{7}$$

And then we can obtain the gradients of s_t for learnable parameters. During the prediction phase, the boundary function is updated normally. This method guarantees the randomness of boundary detection in the training phase and the certainty in the prediction phase.

The underlying sign language feature information flow in the BAE network is shown in Fig.3(c). The first frame of feature information input in the conventional encoder structure



(a) equal-length (s2vt)

(b) fixed-interval (HRN)

(c) adaptive-interval with bidirectional learning (Our BAE)

FIGURE 3. Comparison on different hierarchical encoding networks. The red line shows one of the paths from the input at t = 1 to the final output. Compared to (a) and (b), our hierarchical BAE architecture (c) is flexible.

(Fig.3 (a)) needs to go through a path that is much larger than the BAE network. Compared to the conventional or other structures that use fixed-size blocks (Fig.3 (b)) as the overall time period encoding, BDU can adaptively control the length of the encoding block according to the input information and the hidden state of the encoding layer.

C. ENCODER-DECODER WITH WINDOW ATTENTION MODEL

In the conventional encoder-decoder model, all decoding moments rely on a fixed-length context vector for decoding. This vector is output by the encoder at the last moment and contains all input sign information. However, for long sign sequence modeling, the capacity of encoding information contained in a vector is insufficient in the decoding phase. Therefore, the window attention model is incorporated into our BAE encoder-decoder model. In order to improve the decoding, the attention model is designed to put different weights on the encoding vectors of different times at the time of decoding. Considering the influence of sign language sequence length, a window position vector is introduced to ensure that the decoded output is aligned with the sign language input signal.

As can be seen in Fig.1, the windows attention model is stacked on the BAE encoder. The BAE encodes the underlying sign language features into a semantic vector sequence $\left(\mathbf{h}_1^e, \mathbf{h}_2^e, \ldots, \mathbf{h}_T^e\right)$ containing hierarchical sign sequence information. And then they are decoded to output a target sequence with attention. Fig.4 shows the window attention model.

In the window attention model, each decoding step only focuses on a fixed-length window of encoded vectors. At the decoding step t, we first generate an aligned position g_t, which is calculated as:

$$g_t = T \cdot \sigma\left(\mathbf{v}_g^T \tanh\left(\mathbf{W}_g \mathbf{h}_t\right)\right), \tag{8}$$

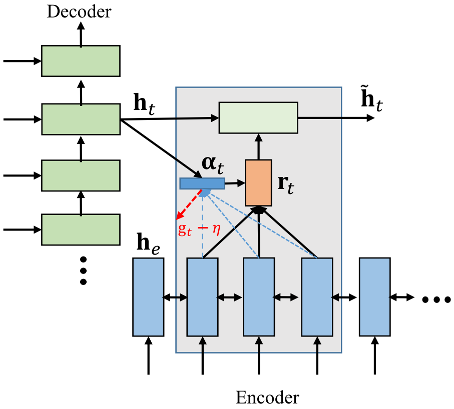


FIGURE 4. The window attention model in encoder-decoder framework.

where $\mathbf{h}_t$ is the hidden state of decoder, $\mathbf{v}_g$ and $\mathbf{W}_g$ are learnable weight parameters, and T is the length of encoding vector sequence. Then we set a window with a length of 2η centered at g_t. The context vector used at decoding step t is:

$$\mathbf{r}_t = \sum\nolimits_{i\in[gt-\eta,gt+\eta]} \alpha_{ti}\mathbf{h}_i^e, \tag{9}$$

where η is empirically set, $\mathbf{h}_i^e$ is the encoded vector of encoding step i, and α_{ti} is the attention weight assigned for $\mathbf{h}_i^e$. Unlike the attention weight α_{ti} in [41], which is affected by the length of the input sequence, we use a sigmoid function instead of softmax function to prevent the attention vector from being normalized. More importantly, we use a differentiable Gaussian function to truncate the window that needs to be aligned. The weight α_{ti} is calculated as follows:

$$\alpha_{ti} = \sigma\left(e_{ti}\right) \exp\left(-\frac{(i-g_t)^2}{2\delta^2}\right), \tag{10}$$

where δ is empirically set as the half of η, and e_{ti} denotes the correlation between $\mathbf{h}_t$ and $\mathbf{h}_i^e$. Note that i is an integer within the window centered at g_t, whereas g_t is a real number. e_{ti} can be evaluated by:

$$e_{ti} = \mathbf{h}_t^T \mathbf{W}_e \mathbf{h}_i^e, \tag{11}$$

where the weight matrix $\mathbf{W}_e$ can be obtained through joint training with the encoder-decoder network.

