Hierarchical Recurrent Deep Fusion Using Adaptive Clip Summarization for Sign Language Translation

Dan Guo, Wengang Zhou, Anyang Li, Houqiang Li, *Senior Member, IEEE*, and Meng Wang, *Senior Member, IEEE*

***Abstract*— Vision-based sign language translation (SLT) is a challenging task due to the complicated variations of facial expressions, gestures, and articulated poses involved in sign linguistics. As a weakly supervised sequence-to-sequence learning problem, in SLT there are usually no exact temporal boundaries of actions. To adequately explore temporal hints in videos, we propose a novel framework named Hierarchical deep Recurrent Fusion (HRF). Aiming at modeling discriminative action patterns, in HRF we design an adaptive temporal encoder to capture crucial RGB visemes and skeleton signees. Specifically, RGB visemes and skeleton signees are learned by the same scheme named Adaptive Clip Summarization (ACS), respectively. ACS consists of three key modules, *i.e.*, variable-length clip mining, adaptive temporal pooling, and attention-aware weighting. Besides, based on unaligned action patterns (RGB visemes and skeleton signees), a query-adaptive decoding fusion is proposed to translate the target sentence. Extensive experiments demonstrate the effectiveness of the proposed HRF framework.**

***Index Terms*— Sign language translation, hierarchical adaptive temporal network, adaptive clip summarization, temporal pooling, score fusion.**

I. Introduction

SIGN language is a form of communication extensively used by the deaf community as well as in some action-based applications, *e.g.*, virtual and augmented reality. Sign language is challenging to interpret because it involves complicated variations of gestures, skeletal movements, finger orientations, and facial expressions under sign linguistics [1]. Figure 1 depicts the video sequences of two types of vision-based sign interpretation, sign language recognition (SLR) and sign language translation (SLT). SLR, which can be considered a video classification task, translates the video sequence into an individual word [2], [3]. SLR aims to address the mapping between visual content and vocabulary. In contrast, SLT outputs a sentence with words in a specific order, which is analogous to continuous action recognition. Since SLT always lacks accurate temporal location information for each sign gloss (word) [4], [5], SLT can be considered to be a type of weakly-supervised sequential learning problem. In this study, we focus on the SLT task, which is more common in terms of daily usage. SLT has attracted considerable attention in the field of computer vision.

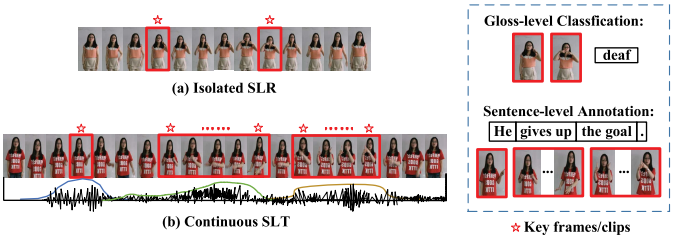


Fig. 1. (a) Isolated SLR and (b) continuous SLT. Signs are easily recognized by discriminative variations of gestures and articulated poses. In this paper, we aim to find out key frames/clips in videos.

Generally, signs are easily recognized by the discriminative variations of gestures and articulated poses. We intend to find discriminative actions in videos. However, in case of SLT, the lack of temporal annotations impedes the possibility of precisely segmenting actions. In addition, SLT suffers from hybrid semantic transformations among visual cues, sign linguistics, and textual grammar, which presents various linguistic challenges. For example, the adverb in the phase "run quickly" is indicated by increasing the speed of signing the word "run" [6]. Other challenges involve uncertain directional verbs and positional signs between the signer and viewers, unknown words with finger spelling, and non-hand auxiliary features, such as facial expression and lip shape.

Among video understanding topics, the closest one to SLT is video captioning [7]. SLT emphasizes the dense variations of sequential actions with complicated linguistics whereas video captioning relies on grammar knowledge and semantic coherence with visual feature representations of object(s), scene(s), and motion for sentence generation. As for the STL task, current researchers have focused on frame-level or gloss-level sign recognition, where **gloss** denotes the textual semantic unit of **sign word** [8]. Commonly applied effective approaches introduce a Hidden Markov Model (HMM) [9],

Manuscript received May 10, 2018; revised November 14, 2018, March 15, 2019, May 8, 2019, and August 21, 2019; accepted August 26, 2019. Date of publication September 23, 2019; date of current version November 7, 2019. The work of D. Guo was supported in part by the National Natural Science Foundation of China (NSFC) under Contract 61876058. The work of W. Zhou and H. Li was supported in part by NSFC under Contract 61632019. The work of M. Wang was supported in part by NSFC under Contract 61725203 and Contract 61732008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yun Fu. *(Corresponding authors: Wengang Zhou; Meng Wang.)*

D. Guo and M. Wang are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: guodan@hfut.edu.cn; eric.mengwang@gmail.com).

W. Zhou and H. Li are with the EEIS Department, University of Science and Technology of China, Hefei 230026, China (e-mail: zhwg@ustc.edu.cn; lihq@ustc.edu.cn).

A. Li was with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China. He is now with Huawei Cloud AI Platform, Jiangsu 20130048, China (e-mail: liay@mail.hfut.edu.cn).

Digital Object Identifier 10.1109/TIP.2019.2941267

[10] or Connectionist Temporal Classification (CTC) [11] into a deep learning framework. These previous studies all involve sequential consistency, *i.e.*, the word order in the sentence corresponds to the visual content in the video. However, two primary impediments exist for sign video interpretation: (1) there is no strict one-to-one mapping between each gloss and its corresponding visual content, and (2) sign actions can be simplified or combined under some specific linguistic rules for gloss/phrase building. Thus, locating the temporal positions of discriminative actions in videos is difficult. The visual units of key frames/clips can help identify implicit motion patterns as shown in Fig. 1.

To address problems associated with SLT, we propose a Hierarchical deep Recurrent Fusion (HRF) network. The proposed HRF employs a hierarchical recurrent architecture to encode the visual semantics with different visual granularities (*i.e.*, frames, clips, and **visemes/signemes**). Motivated by the concept of phonemes in speech recognition, we define **viseme** as a visual unit of discriminative action under RGB channels, *i.e.*, a decomposed sub-visual-word (sub-semantic unit). Similarly, **signeme** is defined under the skeleton channel. Moreover, HRF makes use of complementary of RGB visemes and skeleton signemes to decode a sentence.

The core steps of the HRF are as follows. (1) Based on an encoder-decoder framework, the proposed HRF translates a video into neural languages after encoding the entire visual content. This framework can solve the disordered gloss label issue. (2) We employ Adaptive Clip Summarization (ACS) to explore sign action patterns in SLT. Differing from previous studies that extract key frames or clips with a fixed time interval [12]–[14] or fixed clip number [15], we propose a adaptive temporal segmentation scheme, *i.e.*, ACS. The proposed scheme automatically obtains variable-sized key frames/clips and implements dynamic temporal pooling on less-important frames/clips. Then, the compact vectors are considered as **"visemes/signemes"** under respective RGB/skeleton channels. Learning effective visemes/signemes can help identify crucial semantic units. (3) Sequential learning always suffers from the gradient attenuation of Recurrent Neural Network (RNN) modules along a long-term temporal transition. For videos with long sentence translations, previous temporal hints may disappear gradually during the long encoding time steps. Thus, we construct a hierarchical adaptive temporal encoding network (HRF encoder) to condense the time span by employing the concepts of multi-granularity and viseme/signeme. In this study, we selected LSTM as the basic RNN unit. As shown in Fig. 2, the top layer ($LSTM_1$) learns the recurrent characteristics of original features, the medium layer ($LSTM_2$) learns the recurrent characteristics of compact visemes/signemes, and the bottom layer ($LSTM_3$) primarily transforms visual semantics into textual semantics. (4) Finally, inheriting from a two-stream encoding stage, *i.e.*, frame-level skeleton trajectory vs. clip-level visual RGB images, a stacked decoder jointly translates these two types of visual semantics. To explore the complementarity of different visual semantic types, a query-adaptive fusion scheme based on deep probability scores is proposed. As shown in Fig. 2, the fusion scheme is imposed on $LSTM_3$ at each decoding time stamp.

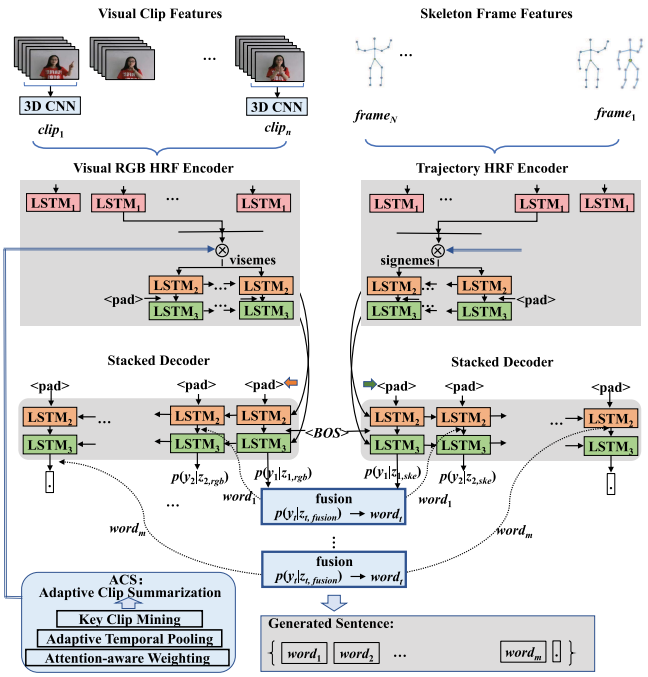


Fig. 2. Basic architecture of the proposed model. Based on an encoder-decoder framework, this model tackles RGB-skeleton-based SLT as follows. (1) Representation learning of local and global temporal hints in videos via a 3D CNN and $LSTM_1$. (2) Inhibiting attenuation of an RNN in long-term sequential learning, *i.e.*, an adaptive temporal encoder (the HRF encoder including $LSTM_{1\sim3}$ with an ACS scheme). (3) A solution of the weak supervision in videos, *i.e.*, directly switching the latent temporal hints (sub-visual-words: visemes/signemes) to sentence-level learning. A single learning flowchart is denoted as HRF-S. (4) Flexible fusion strategy under unaligned sub-semantic units (visemes and signemes), *i.e.*, two-stream HRF.

An overview of the proposed HRF is shown in Fig. 2. For the input data, clip-level 3D Convolutional Neural Network (CNN) features are extracted using C3D [16] on RGB images, while frame-level skeleton descriptors are extracted from skeletal data collected by a Kinect® device. C3D learns the spatiotemporal hints in each short-term clip, while $LSTM_1$ is used to acquire long-term temporal hints among sequential features. The skeletal data are auxiliary to the RGB images, which are characterized by 3D coordinates of articulated joints [17].

Implementation of the ACS scheme on the recurrent outputs of $LSTM_1$ involves the following steps. (1) "ACS-1: key clip segmentation" calculates the residual sum of squares (RSS) of features, which distinguishes variable-sized key and less-important clips. (2) "ACS-2: adaptive temporal pooling" is implemented on redundant less-important clips to obtain visemes/signemes. (3) Finally, "ACS-3: attention-aware weighting" assigns a higher weight to more crucial visemes/signemes, which further balances the intrinsic temporal correlation among visemes/signemes. The design of ACS is described in detail in Section III-A.

The major contributions of this study can be summarized as follows:

- *Representation Learning of Temporal Hints in Videos:* For RGB images, HRF jointly utilizes 3D CNN and $LSTM_1$ to capture more reasonable temporal representations of a video. A 3D CNN obtains short-term transitions based

on the RGB clip unit (*i.e.*, local context correlation), while $LSTM_1$ maintains long-term transitions across the entire video (*i.e.*, global context correlation). Similarly, for skeletal data, $LSTM_1$ remains the long-term recurrent characteristic of the original features.