After obtaining a highly correlated context vector $\mathbf{r}_t$, the attention-derived decoded output $\tilde{\mathbf{h}}_t$ is obtained through a non-linear transformation, as follows:

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_r\left[\mathbf{r}_t, \mathbf{h}_t\right]\right), \tag{12}$$

where $\mathbf{W}_r$ is the weight matrix. The output distribution is then generated by feeding the attentional vector $\tilde{\mathbf{h}}_t$ into the softmax layer:

$$p\left(\mathbf{y_t} \middle| \tilde{\mathbf{h}}_t\right) = \frac{\exp\left(\mathbf{w}_y \tilde{\mathbf{h}}_t\right)}{\sum_{y' \in V} \exp\left(\mathbf{w}_{y'} \tilde{\mathbf{h}}_t\right)}, \tag{13}$$

where $\mathbf{w}_y$ is the corresponding weight parameter that can be obtained through training, y' represents a sign language subword element, and V is the subword dictionary of the dataset. The model can be written as:

$$p\left(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M \left|\mathbf{x}_1, x_2, \ldots, x_{\mathrm{N}}\right.\right) = \prod_{t=1}^{M} p\left(\mathbf{y}_t \left|\tilde{\mathbf{h}}_t, \mathbf{y}_{t-1}\right.\right). \tag{14}$$

During training, the logarithm of the above formula is used for calculation convenience. Therefore the optimal model is:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum\nolimits_{t=1}^{M} \ln p\left(\mathbf{y}_t \left|\tilde{\mathbf{h}}_t, \mathbf{y}_{t-1}\ ; \boldsymbol{\theta}\right.\right), \tag{15}$$

where $\boldsymbol{\theta}$ denotes the model parameters.

IV. EXPERIMENTS

We evaluate our method on two isolated sign language word datasets and one continuous sign language sentence dataset. The first isolated word dataset ID1 is the CSL dataset[1] which is a public large-scale vocabulary Chinese sign language dataset, and the second isolated word dataset ID2 is collected by us using the Microsoft Kinect 2.0 device for Chinese sign language. The continuous sentence dataset CD is often used in continuous Chinese SLR research.[2] We first give some implementation and dataset details, and then show the advantages of our method through some experiments.

A. EXPERIMENT SETUP

1) DATASET

For isolated SLR, we use two isolated word datasets, ID1 and ID2. Specifically, ID1 contains 125k samples consisting of 500 sign words, each of which was recorded 5 times by 50 signers. The RGB video resolution in each sample is 1280 × 720 and the fps is 30. In order to verify the reliability of our method, we have built a larger-scale vocabulary dataset ID2 with more Chinese sign words. The dataset can be divided into two parts, ID2-spit1 and ID2-spit2. Specifically, ID2-split1 contains 50k samples consisting of 500 sign words, each of which was recorded 10 times by 10 signers. And ID2-split2 contains 20k samples consisting of 2000 sign words, each of which was recorded one time by one signer. The RGB video resolution in ID2 is 1560 × 1080 and the fps is 30. For continuous SLR, we use the dataset CD, which contains 25k videos consisting of 100 different sign language sentences. To be specific, each sentence was recorded 5 times by 50 sign speakers. The resolution of the post-processed color video is 1080 × 720and the fps is 30. Besides, the length of each video is 4∼8 sign words, which is about 15 seconds. Details of the three datasets are shown in Table 1.

TABLE 1. The information of the datasets.

Dataset	Words/Sentences	Samples of each word/sentence	Total samples
ID1	500	250	125 k
ID2-split1	500	100	50 k
ID2-split2	2000	10	20 k
CD	100	250	25 k

2) EVALUATION METRICS

For isolated SLR, we concatenate each predicted sign subword sequence into one sign word. And then Error Rate is used as the criteria, that is, the ratio of all the words that are identified incorrectly. For continuous SLR, we use different criterion. Bi-gram algorithm is utilized to combine the predicted subword sequence into sentences. We then use word error rate (WER) to evaluate the performance of continuous SLR and compare it with other work. WER is defined as:

$$WER = \frac{\#substitution + \#deletion + \#insertion}{\#words\ in\ the\ target}, \tag{16}$$

where # represents the number. Note that it is the number of words, not the number of subwords. We also use semantics evaluation metrics widely used in NLP, NMT, i.e., BLEU, METEOR, ROUGE-L and CIDEr.

3) IMPLEMENTATION DETAILS

For a fair comparison, after shuffling the samples, we select 70% of the dataset as the training set and 30% as the test set in experiments. The signers appearing in the test set and training set are the same. Our stacking LSTM model each layer has 1024 cells, and 512-dimensional embeddings. During training, we use Adam optimizer to train our model. The parameters are uniformly initialized in [−0.1, 0.1], the learning rate is set to 0.001, and the batch size is 8. We apply scheduled sampling [45] to the training process. The sampling decay rate ε is from 1 to 0, and the exponential decay method is used, which is multiplied by 0.9 every 10 periods. We add a begin-of-sentence tag <bos> as the initial word to start language generation, and when the sentence ends end-of-sentence tag <eos> appears, so that our model can deal with signs with variable length. In inference, we consider using beam search in our model with search size of 3. Our program is implemented on the TensorFlow platform. Note that sign language signals are mainly expressed by the shape of human hands, whereas other torso parts of the signer body and background environment have less effect on sign language expression. In order to improve efficiency, we use the left-hand and the right-hand patches as input in all subsequent experiments we take.

B. PERFORMANCE ON THE ISOLATED WORD DATASETS

We evaluate our method on two isolated datasets. We design a 2D-CNN-based SLR benchmark method based on the VGG network [48] with temporal pooling [12]. The recognition results are obtained through a max-pooling layer and the last fully connected layer. Considering that the frame-level 2D-CNN features length is too long, we first extract the keyframes of sign language according to the method of [11]. In addition, our previous research [11] shows that all frames within a sign language video are highly differentiated.

[1] http://home.ustc.edu.cn/∼hagjie/

[2] http://mccipc.ustc.edu.cn/mediawiki/index.php/SLR

It leads to poor performance of the frame-level 2D-CNN features obtained by sign language data with video-level labels. Therefore, a 2D-CNN network pre-trained with frame-level sign language labels for sign language recognition has been designed. For the 3D-CNN, we use the C3D model [49] where each video block has 16 frames overlapping 8 frames with adjacent blocks. Finally, we utilize 1024-dimensional 2D-CNN features and 1024-dimensional 3D-CNN features as the basic features used by other SLR methods [11], [12], [49], [47].

TABLE 2. Performance of different methods on the isolated word datasets.

Method/Error Rate	ID1	ID2-split1	ID2-split2
VGG with pooling	0.404	0.429	0.457
2D-CNN [2] pre-trained	0.308	0.346	0.369
3D-CNN [7]	0.279	0.322	0.348
Temp Conv+RNN [46]	0.255	0.279	0.325
3D-CNN+RNN+CTC [47]	0.222	0.232	0.296
2D-CNN+LSTM-encoder-decoder [11]	0.173	0.21	0.307
3D-CNN+LSTM-encoder-decoder [11]	0.152	0.179	0.275
2D-CNN+Bi-LSTM-Attention [12]	0.217	0.201	0.314
3D-CNN+Bi-LSTM-Attention [12]	0.139	0.187	0.281
2D-CNN+Ours (w/o Attention)	0.151	0.164	0.272
3D-CNN+Ours (w/o Attention)	0.116	0.134	0.25
2D-CNN+Ours (with Attention)	0.138	0.15	0.256
3D-CNN+Ours (with Attention)	**0.099**	**0.114**	**0.229**

Note that for a fair comparison, we use the same RGB information in each method. Table 2 summarizes the results of different methods on the dataset ID1 and ID2. It can be seen from the results that the 3D-CNN-based method performs better than 2D-CNN because it captures more spatio-temporal sign information. In addition, the methods based on RNN and LSTM have achieved better results since memory cells can better model time sequences. With the help of the hierarchical LSTM structure and the attention model, the methods in [11] and [12] bring competitive results on the ID2 and ID1 datasets. Table 2 shows the superiority of our method. Specifically, our proposed BAE improves performance encoding both 2D-CNN and 3D-CNN feature sequences without attention model. Especially when BAE with 3D-CNN input, the results on ID1 and ID2-spit1, split2 datasets are on average 4% higher than the results of [11]. Besides, our method incorporating the window attention model can further improve the performance by an average of 2%. The results also show that our method can still perform well even when the sign vocabulary is large but the training samples are scarce.

C. PERFORMANCE ON THE CONTINUOUS SENTENCE DATASET

According to the performance of 2D-CNN and 3D-CNN in the isolated SLR experiments, we use RGB 3D-CNN features in the subsequent continuous SLR experiments for fair comparison. Note that the 3D-CNN features extraction network is pre-trained on isolated word datasets. Experiments are conducted on the continuous sentence dataset CD to compare our proposed method with other continuous SLR methods. The experimental results are summarized in Table 3.