- *Adaptive Clip Summarization in the Deep Framework:* To discover action patterns and condense the temporal transition effectively, the HRF encoder applies the ACS scheme to adaptively explore crucial visual sub-semantic units (*i.e.*, visemes/signemes). With temporal transitions among visemes/signemes in the deep learning framework, HRF achieves high-level feature embedding under RGB/skeleton channels.
- *Addressing Weak Supervision in Videos:* Weak supervision always exists in video analysis wherein videos have sentence-level without exact word-level temporal locations. In the proposed approach, HRF applies visemes and signemes to decode a sentence. In other words, discriminating sub-semantic units (visemes and signemes) is useful for action recognition. Rather than modeling the temporal conversion at word-level or sentence-level, HRF solves the problem based on the encoding of sub-semantic units (*i.e,* sub-visual-words, sub-actions).
- *Flexible Two-Stream Fusion Under Unaligned Sub-Semantics:* Differing from previous fusion strategies that emphasize alignment consistency, our approach allows unaligned sub-semantic sequences, *i.e.*, unaligned visemes and signemes. This is because that our proposed fusion strategy tackles the last encoding vectors under different data-modalities, rather than directly fusing visemes and signemes. The selection of visemes/signemes depends on whether they can help discover discriminative action patterns. Moreover, our fusion approach is implemented using deep scores, which is a heuristic, query-adaptive, and unsupervised strategy.

The remainder of this paper is organized as follows. Section II introduces related work. Section III elaborates the proposed HRF approach. Model functions and derivative discussions are detailed in Section IV. Analysis of the experimental results is presented in Section V. Conclusions are given in Section VI.

II. Related Work

This section reviews work related to three aspects of SLT, *i.e.*, hand-crafted features, classical temporal learning models, and prevalent deep features and models.

A. Hand-Crafted Features

Hand-crafted features include visual and depth features. Among depth features, point clouds [18] and surface ace normals [19] are widely used for gesture recognition by applying 3D coordinates to locate articulated postures. Skeletal coordinates and depth images are always utilized for human behavior recognition. Previous studies have transformed 3D joint coordinates into new person-centric coordinate systems that cover the entire human body [20]–[22]. For example, Rahmani and Bennamoun [20] divided a human pose into 19 body-part. In another study [21], a skeleton is rotated around an axis defined by joint 0 and 1 by $\pi/4$ to $7\pi/4$, which increases the feature length eight times. These features can help achieve a good understanding of the human posture.

However, depth data alone may be insufficient to distinguish similar sign words. Some researchers have resorted to using hand-crafted visual features of RGB images. To address SLR based on RGB images, Wang *et al.* applied HOG [23], and Hernandez-Vela *et al.* employed a the bag-of-visual-and-depth-word descriptor for multimodal feature fusion [24]. Tewari *et al.* employed the AdaBoost algorithm based on a proposed Haar-like feature to integrate several weaker classifiers into a strong classifier [25].

B. Classical Temporal Learning Models

In early gesture classification studies, various models were proposed to handle sequential dynamics. For example, native and improved SVM models have been widely applied [26], [27], such as the Grassmann Covariance Matrix has been employed as the kernel of an SVM classifier [27]. In addition, sparse coding [28] and Dynamic Time Warping (DTW) [29] have been proposed to address SLR tasks. Here, DTW measures the similarity distance between two different sequences. Celebi *et al.* developed a weighted DTW to optimize the discriminant ratio of joints [30]. Researchers have also further analyzed the hidden state transition among sequential features by clustering features into different groups, *i.e.*, states, and modeling these states using the probability distribution or a graph structure, *e.g.*, HMMs [9], [31], [32], hidden conditional random fields [33], and autoregressive models [34]. Among these, HMMs have been investigated most frequently.

C. Deep Learning-Based Approaches

1) Basic Deep Networks: Recently, deep learning-based approaches have achieved impressive success in computer vision studies [35], [36]. Such approaches are frequently applied to SLR tasks, such as the application of CNN [37], [38], LSTM [39], RNN [40]. For example, a CNN-based multi-scale learning framework has been proposed to recognize isolated gestures [17], and Molchanov *et al.* designed a 3D CNN feature extractor to represent visual dynamics in both appearance and motion views [41]. To capture sequential variation, Camgoz *et al.* employed a recurrent 3D CNN with an embedded CTC function to classify feature-level hand gestures [38]. Lefebvre *et al.* proposed a bidirectional LSTM (BLSTM) to tackle dynamic variation of the preceding and following context [42]. In addition, two-stream RNNs have been designed to handle sequential learning based on multimodal features [43].

2) Combined CNN and Sequential Learning Models: A prevalent trend is to exploit the advantages of a CNN for feature learning and temporal models for sequential learning, such as recurrence and temporal convolutions [44], a recurrent 3D CNN [41], and DNN with an embedded HMM [45]. Increasingly complex combinations are being proposed. For example, the DeepConvLSTM model, which combines both convolutional and recurrent neural units, has been proposed [46] to

address wearable activity recognition. In addition, to deeply explore temporal hints, various LSTM-based temporal pooling strategies have been introduced [44], [46]. These approaches model both spatial and temporal contexts of motion variation simultaneously. However, most are designed for isolated gesture or sign recognition and do not apply to continuous sign sentence translation.

In SLT tasks, it is crucial to obtain temporal boundaries of continuous sign words. Thus, sign action spotting [47] and alignment analysis [11], [48] have been studied extensively. Typically, sign spotting has accurate frame-level labels. However, in this study, SLT is considered a weakly-supervised problem without temporal location labels of sign glosses. For the gloss (word)-level alignment, Koller *et al.* integrated a deep CNN and the classical HMM framework [9], [10]. Cui *et al.* proposed a hybrid network based on LSTM and CTC to solve connectionist feature-level alignment [11]. These proposed approaches must satisfy a common prerequisite, *i.e.*, gloss order in the sentence label should be consistent with that of the corresponding visual contents. In contrast, in our previous study [49], we proposed a multilayer and asymmetrical LSTM model for RGB-based SLT, which does not have to satisfy this constrain. In this paper, we have further extended the preliminary version [49] to a unified two-stream encoder-decoder fusion framework to integrate visual semantic embedding from different modality data (*i.e.*, RGB images and skeleton data) toward more effective SLT, and the newly designed linear pooling and trajectory-based HRF encoder are proposed for skeleton-based SLT.

3) Hierarchical Deep Networks: To incorporate both short and long-term temporal transitions for human behavior understanding in videos, researchers have increasingly focused on the design of hierarchical deep networks. For example, a hierarchical attention network (HAN) has been proposed for action recognition [50]. HRNE is a compact hierarchical network with fixed-length compression for video captioning [14]. In addition, a hierarchical network with an embedded DTW model has been proposed for SLT [51]. Similarly, Liu *et al.* proposed a multilayer dilated CNN for online sequential action recognition [22]. However, this proposed method requires strict supervision annotations [22], *i.e.*, exact start and end time labels for each action. In contrast, in our task the labels are sentence-level without such detailed temporal cues.

4) Fusion Approaches: Fusion approaches have also been used for sign interpretation [52], [53]. Wu *et al.* extracted skeletal features using a deep belief network and visual features using a 3D CNN and then feed the extracted features into an HMM model for gesture recognition [45]. Neverova *et al.* proposed ModDrop, a multi-scale and multimodal neural network that targets correlation learning among multiple modalities [17]. The approach proposed by Wu *et al.* [45] implements the score fusion while the approach proposed by Neverova *et al.* [17] includes both feature and score fusions. These approaches are primarily designed to address isolated SLR. In this paper, we propose a two-stream fusion for RGB-skeleton-based SLT, which is based on a sequence-to-sequence learning framework. Our proposed approach integrates visual semantics from different modalities and explores their complementarity using a decoding fusion scheme.

TABLE I
PARAMETER NOTATIONS

Symbol	Description
N	Number of frames in a video, *i.e.*, number of skeleton features.
n	Number of RGB image clips of a video, *i.e.*, number of 3D CNN visual features.
$\bar{n}$	Number of extracted features ($\bar{n} = N$ or $\bar{n} = n$), *i.e.*, encoding time steps of $LSTM_1$.
n'	Number of viseme or signeme units, *i.e.*, encoding time steps of $LSTM_2$ and $LSTM_3$.
m	Number of generated words, *i.e.*, decoding time steps of $LSTM_2$ and $LSTM_3$.

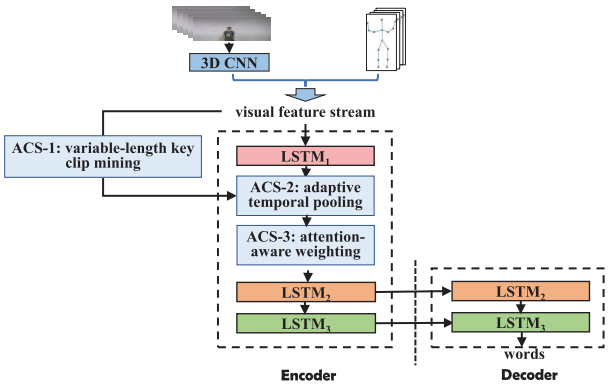


Fig. 3. Modular flowchart of single feature stream of proposed HRF (denoted HRF-S). Here, "ACS-1," "ACS-2," and "ACS-3" denote the steps of variable-length key clip mining, adaptive temporal pooling, and attention-aware weighting, respectively. RGB and skeleton features have their own respective ACS and $LSTM_{1\sim3}$ modules.

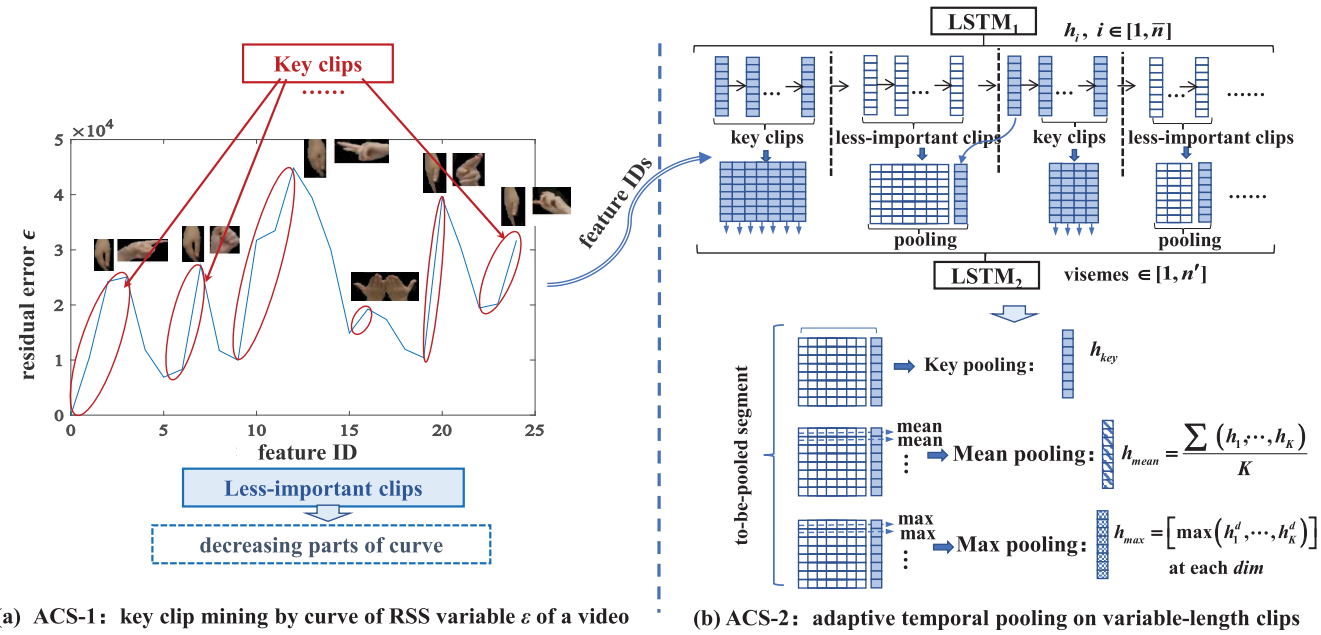
III. PROPOSED APPROACH

The general framework of the proposed model is shown in Fig. 2. Two sequences of a video, *i.e.*, RGB clips ($clip_1$, $\cdots$, $clip_n$) and skeleton frames (f_1, $\cdots$, f_N), are input to the model. The model then jointly outputs a sequence of gloss labels (y_1, $\cdots$, y_m). We discuss each module in detail in the following. Notations related to the adaptive length parameters in the proposed model are given in Table I.