The method of LSTM-LSTM [1] structure with mean pooling is used as the benchmark for recognizing the continuous sign language. Afterward, the LSTM-E network in [2] embeds high-level semantic information. However, using the pooling operation will average the sequence temporal information. Accordingly, the missing of the chronological information of the sign signals results in suboptimal results. In addition, the compared methods LSTM-CTC and SubUnets [3] aiming at frame-level alignment do not learn sign word semantics well.

As an encoder-decoder framework, we also compare the s2vt network with two LSTM layers [6]. At the same time, in order to compare under the same hierarchical structure, the s2vt network is extended to a 3-layer BAE-s2vt. Its bottom layer is the same as the first BAE encoding layer, with the same length and structure. The results show that extended BAE helps performance to be improved. We infer that the structure of adaptive encoding helps establish a hierarchical sign representation of visual and semantic information.

The difference between our method and HRNE [8], HRF-S [10] is that our method uses historical and future information to adaptively learn the temporal boundaries, instead of assigning fixed-length segments [8] or using non-learnable method [10]. Unlike we implicitly encode the hierarchical sign language information, SF-Net constructs information directly at different levels of sign language to obtain competitive results. In addition, the HRNE-att, HRF-S-att, and LS-HAN models with the help of the attention model for wise encoding or decoding have been improved in performance. In the same way, our method achieves state-of-the-art result with window attention. The results clearly show the superiority of our method again.

In addition, we evaluated the performance of different methods on sign language of different lengths to study the effect of sign language length on performance. And the results are shown in Fig.5. The figure shows that the SF-Net and HRF-S methods can achieve good recognition performance when the sign language length is less than about 325 frames. But the performance decreases slightly as the length increases. Differently, our method can still maintain good recognition performance when the length of sign language is long, that is, the length has a slight effect on recognition. It is because of the adaptive adjustment of BAE to longer sequences, and the window attention also guarantees the limited impact of encoding length on decoding. Meanwhile, it illustrates the potential of our method for longer sign language sentence recognition.

D. ANALYSIS ON SUBWORD

In our method, we consider using finer-grained sign language subword units instead of sign word units as the basic unit of

TABLE 3. Performance of different methods on the continuous dataset.

Method	WER	CIDEr	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR
LSTM-LSTM [1]	0.422	6.985	0.823	0.823	0.842	0.854	0.607
LSTM-E [2]	0.347	7.520	0.843	0.822	0.849	0.856	0.595
LSTM-CTC	0.255	8.415	0.852	0.886	0.897	0.899	0.632
SubUnets [3]	0.251	8.376	0.880	0.892	0.896	0.905	0.633
LS-HAN [5]	0.216	8.692	0.880	0.890	0.893	0.894	0.631
s2vt [6]	0.203	8.304	0.850	0.870	0.892	0.892	0.632
HRNE [8]	0.201	8.884	0.875	0.901	0.915	0.917	0.671
HRNE+att [8]	0.155	9.031	0.882	0.904	0.918	0.919	0.674
HRF-S [10]	0.147	8.998	0.898	0.907	0.923	0.924	0.685
HRF-S-att [10]	0.108	8.961	0.902	0.908	0.928	0.930	0.685
SF-Net [13]	0.099	8.883	0.895	0.897	0.906	0.908	0.698
BAE- s2vt	0.160	9.043	0.878	0.887	0.910	0.920	0.677
Ours	0.089	8.904	0.909	0.921	0.921	0.923	0.694
Ours-att	**0.074**	**9.037**	**0.912**	**0.926**	**0.933**	**0.934**	**0.706**

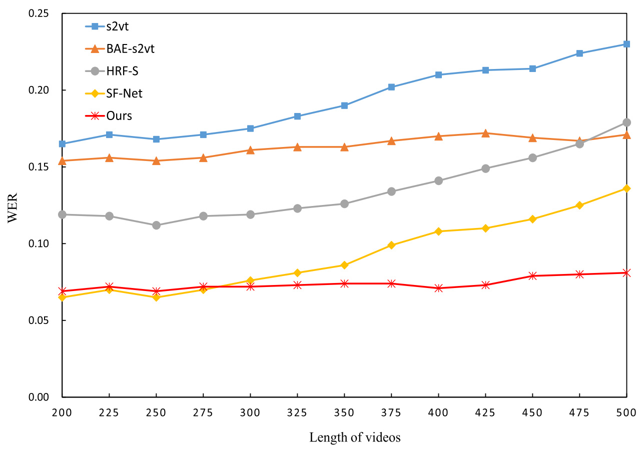


FIGURE 5. Performance of different methods on sign language of different lengths.