A. Adaptive Temporal Encoding Network

As shown in Fig. 3, given a video (either RGB images or skeleton data), the proposed model encodes it into a semantic vector V using a three-layer RNN module, where LSTM is employed as the basic RNN cell. In the HRF model, the top layer ($LSTM_1$) is used to model the recurrent characteristic of the original features, and the other layers are designed for visual viseme/signemes encoding ($LSTM_2$) and textual decoding ($LSTM_3$).

As discussed in Section I, one core idea of the proposed model is learning compact and high-level descriptors of sub-visual-words, *i.e.*, visemes and signemes. Thus, we design an ACS scheme and embed it into the proposed HRF encoder phase. The ACS scheme works between $LSTM_1$ and $LSTM_2$



(a) ACS-1: key clip mining by curve of RSS variable ε of a video **(b) ACS-2: adaptive temporal pooling on variable-length clips**

Fig. 4. Illustration of ACS-1 and ASC-2 on RGB images. (a) ACS-1: each peak point corresponds to an active sub-action of sign glosses. The monotonically increasing parts of ϵ are considered as key clips, while other monotonically decreasing parts correspond to less-important clips. (b) ACS-2: key, mean and max temporal pooling strategies.

layers. It involves three steps denoted "ACS-1", "ACS-2" and "ACS-3", which are described in the following.

1) ACS-1: Variable-Length Key Clip Mining: Differing from current deep models that extract key frames/clips with a fixed temporal interval [12]–[14] or fixed clip number [15], we implement adaptive key clip mining. We use low-rank approximation to obtain the linear correlation of a consecutive feature stream [23]. The ACS-1 step calculates the feature RSS ϵ between the previous features and the current feature. We then design a key clip selection strategy by evaluating ϵ, which adaptively selects variable-length key clips without a threshold for different video samples. Note that the number of key clips is adaptive. We denote the ACS-1 procedure as function Ω_{rss}.

Given a video feature stream $F = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{\bar{n}}]$, we use a correlation matrix $\mathcal{M}$ to calculate residual error ϵ_i for the current feature $\boldsymbol{f}_i$. The subset of all previous features is expressed as $F_{i-1} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{i-1}]$. Here, we initialize $\epsilon_1 = 0$ and $\mathcal{M} = (\boldsymbol{f}_1^T \boldsymbol{f}_1)^{-1}$. At time step i, where $2 \le i \le \bar{n}$, we compute the correlation coefficient $\boldsymbol{\beta}_i$ and residual error ϵ_i using Eq. 1.

$$\begin{cases} \boldsymbol{\beta}_i = \mathcal{M} {F_{i-1}}^T \boldsymbol{f}_i \\ \epsilon_i = (\boldsymbol{f}_i - F_{i-1}\boldsymbol{\beta}_i)^T (\boldsymbol{f}_i - F_{i-1}\boldsymbol{\beta}_i) = \left\| \boldsymbol{f}_i - F_{i-1}\mathcal{M}{F_{i-1}}^T \boldsymbol{f}_i \right\|^2 . \end{cases} \tag{1}$$

Next we update the core matrix for next feature $\boldsymbol{f}_{i+1}$:

$$\mathcal{M} = \begin{bmatrix} \mathcal{M} + \boldsymbol{\beta}_i^T \boldsymbol{\beta}_i \epsilon_i & -\boldsymbol{\beta}_i / \epsilon_i \\ -\boldsymbol{\beta}_i^T \epsilon_i & 1/\epsilon_i \end{bmatrix}, \tag{2}$$

where $\mathcal{M}$ summarizes the intrinsic linear correlation of the feature set F_i, $\boldsymbol{\beta}_i$ uses $\mathcal{M}$ to calculate the relevance of each previous feature in F_{i-1} and $\boldsymbol{f}_i$, and $F_{i-1}\boldsymbol{\beta}_i$ is the approximate reconstruction of $\boldsymbol{f}_i$ obtained by utilizing F_{i-1} at the current time c. Finally, we obtain $\epsilon = [\epsilon_1, \epsilon_2, \cdots, \epsilon_{\bar{n}}]$.

As shown in Fig. 4(a), given a video sample, we identify several interesting findings. (1) Each peak of the curve of residual error ϵ indicates the local maximal gain of the consecutive variation. These peaks unanimously describe most active sub-actions in a video. (2) Local continuous variation is helpful relative to learning the motion pattern, and features in the accumulative ascending region of ϵ cannot be reconstructed using previous features. Thus, we maintain the monotonically increasing parts of the curve as profits (considered **key clips**). (3) Then, while ϵ is reduced gradually, descending parts can be reconstructed linearly using previous features with the reducing error. These parts always correspond to less-important or redundant intervals in videos, such as blank regions in sub-action-to-sub-action and gloss-to-gloss transitions. We consider such monotonically decreasing parts as **less-important clips**.

The key/less-important feature IDs are used to split a video into serval segments. First, we obtain the temporal recurrent representation of a video, *i.e.*, the outputs of $LSTM_1$, $\{h_i\}$ ($i \in [1, \bar{n}]$). Then, If h_i belongs to a key clip, it is taken directly as a viseme/signeme vector; otherwise, it is included into a to-be-pooled segment. Figure 4(b) unrolls the segmentation along the temporal dimension. Each less-important clip region is concatenated with the first adjacent key clip. We define this concatenation as the to-be-pooled segment $\{h_{i',j}\}$ ($j \in [1, l_{i'}]$), where $h_{i',j}$ is the j-th vector in the i'-th segment, and $l_{i'}$ is the length of the i'-th segment. Here $i' \in [1, n']$ and $\sum_{i'=1}^{n'} l_{i'} = \bar{n}$. Up to this point, this segmentation operation is performed on each video sample. Note that there is no gradient derivation in this subsection.

2) ACS-2: Adaptive Temporal Pooling: To weaken negative effects of less-important clips, we propose several temporal pooling strategies to extract visemes/signemes $\{h'_{i'}\}$ from the abovementioned segments. The candidate temporal pooling strategies for the **visual RGB HRF encoder** are listed as follows.

Note that sequence $\{h_i|i \in [1,\bar{n}]\}$ is reassembled into n' segments $\{h_i\} = \bigcup_{i'\in[1,n']} \{h_{i',j}|j \in [1,l_{i'}]\}$. Each segment $\{h_{i',j}\}$ outputs a pooled vector $h'_{i'}$.

- *No Pooling:* If $l_{i'}$=1, $h_{i',j}$ belongs to a key clip. Here, the pooled vector is equal to $h'_{i'} = h_{i',j}$. The gradient of $h'_{i'}$ at time step i' relative to the original h_i is formulated as follows:

$$\frac{\partial h'_{i'}}{\partial h_i} = \begin{cases} 1, & s.t.\ i = \sum_{k=1}^{i'} l_k \ \&\ l_{i'} = 1 \\ 0, & otherwise \end{cases}. \tag{3}$$

- *Key Pooling:* In each to-be-pooled segment $\{h_{i',j}\}$, the last vector $h_{i',l_{i'}}$ is set as the selected viseme $h'_{i'} = h_{i',l_{i'}}$. This pooling drops the entire less-important clip but retains the gradually recurrent characteristic. The current time step i is equal to the sum of $\{l_1, \cdots, l_{i'}\}$. The gradient of $h'_{i'}$ at time step i' relative to the original h_i is formulated as follows:

$$\frac{\partial h'_{i'}}{\partial h_i} = \begin{cases} 1, & \text{if } \ i = \sum_{k=1}^{i'} l_k \ \&\ l_{i'} \geq 1 \\ 0, & otherwise. \end{cases} \tag{4}$$

- *Mean Pooling:* The mean pooling averages the recurrent characteristics of the segment during the original temporal interval $i \in [\sum_{k=1}^{i'-1} l_k + 1, \sum_{k=1}^{i'} l_k]$. The gradient of $h'_{i'}$ relative to the original h_i is formulated as follows:

$$\frac{\partial h'_{i'}}{\partial h_i} = \begin{cases} \frac{1}{l_{i'}}, & i \in [\sum_{k=1}^{i'-1} l_k + 1, \sum_{k=1}^{i'} l_k] \ \&\ l_{i'} \geq 1 \\ 0, & otherwise. \end{cases} \tag{5}$$

- *Max Pooling:* Maximization of the to-be-pooled segment highlights the prominent response of the segment at each feature dimension. Here $h'^d_{i'} = max_{d\in\{1,2,\cdots,D\}}\{h^d_i\}$, where h^d_i is the d-th value in the vector h_i, and D is the vector dimension. The gradient of $h'_{i'}$ is expressed as follows:

$$\frac{\partial h'^d_{i'}}{\partial h^d_i} = \begin{cases} 1, & i = \underset{i^*}{argmax}\{h^d_{i^*}\} \\ 0, & otherwise, \end{cases}$$
$$s.t.\ i \in [\sum_{k=1}^{i'-1} l_k + 1, \ \sum_{k=1}^{i'} l_k] \ \&\ l_{i'} \geq 1. \tag{6}$$

In addition, motivated by the interpolation concept to smooth the trajectory curve of a human pose, we design another pooling strategy for skeleton data, *i.e.*, linear pooling in the **trajectory HRF encoder**. The trajectory HRF encoder has the same modular flowchart as the visual HRF encoder based on RGB images but achieves the best performance with linear pooling. This means that linear pooling excels at modeling the trajectory curve. The linear pooling conducted on skeleton data is illustrated in Fig. 5.

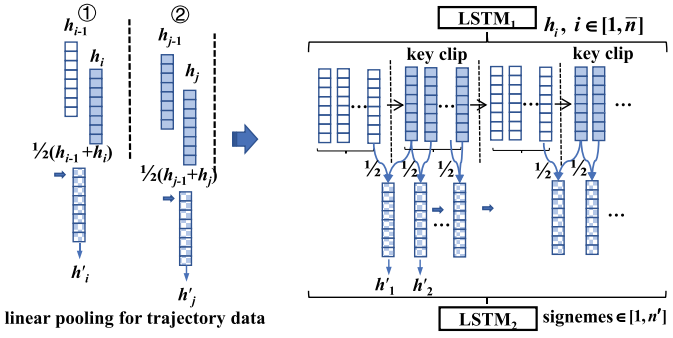


Fig. 5. Linear pooling for trajectory data

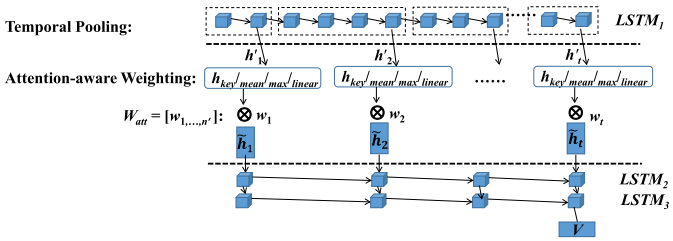


Fig. 6. ACS-3: attention-aware temporal weighting.

- *Linear Pooling:* For each feature in key clips, we average the recurrent characteristic of the current feature and the previous adjacent feature as $h'_{i'} = \frac{h_{i-1}+h_i}{2}$. There are two cases to consider. (1) The junction of a less-important clip and a key clip, where $h'_{i'}$ keeps half of the gradually recurrent characteristic of the previous less-important feature rather than the previous less-important segment. $h'_{i'}$ also adds half of active recurrent characteristic of the current key feature. (2) The average span of two consecutive recurrent features in the key clips. Thus, the gradient of $h'_{i'}$ at time step i' relative to the original h_i is calculated as follows:

$$\frac{\partial h'_{i'}}{\partial h_i} = \begin{cases} \frac{1}{2}, & i = \sum_{k=1}^{i'} l_k - 1 \ \&\ l_{i'} > 1 \\ \frac{1}{2}, & i = \sum_{k=1}^{i'} l_k \ \&\ l_{i'} = 1 \\ 0, & otherwise. \end{cases} \tag{7}$$

3) ACS-3: Attention-Aware Weighting: As signs can be identified easily by crucial action patterns, we propose attention-aware temporal weighting to strengthen the effects of active visemes and signemes, where active visemes and signemes are assigned greater. In other words, the proposed attention balances their impacts. It also measures the impacts of all source positions, but emphasizes different contributions during the temporal transition. The temporal attention weighting is shown in Fig. 6. Matrix $\boldsymbol{W}_{att} \in \mathbb{R}^{n'}$ is automatically learned by training $\widetilde{h}_{i'} = w_{i'} \cdot h'_{i'}$ in our end-to-end deep architecture. Here, $h'_{i'}$ is one type of the previously mentioned outputs h_{key}, h_{mean}, h_{max} and h_{linear} in Fig. 4(b), where $i' \in [1, n']$. At each time step i', the gradient of $\widetilde{h}_{i'}$ relative to

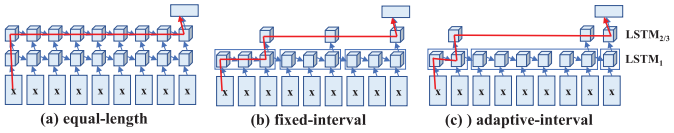


Fig. 7. Comparison on different hierarchical encoding networks. Taking a two-layer LSTM network example in [14], the red line shows one of the paths from the input at $t = 1$ to the final output of the video. Here, there are ten time steps in (a), six in (b) and five in (c). Compared to (a) and (b), our hierarchical architecture (c) is flexible.

the original $h'_{i'}$ is formulated as follows:

$$\frac{\partial \widetilde{h}_{i'}}{\partial h'_{i'}} = w_{i'}, \text{ at each time step } i' . \tag{8}$$

B. HRF Encoder: Fundamentals Function and Analysis

Overall, we divide an input feature sequence $F = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{\bar{n}}\}$ into several segments $(\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_{l_1})$, $(\boldsymbol{f}_{l_1+1}, \boldsymbol{f}_{l_1+2}, \cdots, \boldsymbol{f}_{l_1+l_2})$, $\cdots$, $(\cdots, \boldsymbol{f}_{\bar{n}})$. Here, l_k to denotes the length of the k-th segment ($k \in [1, n']$). Taking a two-layer LSTM network example in [14], Fig. 7(a) shows an equal-length stacked multilayered network with $l_k = 1$, such as S2VT [54]. In Fig. 7(b), the sequence is compressed into $\lceil \frac{\bar{n}}{l_k} \rceil$ with a fixed interval $l_k = 3$, such as HRNE [14]. Fig. 7(c) shows our hierarchical model based on the ACS scheme with adaptive length l_k ($l_1 = 2$, $l_2 = 6$ and $l_3 = 1$). Note that the number of segments n' in the proposed model is adaptive. In Fig. 7(c), the red line shows how the path of the input at $t = 1$ flows to the final output. In this path, the upper layer focuses on the temporal correlation between adjacent segments, and the bottom layer explores the temporal correlation within each segment. Figures 7(b) and (c) indicate that the condensed encoding architecture builds multiple time-scale summarizations.

The ACS scheme ("ACS-1," "ACS-2," and "ACS-3") is embedded into the HRF encoder. Here, "ACS-1: variable-length key clip mining RSS" is denoted as function Ω_{rss}, "ACS-2: adaptive temporal pooling" is denoted as $\mathscr{G}_{tpool}$ and "ACS-3: attention-aware weighting" is denoted as ζ_{att}. The adaptive HRF encoder is formulated as follows:

$$\begin{aligned}
V &= HLSTM\left(\zeta_{att}\left[\mathscr{G}_{tpool}\left\{\Omega_{rss}(\boldsymbol{f}_1, \cdots, \boldsymbol{f}_{\bar{n}})\right\}\right]\right) \\
&= HLSTM\left(\zeta_{att}\left[\mathscr{G}_{tpool}\left\{\boldsymbol{f}_1, \cdots, \boldsymbol{f}_{\bar{n}}\right\}\big|_{\{l_1,\cdots,l_{n'}\}}\right]\right) \\
&= LSTM_{2,3}\left(\zeta_{att}\left[\mathscr{G}_{tpool}\left\{LSTM_1(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{\bar{n}})\big|_{\{l_1,\cdots,l_{n'}\}}\right\}\right]\right) \\
&= LSTM_{2,3}\Big(\zeta_{att}\Big[\mathscr{G}_{tpool} \\
&\quad \Big\{LSTM_1(\underbrace{\boldsymbol{f}_1,\cdots,\boldsymbol{f}_{l_1}),(\boldsymbol{f}_{l_1+1},\cdots,\boldsymbol{f}_{l_1+l_2}),\cdots,(\cdots,\boldsymbol{f}_{\bar{n}})}_{n'}\Big\}\Big]\Big) \\
&= LSTM_{2,3}\left(\zeta_{att}\left[\mathscr{G}_{tpool}\left\{\underbrace{(h_1,\cdots,h_{l_1}),\cdots,(\cdots,h_{\bar{n}})}_{n'}\right\}\right]\right) \\
&= LSTM_{2,3}\left(\zeta_{att}[h'_1, \cdots, h'_{n'}]\right) \\
&= LSTM_{2,3}\left(\widetilde{h}_1, \cdots, \widetilde{h}_{n'}\right) \\
&= (v_1, \cdots, v_{n'}) = v_{n'},
\end{aligned} \tag{9}$$

where parameters $\bar{n}$ and n' are defined in Table I. To be specific, Ω_{rss} differentiates key and less-important clips. Based on the segmentation of key and less-important clips, we split a video into serval segments with variable lengths $(l_1, \cdots, l_{n'})$. Variables n' and $(l_1, \cdots, l_{n'})$ are online adaptively calculated for different video samples. After implementation $\mathscr{G}_{tpool}$, each segment outputs a viseme or signeme. Thus we can obtain visemes/signemes $\{h'_1, \cdots, h'_{n'}\}$. Using ζ_{att}, we weight visemes/signemes into $\{\widetilde{h}_1, \cdots, \widetilde{h}_{n'}\}$ as the inputs for $LSTM_2$.

In other words, the HRF encoder implements functions as follows: $\Omega_{rss} \rightarrow LSTM_1 \rightarrow \mathscr{G}_{tpool} \rightarrow \zeta_{att} \rightarrow LSTM_2 \rightarrow LSTM_3 \rightarrow V = v_{n'}$, where $v_{n'}$ is a combination of the last outputted encoding vectors of $LSTM_2$ and $LSTM_3$. The adaptivity of function Ω_{rss} results in $\mathscr{G}_{tpool}$ and the entire HRF encoder are adaptive. In other words, the HRF encoder reduces computational complexity by compacting encoding lengths for different videos, while retaining nonlinearity and adaptability.

C. Stacked Decoding Network

Here, we clarify a single stacked decoding procedure. Following either the visual RGB or trajectory HRF encoder, each visual semantic vector V arrives at the decoding stage. Here, $LSTM_2$ and $LSTM_3$ remain stacked for sentence generation, where $LSTM_2$ transits the visual content continuously and $LSTM_3$ decodes the textual semantics sequentially. $LSTM_3$ outputs glosses until meeting the ending tag $<EOS>$ (".").

According to the recurrent characteristics of HRF, the current predicted gloss is influenced by its previous gloss. At each decoding time step, $LSTM_3$ sequentially feeds the word embedding vector of the previous gloss to predict the current gloss. During training, $LSTM_3$ feeds the word embedding of the previous ground truth gloss into the sequential learning process. In the practical application scenarios without ground truth glosses, it takes the previous predicted gloss.

The output z_t of $LSTM_3$ is used to predict gloss y_t using Eq. 10 (a softmax function). We select the gloss with the maximum value in the probability vector $p(y_t|z_t)$ and consider it as y_t. Here, $p(y_t|z_t)$ is a C-dim vector and C is the vocabulary size. Each value in $p(y_t|z_t)$ indicates the relevance probability of the c-th sign gloss ($c \in [1, C]$) at decoding time t. Note that each HRF decoder has its own $LSTM_2$ and $LSTM_3$ under different data-modalities (*i.e.*, RGB and skeleton channels).

$$p(y_t|V, y_{t-1}) = p(y_t|z_t) = \frac{exp(\boldsymbol{W}_y z_t)}{||exp(\boldsymbol{W}_y z_t)||_1}. \tag{10}$$

Here, $||\cdot||_1$ denotes L_1-norm, and $\boldsymbol{W}_y$ is a trainable matrix parameter that transforms embedding feature z_t to probability vector y_t over C sign classes.

D. Query-Adaptive Fusion for Sentence Generation

Based on the decoding probability vectors $p(y_t|z_{t,rgb})$ and $p(y_t|z_{t,ske})$ under the RGB and skeleton channels at each time t, respectively, we propose a fusion scheme to explore their complementarity. The fusion module is an online heuristic algorithm that utilizes the deep probability scores to fuse and predict gloss words in sequence for sentence generation.

We attempt to assign different weights to $p(y_t|z_{t,rgb})$ and $p(y_t|z_{t,ske})$ at decoding time t. Here, we adopt the idea that a good probability vector should have the highest score on the correct class and a lower score on irrelevant classes [55]. In other words, the sorted probability vector is considerably more discriminative to identify sign class if it has a much sharper curve. (1) We sort the probability vector in descending order and perform min-max normalization on it. We denote the two types of sorted gloss probability vectors as $p'_{t,rgb}$ and $p'_{t,ske}$. (2) We then calculate the curve area of the sorted vector $Area_{p'_{t,k}}$. The weight is inversely proportional to the area of the curve of the normalized sorted probability vector (Eq. 11). Note that the weighting phase is query-adaptive and unsupervised. In addition, it is tightly related to $p(y_t|z_{t,k})$ ($k \in \{rgb, ske\}$).

$$\begin{cases} p'_{t,k} = \dfrac{p'_{t,k} - \Delta_{min}(p'_{t,k})}{\Delta_{max}(p'_{t,k}) - \Delta_{min}(p'_{t,k})} \\ w_{t,k} = \dfrac{1/Area_{p'_{t,k}}}{\sum\limits_{k\in\{rgb,ske\}} 1/Area_{p'_{t,k}}}. \end{cases} \tag{11}$$

Finally, as the superiority of the product rule stated in biometric multi-modality fusion [55], [56], our fusion adopting this product rule is formulated as follows:

$$\begin{aligned} p(y_t|z_{t,fusion}) &\propto \sum_{k\in\{rgb,ske\}} p(y_t|z_{t,k})^{w_{t,k}} \\ &\propto \sum_{k\in\{rgb,ske\}} w_{t,k}\cdot\log(p(y_t|z_{t,k})). \end{aligned} \tag{12}$$

At each decoding time t, the maximum value in the fused probability vector $p(y_t|z_{t,fusion})$ indicates the gloss class y_t:

$$\begin{aligned} y_t = c_t^* &= \arg\max_{c_t^*\in C}[p(y_t|z_{t,fusion})] \\ &s.t. \sum_k w_{t,k} = 1. \end{aligned} \tag{13}$$

E. HRF Loss Function

We employ the entropy of the generated sentences to learn the whole model parameter $\psi^* = \arg\min_\psi \mathcal{L}$, where $\mathcal{L}$ is formulated according to Eq. 14. The entropy can be used to solve a single RGB-based SLT, the skeleton (Depth)-based SLT, and the RGB-skeleton-based SLT. If $p(y_t|V, y_{t-1}) = p(y_t|z_{t,fusion})$, $\mathcal{L}$ is an end-to-end fusion framework for RGB-skeleton-based SLT. Otherwise, if $p(y_t|V, y_{t-1}) = p(y_t|z_{t,rgb})$ or $p(y_t|z_{t,ske})$, $\mathcal{L}$ is used for single HRF-S training.