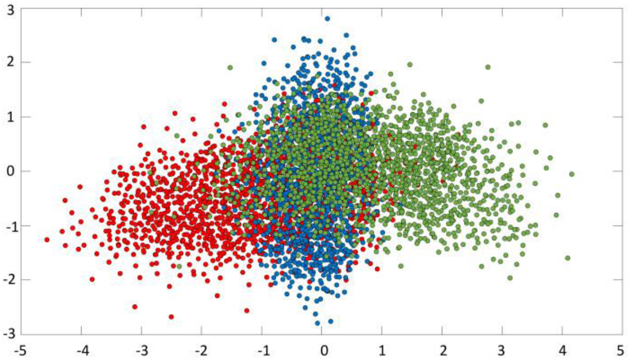
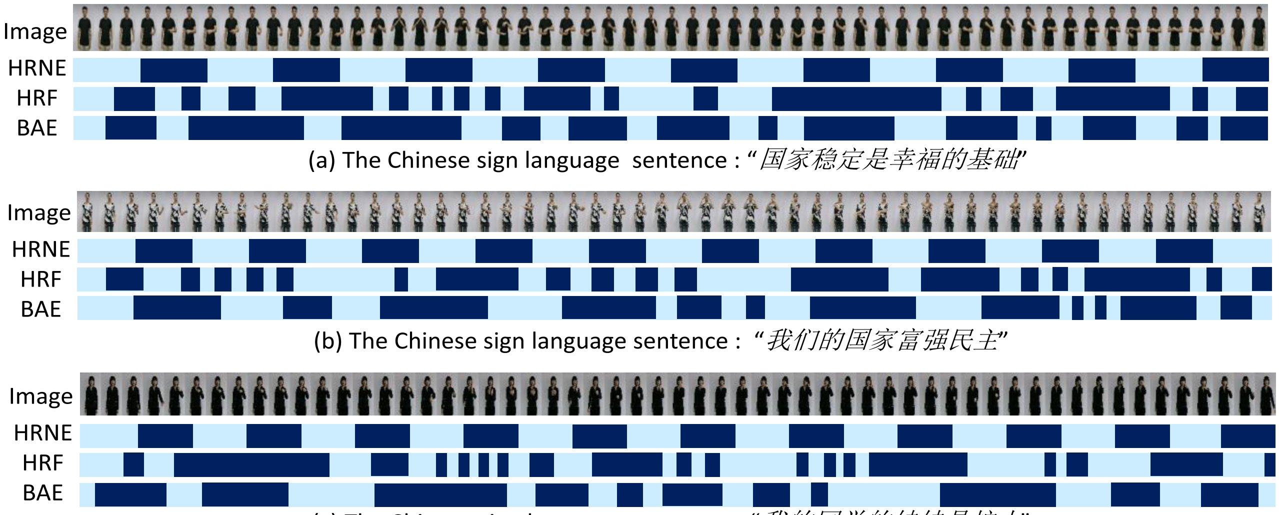


FIGURE 6. Feature visualizations by t-SNE projection. Red, green, and blue represent the Chinese sign language word sign "kindergarten", "station", and "house" respectively.

recognition in both isolated and continuous SLR. To illustrate the advantages of sub-words, we visualize the performance of some sign language words in the feature space. The 3D-CNN features of three words, ''station, kindergarten, and house'', are visualized by the t-SNE projection in Fig.6 with 50 samples each word. The features of each sign word are in the same color. Since the sign language is expressed by continuous multiple frames, the temporally adjacent frames do not change much. The figure tells that each word has a certain continuity in the feature space. More importantly, the figure clearly shows that some of the features of sign language words will overlap with others in the feature space. In other words, it is difficult to distinguish the sign language described at the word level. It shows that using smaller-grained subwords instead of sign words to represent sign language has better potential for modeling. We obtain the semantic subword units according to the method in [11]. Consequently, 310 sign language subwords are defined as the basic sign language description units for subsequent experiments.

We mix the dataset ID1 and CD to one bigger dataset and conduct experiments on the mixed dataset. It aims to simulate the situation that the target of SLR can be a word or sentence but unknown in advance in the real environment. And we then discuss the advantage of the sign subwords. Specifically, we compare our method with LSTM-CTC and SubUnets based on frame-level alignment, s2vt, and BAE-s2vt based on encoder-decoder, HRNE-att and HRF-s-att based on hierarchical model with attention, and SF-Net based on structural features. We use WER as the evaluation criterion. Note that other methods treat isolated sign language words as sequences with only one element, while our method uses or does not use sign language sub-word sequences. The experimental results are shown in Table 4. The results show that the performance of each method on the mixed dataset is slightly worse than that performed on the CD dataset, although there are larger training data and shorter sign language sequences on average. Presumably, the expansion of



(a) The Chinese sign language sentence : “国家稳定是幸福的基础”

(b) The Chinese sign language sentence : “我们的国家富强民主”

(c) The Chinese sign language sentence : “我的同学的妹妹是护士”

FIGURE 7. The visualization of the boundaries of three continuous Chinese sign language sentences obtained by different methods. Different color blocks indicate different segmented fragments.