$$\mathcal{L} = -\sum_{t=1}^{m} p(y_t|V, y_{t-1}; \psi)\log p(y_t|V, y_{t-1}; \psi). \tag{14}$$

IV. Function Analysis and Derivative Discussions

The entire model parameter ψ is differentiable and trainable. We first introduce the derivation procedure of a single HRF flowchart (HRF-S), and then analyze the two-stream parallel HRF-Fusion. While loss $\mathcal{L}$ is backpropagated, model parameter ψ is updated with the gradient of $\mathcal{L}$.

$$\begin{aligned} \nabla_\psi \mathcal{L} &= \sum_{\hat{t}=1}^{m} \nabla_{\psi_{\hat{t}}} \mathcal{L} \\ &= \nabla_{\psi_m}\mathcal{L} + \nabla_{\psi_{m-1}}\mathcal{L} + \cdots + \nabla_{\psi_1}\mathcal{L}. \end{aligned} \tag{15}$$

We focus on the adaptability of the HRF model. The function Ω_{rss} is merely a video segmentation operation, and its partial derivation is embedded into function $\mathscr{G}_{tpool}$. For $\mathscr{G}_{tpool}$, the gradient inductions of different pooling strategies are given in Eqs. 3-8 (Section III-A.2). The decoding parameter W_y is a softmax function under L_1-normalization. The remaining model parameter ψ is set into the combination of notations $\{\mathbf{net}^1, \mathbf{net}^2, \mathbf{net}^3, \mathbf{W_{att}}\}$ corresponding to $LSTM_1$, $LSTM_2$, $LSTM_3$ and function ζ_{att} respectively.

We clarify the backward operational stream of HRF-S: $\partial\mathcal{L} \rightarrow \partial p_t \xrightarrow{\partial \mathbf{W_y}} \partial h_t^3 \rightarrow \partial \mathbf{net}_t^3 \rightarrow \partial h_t^2 \rightarrow \partial \mathbf{net}_t^2 \rightarrow \partial h_t^1 \rightarrow \mathbf{W_{att}} \rightarrow \partial h'_t \rightarrow \mathscr{G}_{tpool} \xrightarrow{\Omega_{rss}} \partial h_i \rightarrow \partial \mathbf{net}_i^1 \rightarrow \{\boldsymbol{f}_i\}$, where h_t^1 is the output after the $LSTM_1$ and ACS, and h_t^2 and h_t^3 denote the outputs of $LSTM_2$ and $LSTM_3$ at time step t, respectively. Here, $h_t^1 = \widetilde{h}_t$ in Eq. 9 and $h_t^3 = z_t$ in Eq. 10. To simplify the derivation procedure, we define t in the following statements to denote both the encoding and decoding time of $LSTM_{2\sim3}$, and i simply to denote the encoding time of $LSTM_1$. Note that $\hat{t}$ in Eq. 15 just indicates the decoding time of $LSTM_3$.

A. Derivation of HRF-S

- *Backpropagation of* $\frac{\partial\mathcal{L}}{\partial\mathbf{net}_t^3}$*:*

$LSTM_3$ in the proposed HRF is used as a basic LSTM. Based on the principle of LSTM, we give the conclusions as follows. We define the error term at time step t as $\delta_{T\rightarrow t} \overset{def}{=} \sum\limits_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \dfrac{\partial\mathcal{L}}{\partial\mathbf{net}_{*,t}^3}$, and the detailed backpropagation of $LSTM_3$ is calculated as follows:

$$\begin{aligned} \delta_{\mathbf{o},T\rightarrow t} &= \delta_{T\rightarrow t}\circ\tanh(\mathbf{c}_t)\circ\mathbf{o}_t\circ(1-\mathbf{o}_t) \\ \delta_{\mathbf{f},T\rightarrow t} &= \delta_{T\rightarrow t}\circ\mathbf{o}_t\circ(1-\tanh(\mathbf{c}_t)^2)\circ\mathbf{c}_{t-1}\circ\mathbf{f}_t\circ(1-\mathbf{f}_t) \\ \delta_{\mathbf{i},T\rightarrow t} &= \delta_{T\rightarrow t}\circ\mathbf{o}_t\circ(1-\tanh(\mathbf{c}_t)^2)\circ\tilde{\mathbf{c}}_t\circ\mathbf{i}_t\circ(1-\mathbf{i}_t) \\ \delta_{\tilde{\mathbf{c}}},\ T\rightarrow t &= \delta_{T\rightarrow t}\circ\mathbf{o}_t\circ(1-\tanh(\mathbf{c}_t)^2)\circ\mathbf{i}_t\circ(1-\tilde{\mathbf{c}}^2). \end{aligned} \tag{16}$$

where $T = m$ and $\{\mathbf{f_t}, \mathbf{c_t}, \mathbf{i_t}, \mathbf{o_t}\}$ indicates the intrinsic gate cells of LSTM.

- *Backpropagation of* $\frac{\partial\mathcal{L}}{\partial\mathbf{net}_t^3} \rightarrow \frac{\partial\mathcal{L}}{\partial\mathbf{net}_t^2}$*:*

In the proposed multiply-layer architecture for HRF, $\{h_t^2\}$ is the input of $LSTM_3$, and the output of $LSTM_2$. Here, $\{h_t^2\} = f(\mathbf{net}_{*,t}^2)$, where $f(\cdot)$ indicates the activation function of the LSTM cell. Thus, the derivatives formula of $\frac{\partial\mathcal{L}}{\partial\mathbf{net}_{*,t}^2}$ is expressed as follows:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{net}_t^2} &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \frac{\partial\mathcal{L}}{\partial\mathbf{net}_{*,t}^3}\frac{\partial\mathbf{net}_{*,t}^3}{\partial h_t^2}\frac{\partial h_t^2}{\partial\mathbf{net}_{*,t}^2} \\ &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \delta_{*,T\rightarrow t}\mathbf{W}_{*,t}^3\circ f'(\mathbf{net}_{*,t}^2), \end{aligned} \tag{17}$$

where $\delta^3_{*,T\to t}$ is calculated in $\frac{\partial \mathcal{L}}{\partial \mathbf{net}^3_t}$. $\mathbf{W}^3_{*,t}$ is the model parameter of the gate cells in $\mathbf{net}^3_t$, and $f'(\cdot)$ denotes the partial derivative function.

- *Backpropagation of* $\frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_t} \to \frac{\partial \mathcal{L}}{\partial \mathbf{net}^1_i}$*:*

Due to the adaptivity of the ACS scheme with the Ω_{rss}, $\mathscr{G}_{tpool}$, and ζ_{att} functions, the derivation is very different from Eq. 17 ($\frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_i}$). We adopt the gradient formulas Eqs. 3∼7 of function $\mathscr{G}'_{tpool}$ ($\frac{\partial h'_t}{\partial h_i} = \frac{\partial h'_{i'}}{\partial h_i}$) and Eq. 8 of function ζ'_{att} ($\frac{\partial h^1_t}{\partial h'_t} = \frac{\partial \widetilde{h}_{i'}}{\partial h'_{i'}}$) to solve $\frac{\partial \mathbf{x}^2_t}{\partial \mathbf{net}^1_i}$, where $i' \equiv t$. We express the partial derivatives of $\frac{\partial \mathcal{L}}{\partial \mathbf{net}^1_i}$ as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{net}^1_i} &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_{*,t}} \times \frac{\partial \mathbf{net}^2_{*,t}}{\partial h^1_t} \frac{\partial h^1_t}{\partial h'_t} \frac{\partial h'_t}{\partial h_i} \frac{\partial h_i}{\partial \mathbf{net}^1_i} \right\} \\ &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_{*,t}} \cdot \mathbf{W}^2_{*,t} \cdot w^{att}_t \circ \frac{\partial h'_t}{\partial h_i} f'(\mathbf{net}^1_{*,i}) \right) \\ &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \left\{ w^{att}_t \cdot \frac{\partial h'_t}{\partial h_i} \cdot \delta_{*,T\to t} \cdot \mathbf{W}^3_{*,t} \circ f'(\mathbf{net}^2_{*,t}) \right. \\ &\qquad \left. \circ \mathbf{W}^2_{*,t} \circ f'(\mathbf{net}^1_{*,i}) \right\}, \end{aligned} \tag{18}$$

where t is equal to i' in Eqs. 3-8.

- *Partial Derivatives of* $\frac{\partial \mathcal{L}}{\partial W_{att}}$*:*

In the same manner as $\frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_t}$, the partial derivatives of $\frac{\partial \mathcal{L}}{\partial W_{att}}$ are expressed as follows.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{W_{att}}} &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_{*,t}} \times \frac{\partial \mathbf{net}^2_{*,t}}{\partial h^1_t} \frac{\partial h^1_t}{\partial \mathbf{W_{att}}} \right\} \\ &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{net}^2_{*,t}} \times \mathbf{W}^2_{*,t} \cdot w^{att}_t \right\} \\ &= \sum_{*\in\{\mathbf{f_t},\mathbf{c_t},\mathbf{i_t},\mathbf{o_t}\}} \left\{ w^{att}_t \cdot \delta^3_{*,T\to t} \cdot \mathbf{W}^3_{*,t} \circ \mathbf{W}^2_{*,t} \circ f'(\mathbf{net}^2_{*,t}) \right\}. \end{aligned} \tag{19}$$

B. Derivation on Two-Stream HRF

When training two single HRF-S models separately with $p(y_t|z_{rgb})$ and $p(y_t|z_{ske})$, and fusing the two deep scores to generate a sentence, we obey the reduction of HRF-S (Section IV-A). The fusion is not involved in the training process. If setting $p(y_t|z_{t,fusion})$ in Eq. 14, the proposed HRF-Fusion is changed to an end-to-end training framework.

As given in Eq. 20, the value of $\nabla_\psi \mathcal{L}_{fusion}$ with $\partial p_{t,fusion} = \partial p(y_t|z_{t,fusion})$, differs from HRF-S $\nabla_\psi \mathcal{L}$ with $\partial p_t = \partial p(y_t|z_{t,rgb})$ or $\partial p_t = \partial p(y_t|z_{t,ske})$. In contrast, with module $LSTM_3$ in the backpropagation, $\nabla_\psi \mathcal{L}_{fusion}$ transforms into two streams. For either RGB or skeleton channels, the gradient of $\nabla \mathcal{L}_{fusion}$ is closely related to $\nabla \mathcal{L}_{rgb}$ and $\nabla \mathcal{L}_{ske}$, and their fused weight, as in Eqs. 21 and 22.

TABLE II
TWO SPLITTING STRATEGIES FOR THE DATASET

		Signers	Sentences	Samples
Split I	Train	40	100	$40 \times 100 = 4,000$
	Test	10	100	$10 \times 100 = 1,000$
Split II	Train	50	94	$50 \times 94 = 4,700$
	Test	50	6	$50 \times 6 = 300$

$$\frac{\partial \mathcal{L}_{fusion}}{\partial p_{t,fusion}} = -(\log p_{t,fusion} + 1). \tag{20}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{fusion}}{\partial z_{t,rgb}} &= \frac{\partial \mathcal{L}_{fusion}}{\partial p_{t,fusion}} \times \frac{\partial p_{t,fusion}}{\partial p_{t,rgb}} \times \frac{\partial p_{t,rgb}}{\partial z_{t,rgb}} \\ &= -(\log p_{t,fusion} + 1) \cdot w_{t,rgb} \times \frac{\partial p_{t,rgb}}{\partial z_{t,rgb}} \\ &= \frac{w_{t,rgb} \cdot (\log p_{t,fusion} + 1)}{(\log p_{t,rgb} + 1)} \times \frac{\partial \mathcal{L}_{rgb}}{\partial z_{t,rgb}}. \end{aligned} \tag{21}$$

$$\frac{\partial \mathcal{L}_{fusion}}{\partial z_{t,ske}} = \frac{w_{t,ske} \cdot (\log p_{t,fusion} + 1)}{(\log p_{t,ske} + 1)} \times \frac{\partial \mathcal{L}_{ske}}{\partial z_{t,ske}}. \tag{22}$$

C. Summary

The parameters of $LSTM_{1\sim3}$ and ζ_{att} are differentiable and trainable in the deep framework, which employs the adaptive ACS scheme. In the paper, we derive the proposed HRF based on the LSTM cell. The proposed model can also utilize other RNN units, *e.g.*, RNN, GRU, BRGU, and BLSTM. The theoretical fundamental and derivatives are similar. Moreover, due to the compressed encoding time steps, the model is easier to train via stochastic gradient methods using Back-propagation Through Time (BPTT).

V. EXPERIMENT

A. Experimental Setup

1) Dataset: We experimented on a Chinese sign language dataset that includes 100 common sentences[1], and each sentence is spoken in sign language by 50 signers (5000 videos in total). In this dataset, the vocabulary size is 179, and each sentence contains four to eight (average five) sign glosses (phases).

To validate the proposed approach, we split the dataset according to two strategies (Table II). (1) **Split I - signer independent test**. This splits video samples of 40 signers as the training set and the remaining 10 signers as the test set. Note that sentences in the training and test sets are the same; however, the signers differ. (2) **Split II - unseen sentence test**. This strategy selects six sentences as the test set, and the remaining 94 sentences are taken as the training set. The split follows the criteria that glosses in the six sentences have appeared separately in the remaining 94 sentences; however, each gloss' context, occurrence order, and application scenarios differ entirely.

[1] http://mccipc.ustc.edu.cn/mediawiki/index.php/SLR_Dataset

2) Evaluation Metrics: **Precision** is the ratio of correct sentences with respect to the total number of sentences. When a generated sentence is exactly the same as the reference, it is considered correct. ***Acc-w*** is the average ratio of correct words to reference words in a sentence. The word error rate **WER** [11] measures the smallest number of operations required to transform a generated sentence into the reference. In addition, we adopt textual semantics metrics commonly used in NLP, NMT, and image description, *i.e.*, **BLEU**, **METEOR**, **ROUGE-L**, and **CIDEr**.

3) Model Setting: We applied C3D [16] to extract 3D CNN features from each subsequence clipped every 16 frames with an eight-frame overlap. Thus, there is $N = 8 \times n$, where N is the number of frame-level skeleton descriptors and n is the number of clip-level 3D CNN features. In the initial step, we pretrained C3D on an isolated SLR dataset [57] using the stochastic gradient descent optimizer with a learning rate of 0.001, momentum of 0.9, and weight decay of 5×10^{-5}. For skeleton data, we directly concatenated the coordinates of four joint points, *i.e.*, the left elbow, right elbow, left hand, and right hand, as the skeleton descriptor, which is a 12-dim vector.

In the pooling setting, the proposed model HRF selects linear pooling for skeleton data. With RGB image data, key, mean, and max pooling demonstrate are adopted. As for RGB data, the proposed HRF has the best performance with mean pooling for Split I and max pooling for Split II. The experimental results and analysis are given in the following model validation.

Our code was implemented using the Tensorflow platform, which must feed features into each standard batch with a fixed temporal length n' in LSTM learning. Note that some samples in the dataset have overlong viseme sequences. To train the model easily and quickly, we set a smaller n' as the batch size. In these experiments, we set $n' = l_{min,ske}$ and $n' = l_{ave,RGB}$ for different modal data, where $l_{ave,RGB}$ is the average number of 3D CNN features of all training videos and $l_{min,ske}$ is the minimum number of skeleton features of all training videos. We compressed overlong viseme/signeme sequences to length n' via systematic sampling and fill zero-padding vectors into the short viseme/signeme sequences to meet the length n'. Finally, limited to long sequence videos in the dataset, the end-to-end fusion training process is slow. Thus, we train two single HRF-S models separately, *i.e.*, visual RGB and trajectory HRF-S. HRF-Fusion outputs the generated sentence by the fusion phase (Eq. 13).

To verify the effectiveness of each modular in the proposed HRF, we denote **HRF-S** as the architecture of a single feature stream of HRF without temporal attention. **HRF-S-att** adds temporal attention to HRF-S, and **HRF-Fusion** represents the model with the fusion phase.

4) Compared Approaches: We compared the proposed model to the **LSTM&CTC** model[2], which solves feature-level classification and is widely used in sequential speech recognition. As the proposed HRF belongs to the encoder-decoder framework, we compared it to the following similar models:

[2]https://github.com/baidu-research/warp-ctc

TABLE III
COMPARISON OF RGB-BASED SLT USING SPLIT I

Model	Precision
LSTM&CTC (Warp-ctc)	0.858
S2VT [54]	0.897
S2VT (3-layer)	0.903
LSTM-E [58]	0.882
LSTM-Attention [7]	0.851
LSTM-global-Attention [59]	0.858
HAN [61]	0.793
LS-HAN [51]	0.827
HRNE [14]	0.910
HRF-S	**0.924**
HRF-S-att	**0.929**
Model	**WER**
LSTM+CTC	0.119
CTF [48]	0.112
HRF-S	**0.107**
HRF-S-att	**0.102**

S2VT (a standard two-layer stacked encoder-decoder architecture with equal length [54]); **LSTM-E** (inputs deep 2D and 3D CNN features with mean pooling for high-level semantic embedding [58]); **LSTM-Attention** (embeds an attention mechanism to capture the temporal relationship among frames [7]); and **LSTM-global-Attention** (explores a global attention mechanism for NMT [59]).

To verify the proposed hierarchical network architecture, we also compared the HRF model to **HRNE** [14], which is a compact hierarchical network with a fixed-length interval (Fig. 7(b)). HRNE replaces key clip mining function $\mathbf{\Omega}_{rss}$ with systematic sampling in HRF. S2VT achieved the closest performance to the proposed HRF model in our experiments; thus, we extended it to **S2VT (3-layer)**. Similar to HRF, S2VT (3-layer) includes $LSTM_1$ in the encoding stage of S2VT. Note that both S2VT and S2VT (3-layer) belong to the hierarchical network with equal-length encoding length as shown in Fig. 7(a). **HRNN-Ske** [60] is a hierarchical network designed for skeleton data.

In addition, **CTF** [48] fuses different visual feature embeddings to output the generated sentence. Based on multimodal data, **ELM** [52] embeds two fusion schemes separately, *i.e.*, feature fusion (ELM-Early) and score fusion (ELM-Late) into the deep learning framework. Both ELM-Early and ELM-Late implement the fusion from multimodal data.

B. Evaluation of the Visual RGB HRF-S

1) Main Comparison of Seen Sentence Test: As shown in Table III, compared to LSTM&CTC, the proposed model achieves better performance. The LSTM&CTC framework outputs and merges the feature-level labels, which is a classification solution without any textual semantics learning. In contrast, the HRF decoder learns the textual embedding and has a positive effect.

We obtained four conclusions from the comparison of the encoder-decoder frameworks. (1) Compared to S2VT with the fixed-length stacked LSTM, the proposed flexible-length HRF architecture achieves better performance. (2) LSTM-E implements average pooling on the entire feature sequence,

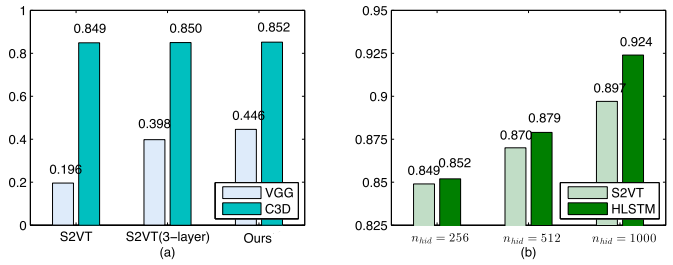


Fig. 8. Precision comparison using Split I. (a) Different visual features with $n_{hid} = 256$. (b) Different dims of the hidden state of LSTM with C3D features.

while the proposed model pools on subsequences. The experimental result indicates that adaptive temporal pooling in HRF demonstrates better performance compared to LSTM-E. (3) Classic attention strategies that calculate on the relevance of source positions and a target position, does not work well with our dataset, such as LSTM-Attention, LSTM-global-Attention, HAN, and LS-HAN. In contrast, our attention strategy emphasizes the accumulative weighting transition of source positions along the temporal dimension, which means that SLT has a dense correlation among consecutive features under sign linguistics. Our attention highlights important source positions, and spreads the accumulative influence until the current position. (4) Among the different architectures in the hierarchical networks, HRNE and S2VT (3-layer) demonstrate comparable performance; however, HRF-S shows better performance, while HRF-S-att achieves the best performance. This is attributed to adaptive key clip selection and temporal attention-aware weighting.

a) Evaluation of 2D and 3D CNN Features: Here, we used the VGG [62] and C3D models as feature extractors to verify the performance of 2D and 3D CNN features, respectively. Similar to C3D, the VGG model was pretrained using the isolated sign language dataset [57]. Limited by the high dimension (4096-dim) and a large number of VGG features, we experimented with the dimension of the LSTM's hidden states $n_{hid} = 256$ under the constraint of GPU calculation memory. Figure 8(a) shows that the C3D feature is better than the VGG feature because C3D has the advantage of action capturing for SLT. Under 3D CNN features, the gradient disappearance defect for long sequence learning is alleviated. Thus, we used the C3D model as the feature extractor in our subsequent experiments.

b) Evaluation of Different LSTM Hidden State Numbers: To test precision with different LSTM settings, we set the hidden state number n_{hid} to 256, 512, and 1000. As shown in Fig. 8(b), precision improved as n_{hid} increased. In addition, when n_{hid} was small, the experimental results were unstable under multiple random tests. However, with $n_{hid} = 1000$, the results were stable. This may be due to network overfitting learning with fewer hidden states. Note that many different sign patterns are still undistinguishable under similar gestures and pose trajectories, and many more neural units are required to promote the learning capability of LSTM. Thus, we selected the large value ($n_{hid} = 1000$) for the LSTM parameter setting.

c) Evaluation of Different Pooling Strategies: Table IV verifies the different characteristics of various temporal pool-

TABLE IV

COMPARISON OF DIFFERENT POOLING STRATEGIES BASED ON C3D FEATURES

Pooling strategy	Precision on Split I	$Acc - w$ on Split II
Key pooling	0.920	0.479
Mean pooling	**0.924**	0.458
Max pooling	0.912	**0.482**

TABLE V

PRECISION COMPARISON OF DIFFERENT ENCODER FRAMEWORKS BASED ON RGB IMAGE DATA USING SPLIT I

Model	Precision
S2VT ($\bar{n} = n' = 21$) [54]	0.897
S2VT ($\bar{n} = n' = 66$) [54]	0.850
S2VT (3-layer, $\bar{n} = n' = 21$)	0.903
S2VT (3-layer, $\bar{n} = n' = 66$)	0.854
HRNE [14]	0.910
HRF-S	**0.924**
HRF-S-att	**0.929**

ings. As shown, key pooling maintains its recurrent characteristics along the temporal dimension, mean pooling averages its recurrent outputs, and max pooling highlights its prominent responses. Therefore, mean pooling was the most effective at remembering the average response of observed sentences. For the unseen sentence test, max pooling was the best at retaining the maximum response of discriminative gestures of sign glosses. Note that key pooling achieves a mediate performance. We used mean and max pooling for RGB-based Split I and Split II in subsequent experiments.

d) Evaluation of Different n' Settings: In the training dataset, the average and maximum lengths of the C3D features of videos are 21 and 66, respectively. S2VT is a network of stacked LSTMs of equal-length. Thus, we can set $\bar{n} = n' = 21$ or 66 in S2VT. If $\bar{n} = n' = 66$, S2VT retains all sequential features. When $\bar{n} = n' = 21$, it samples features equidistantly into a compact subsequence of features. As shown in Table V, S2VT obtained better results with $\bar{n} = n' = 21$. In other words, a compact representation helps achieve better performance than original features. In addition, compared to S2VT and S2VT (3-layer), HRNE and the proposed HRF demonstrate flexible compact representations with variable n' for different video samples. The experimental results also verify their effectiveness.

e) Evaluation of Different Hierarchical Frameworks: S2VT is a two-layer stacked encoder-decoder network, and S2VT (3-layer) is an extensible S2VT that adds $LSTM_1$ to the encoding stage. Note that S2VT and S2VT (3-layer) have equal-length encoding time steps, as shown in Fig. 7(a), HRNE is a hierarchical compressible network (Fig. 7(b)), and the proposed HRF-S is an adaptive hierarchical compressible network (Fig. 7(c)).