TABLE 4. Performance of different methods on the mixed dataset.

Method	WER	CIDEr	ROUGE-L	METEOR
LSTM-CTC	0.308	8.207	0.853	0.597
SubUnets [3]	0.347	8.132	0.880	0.601
LS-HAN [5]	0.366	8.458	0.846	0.612
s2vt [6]	0.256	8.236	0.883	0.618
HRNE+att [8]	0.191	8.994	0.907	0.648
HRF-S-att [10]	0.143	8.852	0.918	0.647
SF-Net [13]	0.129	8.452	0.894	0.664
BAE-s2vt	0.179	8.979	0.911	0.653
Ours w/o subword	0.117	8.991	0.909	0.656
Ours with subword	**0.079**	**9.022**	**0.918**	**0.678**

the search space (projecting space of recognition target) for recognition caused difficulties. However, our method based on sign language subwords performs well on the mixed dataset. Because we use the same sign subwords to represent isolated and continuous sign language, the search space for recognition has not been expanded. The experimental results demonstrate the adaptability and robustness of our method for sign language sequences of unknown length.

E. ANALYSIS ON BAE

In order to learn the performance of BAE in sign language boundary learning, we show the boundary learning results of three continuous sign language sentences in Fig.7. Note that the figure does not draw all the frames of videos. It can be seen from Fig.7 that HRNE segments sign language video into multiple very small and equal length fragments. It is based on the distribution of the overall sample set. Unlike HRNE, BAE and HRF both segment the video into adaptive fragments of appropriate length according to the distribution of the current video. The difference is that HRF segmentation is based on changes between source sequence features, but BAE segmentation is flexibly learned from the source and target sequences. HRF only splits sharply changing parts of the video, but cannot effectively segment some slowly changing but meaningful transitions. In addition, some continuous but dramatic motion changes can be semantically encoded by the BAE as a whole, rather than frequently segmented by HRF. After many iterations, the BAE learns the cumulative information in sign language over a certain period of time, thereby realizing boundary learning and boundary encoding. Obviously, the feature information of sign language is converted into encoded information with a hierarchical structure.

TABLE 5. Performance of ours methods with/without bdu.

Method/Error Rate/WER	ID1	ID2-split1	ID2-split2	CD
Single-layer Bi-LSTM w/o BDU	0.235	0.267	0.291	0.302
Single-layer Bi-LSTM + BDU	0.194	0.214	0.256	0.238
Double-layer Bi-LSTM w/o BDU	0.178	0.228	0.261	0.213
Double-layer Bi-LSTM + BDU	**0.099**	**0.114**	**0.229**	**0.074**

We conducted some comparison experiments with and without BDUs in different structures on the dataset ID1, ID2-split1, ID2-split2 and CD. We use single-layer Bi-LSTM and double-layer Bi-LSTM structures, respectively. The experimental results are summarized in Table 5. The experimental results show that adding BUD between the LSTM is more conducive to the modeling of sign language segmentation information, thereby improving the performance of sign language recognition. In addition, using a double-layer LSTM structure results better than using a single-layer LSTM structure. This is because the single-layer structure

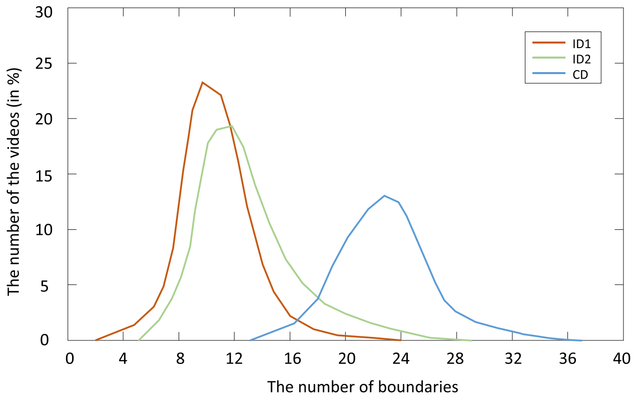


FIGURE 8. The distribution of the number of the detected boundaries on the three datasets.

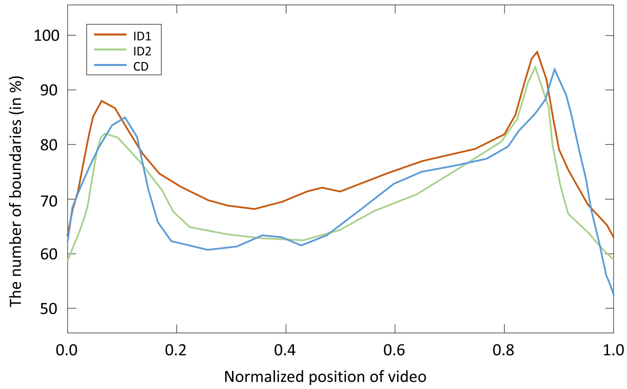


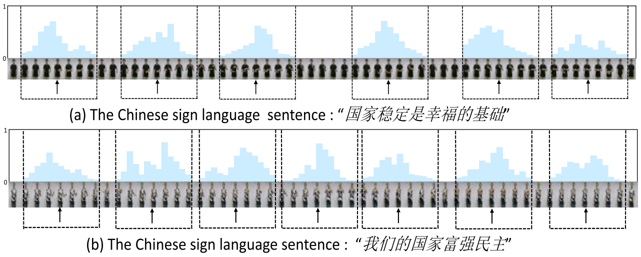
FIGURE 9. The distribution of the position of the detected boundaries on the three datasets.

is not enough to model the long-term and short-term timing relationships of sign language. At the same time, the multi-layer structure can add more nonlinear.

In addition, we learn the boundary information of each data set. The distribution of the number and position of the detected boundaries on each dataset are illustrated in Fig.8 and Fig.9. Fig.8 shows that the number of boundaries of isolated word datasets is mostly around 10, and the number of boundaries of continuous sign language sentence dataset is about 20-26. In Fig.9, the position of very video is normalized to 0 to 1. And it shows that there are two peaks at the beginning and the end of sign videos, which are the beginning and end of sign language, respectively. We also can observe that the boundary distribution of the middle position of the video increases with time. It means that the information accumulated by the encoder increases with time, and the more detailed the video segmentation.

F. ANALYSIS ON ATTENTION

The window attention model we designed can align the decoded output with the input sign signal. As a result, the weight assignment of different input signals is realized, and on the other hand, the related encoded information is not diluted when the sign language length is long. Visualizing the attention weights of some samples is shown in Fig.10. The figure tells that the window alignment position corresponding to each sign subword is often near the key position of the chunk content. The attention addresses inputs that are more useful for decoding at the current moment. Different from using the softmax function in [41] to constrain the sum of attention weights to 1, we use the sigmoid function to relax the constraint. Therefore, the attention weight of an element at a certain moment can be prevented from being affected by the attention values of other elements. In this way, the attention along time will not be destabilized by other elements, so it is easier to optimize. Compared with the method in [41], we can keep the important elements at a more effective weight instead of a very small value. And considering the adaptive performance of the BAE layer, even though the size of the attention window is fixed, the length of the corresponding sign language segment is flexible.



(a) The Chinese sign language sentence : “国家稳定是幸福的基础”

(b) The Chinese sign language sentence : “我们的国家富强民主”

FIGURE 10. Two Chinese sign language sentences and attention of them. (The arrow points to the center of the window. Not all frames and results are shown).

V. CONCLUSION

In this paper, we developed a new SLR method based on the boundary-adaptive encoder incorporating with window attention and achieved competitive results across popular benchmarks. We have taken strong steps to solve the problem that the previous SLR methods would severely reduce performance in the case of long sequences. Besides, we unified isolated and continuous SLR into the same recognition method that is more practical by introducing sign language subwords. But there are still a limitations in our models. The one is real-time performance and another one is the model is not lightweight enough. In the future, we will improve our model from the two aspects to bring better performance.

REFERENCES

[1] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, “Translating videos to natural language using deep recurrent neural networks,” 2014, *arXiv:1412.4729*. [Online]. Available: http://arxiv.org/abs/1412.4729

[2] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign language recognition using convolutional neural networks,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 572–578.

[3] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, “SubUNets: End-to-end hand shape and continuous sign language recognition,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3075–3084.

[4] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, “SSNet: Scale selection network for online 3D action prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8349–8358.

[5] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[6] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—Video to text," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4534–4542.

[7] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3D convolutional neural networks," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jun. 2015, pp. 1–6.

[8] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1029–1038.

[9] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 1–7.

[10] D. Guo, W. Zhou, A. Li, H. Li, and M. Wang, "Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation," *IEEE Trans. Image Process.*, vol. 29, pp. 1575–1590, 2020.