As shown in Table V, a network of equal length is not a good choice, such as S2VT and S2VT (3-layer). S2VT (3-layer) is better than S2VT because it incorporates recurrent characteristic via the extensible top $LSTM_1$. HRNE and the propsed HRF perform better than S2VTs due to the design of hierarchical compressible encoding network. Among these networks, HRF holds the best performance as it uses a flexible

TABLE VI
EVALUATION OF SEEN SENTENCE TRANSLATION OF RGB-BASED SLT USING SPLIT I

	Precision	CIDEr	BLEU-4	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR
LSTM&CTC	0.858	8.632	0.899	0.907	0.918	0.936	0.940	0.646
S2VT [54]	0.897	8.512	0.874	0.879	0.886	0.902	0.904	0.642
S2VT (3-layer)	0.903	8.592	0.884	0.889	0.896	0.911	0.911	0.648
HRNE [14]	0.910	8.907	0.911	0.916	0.922	0.935	0.938	0.683
HRF-S	**0.924**	**9.019**	**0.922**	**0.927**	**0.932**	**0.942**	**0.944**	**0.699**
HRF-S-att	**0.929**	**9.084**	**0.928**	**0.933**	**0.938**	**0.948**	**0.951**	**0.703**

TABLE VII
EVALUATION OF UNSEEN SENTENCE TRANSLATION OF RGB-BASED SLT USING SPLIT II

	Acc-w	CIDEr	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR	WER
LSTM&CTC	0.332	0.241	0.039	0.124	0.343	0.362	0.111	0.757
S2VT [54]	0.457	0.479	0.135	0.258	0.466	0.461	0.189	0.670
S2VT (3-layer)	0.461	0.477	0.145	0.265	0.475	0.465	0.186	0.652
HRNE [14]	0.459	0.476	0.185	0.293	0.463	0.462	0.173	**0.630**
HRF-S	**0.482**	**0.561**	**0.195**	**0.315**	**0.487**	**0.481**	**0.193**	0.662
HRF-S-att	**0.506**	**0.605**	**0.207**	**0.330**	**0.508**	**0.503**	**0.205**	0.641

ACS scheme to condense compressible with adaptive intervals in the hierarchical deep framework. Furthermore, with temporal attention, HRF-S-attn achieves the best performance.

f) Summary of Experiments on Seen Sentence Test: Table VI concludes the experimental comparison on RGB image data. The performances of precision and semantic metrics are consistent. These results verify the robust capability of our HRF model for RGB-based SLT on seen sentence under signer-independence test again.

2) Evaluation on Unseen Sentence Test: As shown in Table VII, it is much more difficult for SLT on unseen sentences than on seen sentences. In the evaluation dataset, glosses in the six testing sentences have dispersedly appeared in the 94 sentences under different semantics context, occurrence order and application scenarios. Meanwhile, distribution of gloss samples in videos is imbalanced. Even so, the proposed HRF still excels at recognizing more meaningful words than others.

C. Evaluation of the Trajectory HRF-S

Here, we discuss tests conducted to verify the proposed HRF-S with skeleton data. We discuss the trajectory HRF-S parameters used to obtain its best performance. In the training dataset, the maximum number of skeleton features of a video is up to 948, and the minimum number is 89. In addition, greater than 80% of the videos contain no more than 300 frames. We implemented the key clip mining function Ω_{rss} on the training samples to obtain the number of summarized signemes n'. Here, the range of n' was [64, 293]. Note that there were very few samples with higher n', such as only one sample with $n' = 293$. To save calculation memory for sequential learning, we set $n' = 89$ (minimum video length) as the longest encoding length for both $LSTM_2$ and $LSTM_3$. If samples under $n > n'$, we performed systematic sampling on these samples to compress them to n'. We also set the hidden state number of LSTM to $n_{hid} = 200$ to avoid out of memory problems.

TABLE VIII
COMPARISON OF DIFFERENT MODEL FRAMEWORKS WITH SKELETON DATA

	Precision on Split I	WER on Split II
LSTM&CTC	/	/
S2VT ($\bar{n} = n' = 293$) [54]	0.840	0.811
S2VT ($\bar{n} = n' = 89$) [54]	0.940	0.772
S2VT (3-layer, $\bar{n} = n' = 89$)	0.946	0.815
HRNE [14]	0.917	0.787
HRNN-Ske [60]	0.907	1.020
HRF-S (RSS+Key pooling)	0.941	0.803
HRF-S (RSS+Linear pooling)	0.940	0.779
HRF-S (RSS+SYS+Linear pooling)	**0.947**	**0.734**

1) Comparison of Seen Sentence: The precision comparison of Split I shown in Fig. 9(a) indicates that both linear pooling and key pooling are suitable for skeleton-based SLT. As shown, the linear pooling demonstrates the best performance. Taking the linear pooling as the base, Fig. 9(b) compares different HRF network settings. Due to the continuity of the curve of the skeleton trajectory, the idea of linear interpolation is particularly suitable relative to modeling the trajectory curve. We adjusted a little strategy in the proposed ACS scheme, *i.e.*, we used the key clip mining function Ω_{rss} to obtain the adaptive n', and adopted systematic sampling (SYS) to segment each video into n' to-be-pooled segments. Thus, the "RSS + Linear pooling" setting was transformed to "RSS + SYS + Linear pooling." The experimental results demonstrate that this improves performance. Trajectory HRF-S resembles the interpolation idea, which is suitable to model the curve of skeleton trajectory. For the skeleton data, HRF-S with RSS mining, SYS sampling, and linear pooling is the default setting in the following experiments.

Note that Table VIII does not show the results of LSTM&CTC model due to non-convergent training. In the training dataset, each sentence contains only four to eight (average: five) decoded glosses, but has 89-293 to-be-encoded skeleton features. As a result of the difference between

TABLE IX
EVALUATION OF SEEN SENTENCE TRANSLATION OF SKELETON-BASED SLT USING SPLIT I

	Precision	CIDEr	BLEU-4	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR
S2VT ($\bar{n} = n' = 293$) [54]	0.840	8.226	0.841	0.848	0.859	0.881	0.885	0.612
S2VT ($\bar{n} = n' = 89$) [54]	0.940	9.178	0.941	0.944	0.947	0.953	0.953	0.729
S2VT (3-layer, $\bar{n} = n' = 89$)	0.946	9.232	0.950	0.950	0.952	0.958	0.956	**0.740**
HRNE [14]	0.917	9.036	0.932	0.934	0.937	0.944	0.942	0.711
HRNN-Ske [60]	0.907	8.868	0.911	0.914	0.919	0.930	0.930	0.684
HRF-S	**0.947**	**9.251**	**0.951**	**0.953**	**0.955**	**0.960**	**0.959**	0.738
HRF-S-att	**0.948**	**9.258**	**0.947**	**0.949**	**0.952**	**0.958**	**0.960**	0.738

TABLE X
EVALUATION OF UNSEEN SENTENCE TRANSLATION OF SKELETON-BASED SLT USING SPLIT II

	Acc-w	CIDEr	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR	*WER*
S2VT ($\bar{n} = n' = 293$) [54]	0.279	0.130	0	0.096	0.299	0.286	0.090	0.811
S2VT ($\bar{n} = n' = 89$) [54]	0.300	0.136	0	0.120	0.323	0.312	0.096	0.772
S2VT (3-layer, $\bar{n} = n' = 89$)	0.270	0.160	0.025	0.111	0.289	0.285	0.097	0.815
HRNE [14]	0.264	0.169	0	0.117	0.297	0.298	0.093	0.787
HRNN-Ske [60]	0.128	0.032	**0.098**	**0.306**	0.299	0.091	**0.279**	1.020
HRF-S	**0.313**	**0.176**	0.027	0.127	**0.346**	**0.343**	0.109	**0.734**
HRF-S-att	0.296	**0.168**	0	0.120	**0.325**	**0.315**	0.103	**0.743**

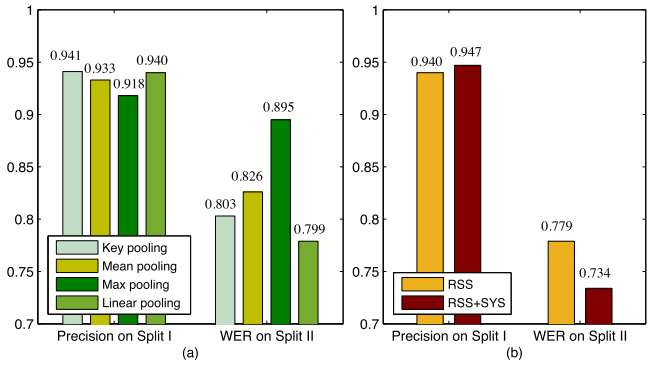


Fig. 9. Comparison with skeleton data, where higher precision and lower WER is better. (a) Different pooling strategies with RSS. (b) Key clip selection strategies RSS vs. RSS+SYS with linear pooling.

encoding and decoding time steps, CTC based on idea of temporal alignment classification can not converge in this situation. However, the encoder-decoder frameworks (*i.e.*, S2VT, HRNN-Ske and the proposed HRF-S) still perform well. Among the compared approaches, HRNN-Ske is the only one that specifically designed for skeleton data. In the proposed method, we concatenate the coordinates of multiple skeletal joints as the skeleton feature at each time step. However, HRNN-Ske takes each skeletal joint coordinate separately as a feature, and then performs feature embedding fusion. The worse performance of HRNN-Ske indicates that sequential learning on each independent skeleton point is not an effective choice for SLT because a sign action requires the collaboration of multiple skeleton joints.

In addition, similar to the conclusion obtained for RGB-based SLT, S2VT (3-layer) outperformed S2VT because it incorporates additional recurrent learning via the top $LSTM_1$. However, HRNE, *i.e.*, a flexible compact network with fixed intervals after $LSTM_1$, does not outperform S2VT (3-layer) because, with skeleton data, only online systematic sampling in deep learning framework is not suitable. In contrast, the proposed HRF with both adaptive clip selection n' and linear pooling demonstrates the best performance. Table IX indicates the superiority of the proposed HRF. Both HRF-S and HRF-S-att perform better than the compared methods. Thus, we conclude that skeleton data are helpful relative to solving the seen SLT task, and trajectory HRF-S performs well on fitting and discriminating the curve of the body pose trajectory.

2) Challenge With Unseen Skeleton-Based SLT: Unfortunately, using skeleton data to recognize unseen sentences remains a significant challenge. The experimental results are listed according to WER in Fig. 9, Table IX, and Table X. Taking Table X as an example, the proposed HRF-S outperforms all compared methods. However, compared to the results obtained with RGB-based unseen SLT shown Table VII, the skeleton-based unseen SLT is much worse. We conclude that none of these models work well for practical applications due to rare and unbalanced data distribution. Compared to RGB image data, our skeleton data contain only four joint coordinates, which lack a comprehensive description of the visual shape and spatial layout of a gesture or posture in the images. In addition, other challenges of skeleton data exist. (1) If different glosses have the same skeleton coordinates but different gestures, *e.g.*, glosses "one" and "two," using skeleton data is not beneficial. (2) For most similar trajectories with different spatial layouts, it is difficult to discriminate such trajectories. For example, Chinese sign glosses "one hundred" and "who" are gestured by shaking a clenched fist; however, their coordinates for the first and frequencies of the shaking action differ.