[11] S. Huang, C. Mao, J. Tao, and Z. Ye, "A novel Chinese sign language recognition method based on keyframe-centered clips," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 442–446, Mar. 2018.

[12] J. Huang, W. Zhou, H. Li, and W. Li, "Attention-based 3D-CNNs for large-vocabulary sign language recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2822–2832, Sep. 2019.

[13] Z. Yang, Z. Shi, X. Shen, and Y.-W. Tai, "SF-Net: Structured feature network for continuous sign language recognition," 2019, *arXiv:1908.01341*. [Online]. Available: http://arxiv.org/abs/1908.01341

[14] S. Zhang, W. Meng, H. Li, and X. Cui, "Multimodal spatiotemporal networks for sign language recognition," *IEEE Access*, vol. 7, pp. 180270–180280, 2019.

[15] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.

[16] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1657–1666.

[17] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[18] C. Vogler and D. Metaxas, "Parallel hidden Markov models for American sign language recognition," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, Sep. 1999, pp. 116–122.

[19] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.

[20] P. Jangyodsuk, C. Conly, and V. Athitsos, "Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented gradient features," in *Proc. 7th Int. Conf. Pervas. Technol. Rel. Assistive Environ. (PETRA)*, 2014, pp. 1–6.

[21] H.-D. Yang, "Sign language recognition with the kinect sensor based on conditional random fields," *Sensors*, vol. 15, no. 1, pp. 135–147, Dec. 2014.

[22] A. Agarwal and M. K. Thakur, "Sign language recognition using microsoft kinect," in *Proc. 6th Int. Conf. Contemp. Comput. (IC)*, Aug. 2013, pp. 181–185.

[23] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 44–52.

[24] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[25] S. Wang, D. Guo, W.-G. Zhou, Z.-J. Zha, and M. Wang, "Connectionist temporal fusion for sign language translation," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1483–1491.

[26] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2871–2875.

[27] J. Zhang, W. Zhou, and H. Li, "A threshold-based HMM-DTW approach for continuous sign language recognition," in *Proc. Int. Conf. Internet Multimedia Comput. Service (ICIMCS)*, 2014, pp. 237–240.

[28] K. Li, Z. Zhou, and C.-H. Lee, "Sign transition modeling and a scalable solution to continuous sign language recognition for real-world applications," *ACM Trans. Accessible Comput.*, vol. 8, no. 2, pp. 1–23, Jan. 2016.

[29] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, "Chinese sign language recognition with adaptive HMM," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[30] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 474–490.

[31] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7361–7369.

[32] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4297–4305.

[33] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid CNN-HMM for continuous sign language recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016.

[34] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical LSTM for sign language translation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[35] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[36] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[37] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[38] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," 2014, *arXiv:1412.7755*. [Online]. Available: http://arxiv.org/abs/1412.7755

[39] P. Sermanet, A. Frome, and E. Real, "Attention for fine-grained categorization," 2014, *arXiv:1412.7054*. [Online]. Available: http://arxiv.org/abs/1412.7054

[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[41] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*. [Online]. Available: http://arxiv.org/abs/1508.04025

[42] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," 2015, *arXiv:1511.04119*. [Online]. Available: http://arxiv.org/abs/1511.04119

[43] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li, "Hierarchical attention network for action recognition in videos," 2016, *arXiv:1607.06416*. [Online]. Available: http://arxiv.org/abs/1607.06416

[44] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[45] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1171–1179.

[46] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 430–439, Apr. 2018.

[47] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4207–4215.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[49] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[50] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13009–13016.

[51] N. Wang, S. Ma, J. Li, Y. Zhang, and L. Zhang, "Multistage attention network for image inpainting," *Pattern Recognit.*, vol. 106, Oct. 2020, Art. no. 107448.

[52] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, "Text2Sign: Towards sign language production using neural machine translation and generative adversarial networks," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 891–908, Apr. 2020.

[53] N. Wang, Y. Zhang, and L. Zhang, "Dynamic selection network for image inpainting," *IEEE Trans. Image Process.*, vol. 30, pp. 1784–1798, 2021.

SHILIANG HUANG received the B.E. degree in electronic information science and technology from North China Electric Power University, Baoding, China, in 2011. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China, Hefei, China. His research interests include image processing and video understanding, especially the sign language recognition, video caption, and action recognition.

ZHONGFU YE received the B.Eng. and M.S. degrees in electronic and information engineering from the Hefei University of Technology, Hefei, China, in 1982 and 1986, respectively, and the Ph.D. degree from the University of Science and Technology of China, Hefei, in 1995. He is currently a Professor with the University of Science and Technology of China. His current research interests include statistical and array signal processing, speech processing, and image processing.