D. Evaluation of HRF-Fusion

To explore the complementarity of RGB and skeleton data, we tested the fusion phase with the best settings. For example, S2VT-fusion fuses the RGB-based S2VT ($n' = 21$)

TABLE XI
FUSION COMPARISON ON SPLITS I AND II

Evaluation on Split I for seen sentence recognition								
	Precision	CIDEr	BLEU-4	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR
S2VT-Fusion	0.979	9.549	0.979	0.980	0.981	0.984	0.984	0.793
S2VT-Fusion (3-layer)	0.957	9.344	0.950	0.954	0.958	0.966	0.970	0.739
ELM-Early fusion [52]	0.826	8.101	0.832	0.840	0.851	0.874	0.874	0.559
ELM-Late fusion [52]	0.970	9.462	0.967	0.972	0.977	0.979	0.970	0.760
HRF-Fusion	**0.991**	**9.665**	**0.990**	**0.991**	**0.992**	**0.993**	**0.994**	**0.817**
Evaluation on Split II for unseen sentence recognition								
	Acc-w	CIDEr	BLEU-3	BLEU-2	BLEU-1	ROUGE-L	METEOR	WER
S2VT-Fusion	0.406	0.335	0.107	0.213	0.419	0.407	0.150	0.732
S2VT-Fusion (3-layer)	0.374	**0.504**	0.130	0.217	0.373	0.406	0.149	0.680
ELM-Early fusion [52]	0.367	0.240	0.073	0.161	0.348	0.352	0.116	0.968
ELM-Late fusion) [52]	0.175	0.028	**0.142**	0.381	0.376	0.120	0.388	0.987
HRF-Fusion	**0.445**	0.398	0.127	**0.238**	**0.450**	**0.449**	**0.171**	**0.672**

and skeleton-based S2VT ($n' = 89$) with compact encoding lengths. S2VT-fusion (3-layer) has the same settings, *i.e.*, $n' = 21$ and $n' = 89$. To obtain the best performance, HRF-Fusion combines RGB-based HRF-S-attn and skeleton-based HRF-S. As shown in Table XI, the sentence precision of the proposed HRF-Fusion on Split I was 99.0% under a signer-independence test. It should be noted that S2VT (3-layer) generally outperformed S2VT significantly on single modality data; however, its fusion performance was worse. In this case, the collocation of V_{rgb} and V_{ske} in S2VT (3-layer) did not perform well. This demonstrates that obtaining the proper V_{rgb} and V_{ske} for fusion is important. HRF-Fusion excels at this. In addition, the proposed fusion phase (Eq. 12) belongs to score fusion. Compared to ELM-Late fusion, which also performs adaptive score fusion, the experimental results demonstrate the superiority of our fusion process again.

For unseen SLT, with the exception of CIDEr and BLEU-3, HRF-Fusion achieves the best performance. A longer gloss phrase always devotes higher relevant semantic values to the metrics CIDEr and BLEU-3. It remains challenging to discover longer gloss phrases. In addition, these fusion models demonstrate worse performance compared to their single RGB-based models because the to-be-fused skeleton-based probability scores suffers from poor performance, which negatively impacts the fusion process. This negative effect indicates that unseen SLT fusion remains challenging, which will be the focus of future work.

VI. CONCLUSION

The paper has proposed a hierarchical adaptive recurrent network with variable-length key clip mining, temporal pooling, and attention-aware weighting mechanisms. The proposed network builds a high-level visual semantic fusion for SLT. The experimental results have demonstrated that this model achieves promising performances for RGB-based SLT, skeleton-based SLT, and RGB-skeleton-based SLT. However, SLT still faces many challenges, *e.g.*, imbalanced data distributions of obscure glosses, unsolved out-of-order gloss-level alignment, and unseen sentence SLT. In future, we plan to explore adaptive gloss-level alignment for unseen sentence translation.

REFERENCES

[1] P. R. Futane, R. V. Dharaskar, and V. M. Thakare, "A comparative study for approaches for hand sign language," in *Proc. IJCA Proc. Nat. Conf. Innov. Paradigms Eng. Technol.*, 2012, pp. 36–39.

[2] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1290–1297.

[3] X. Cai, W. Zhou, L. Wu, J. Luo, and H. Li, "Effective active skeleton representation for low latency human action recognition," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 141–154, Feb. 2016.

[4] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Comput. Vis. Image Understand.*, vol. 141, pp. 108–125, Dec. 2015.

[5] J. Pu, W. Zhou, and H. Li, "Dilated convolutional network with iterative optimization for continuous sign language recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 885–891.

[6] H. Cooper, B. Holt, and R. Bowden, "Sign language recognition," in *Visual Analysis of Humans: Looking at People*, T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, Eds. London, U.K.: Springer, 2011, pp. 539–562. doi: 10.1007/978-0-85729-997-0_27.

[7] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4507–4515.

[8] S. C. W. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 873–891, Jun. 2005.

[9] O. Koller, O. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid CNN-HMM for continuous sign language recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–12.

[10] 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.*, Jun. 2017, pp. 4297–4305.

[11] 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.*, Jul. 2017, pp. 7361–7369.

[12] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[13] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao, "A key volume mining deep framework for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1991–1999.

[14] 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.*, Jun. 2016, pp. 1029–1038.

[15] L. Wang *et al.*, "Temporal segment networks for action recognition in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[16] 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.*, Jun. 2015, pp. 4489–4497.

[17] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "ModDrop: Adaptive multi-modal gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1692–1706, Aug. 2016.

[18] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 872–885.
[19] Y. Lin, X. Chai, Y. Zhou, and X. Chen, "Curve matching from the view of manifold for sign language recognition," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 233–246.
[20] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 5832–5841.
[21] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3D action recognition," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2842–2855, Jun. 2018.
[22] 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 Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8349–8358.
[23] H. Wang, X. Chai, Y. Zhou, and X. Chen, "Fast sign language recognition benefited from low rank approximation," in *Proc. IEEE Conf. Workshops Autom. Face Gesture Recognit.*, May 2015, pp. 1–6.
[24] A. Hernández-Vela *et al.*, "BoVDW: Bag-of-visual-and-depth-words for gesture recognition," in *Proc. 21st Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 449–452.
[25] Y. C. Tewari, K. Koduru, V. Mishra, and P. K. Upadhyay, "American sign language recognition using HAAR type classifier," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 19, no. 1, pp. 63–68, 2015.
[26] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and Kinect devices," in *Proc. IEEE Conf. Image Process.*, Oct. 2014, pp. 1565–1569.
[27] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen, "Isolated sign language recognition with Grassmann covariance matrices," *ACM Trans. Accessible Comput.*, vol. 8, no. 4, 2016, Art. no. 14.
[28] F. Yin, X. Chai, Y. Zhou, and X. Chen, "Semantics constrained dictionary learning for signer-independent sign language recognition," in *Proc. IEEE Conf. Image Process.*, Sep. 2015, pp. 3310–3314.
[29] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.
[30] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici, "Gesture recognition using skeleton data with weighted dynamic time warping," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2013, pp. 620–625.
[31] R. Yang and S. Sarkar, "Gesture recognition using hidden Markov models from fragmented observations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 766–773.
[32] C. Keskin, A. Erkan, and L. Akarun, "Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM," in *Proc. Int. Conf. Neural Inf. Process.*, 2003, pp. 26–29.
[33] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1521–1527.
[34] T. Ishihara and N. Otsu, "Gesture recognition using auto-regressive coefficients of higher-order local auto-correlation features," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2004, pp. 583–588.
[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[37] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3D convolutional neural networks," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Jul. 2015, pp. 1–6.
[38] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Using convolutional 3D neural networks for user-independent continuous gesture recognition," in *Proc. Int. Conf. Pattern Recognit.*, Dec. 2016, pp. 49–54.
[39] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *Proc. IEEE Conf. Image Process.*, Sep. 2016, pp. 2871–2875.
[40] N. Neverova, C. Wolf, G. Paci, G. Sommavilla, G. W. Taylor, and F. Nebout, "A multi-scale approach to gesture detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 484–491.
[41] 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 network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4207–4215.
[42] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia, "Inertial gesture recognition with BLSTM-RNN," in *Artificial Neural Networks*, P. Koprinkova-Hristova, V. Mladenov, and N. K. Kasabov, Eds. Cham, Switzerland: Springer, 2015, pp. 393–410.
[43] X. Chai, H. Wang, F. Yin, and X. Chen, "Communication tool for the hard of hearings: A large vocabulary sign language recognition system," in *Proc. Int. Conf. Affect. Comput. Intell. Interact.*, Sep. 2015, pp. 781–783.
[44] 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, 2015.
[45] D. Wu *et al.*, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1583–1597, Aug. 2016.
[46] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, pp. 115–139, 2016.
[47] 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.*, 2014, pp. 474–490.
[48] 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*, 2018, pp. 1483–1491.
[49] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical LSTM for sign language translation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6845–6852.
[50] 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: https://arxiv.org/abs/1607.06416
[51] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2257–2264.
[52] X. Chen and M. Koskela, "Using appearance-based hand features for dynamic RGB-D gesture recognition," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 411–416.
[53] D. Guo, W. Zhou, H. Li, and M. Wang, "Online early-late fusion based on adaptive HMM for sign language recognition," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, 2017, Art. no. 8.
[54] 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.*, Jun. 2015, pp. 4534–4542.
[55] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1741–1750.
[56] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
[57] 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*, Jul. 2016, pp. 1–6.
[58] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4594–4602.
[59] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
[60] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3010–3022, Jul. 2016.
[61] 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.
[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Conf. Learn. Represent.*, 2015, pp. 1–14.

Dan Guo received the B.E. degree in computer science and technology from Yangtze University, China, in 2004, and the Ph.D. degree in system analysis and integration from the Huazhong University of Science and Technology, China, in 2010. She is currently an Associate Professor with the School of Computer Science and Information Engineering, Hefei University of Technology, China. Her research interests include computer vision, machine learning, and intelligent multimedia content analysis.

Wengang Zhou received the B.E. degree in electronic information engineering from Wuhan University, China, in 2006, and the Ph.D. degree in electronic engineering and information science from the University of Science and Technology of China (USTC), China, in 2011. From September 2011 to 2013, he was a Postdoctoral Researcher with the Computer Science Department, The University of Texas at San Antonio. He is currently a Professor with the EEIS Department, USTC. His research interests include multimedia information retrieval and computer vision.

Anyang Li received the B.E. degree in computer science and technology from Nanjing Normal University, China, in 2016, and the M.S. degree in computer technology from the Hefei University of Technology, China, in 2019. He is currently a Software Development Engineer with Huawei Cloud AI Platform. His research interests include computer vision, big data analysis, and distributed computing.

Houqiang Li (M'10–SM'12) received the B.S., M.Eng., and Ph.D. degrees in electronic engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1992, 1997, and 2000, respectively. He is currently a Professor with the Department of Electronic Engineering and Information Science, USTC. He has authored or coauthored over 100 articles in journals and conferences. His research interests include multimedia search, image/video analysis, and video coding and communication. He was a recipient of the Best Paper Award at the Visual Communications and Image Processing in 2012, the Best Paper Award at the International Conference on Internet Multimedia Computing and Service in 2012, and the Best Paper Award at the International Conference on Mobile and Ubiquitous Multimedia from ACM in 2011. He was a Senior Author of the Best Student Paper of the 5th International Mobile Multimedia Communications Conference in 2009. He has served on technical/program committees and organizing committees and as the program co-chair or the track/session chair for over ten international conferences. He has served an Associate Editor for the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2013. He has been serving on the Editorial Board for the *Journal of Multimedia* since 2009.

Meng Wang (SM'17) received the B.E. and Ph.D. degrees, in the special class for the gifted young, from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively. He is currently a Professor with the Hefei University of Technology, China. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored over 200 book chapters and journal and conference articles in these areas. He was a recipient of the ACM SIGMM Rising Star Award in 2014. He is also an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.