

# Continuous Sign Language Recognition with Correlation Network

Lianyu Hu, Liqing Gao, Zekang Liu, Wei Feng✉

College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

Code : <https://github.com/hulianyuuy/CorrNet>

## Abstract

Human body trajectories are a salient cue to identify actions in the video. Such body trajectories are mainly conveyed by hands and face across consecutive frames in sign language. However, current methods in continuous sign language recognition (CSLR) usually process frames independently, thus failing to capture cross-frame trajectories to effectively identify a sign. To handle this limitation, we propose correlation network (CorrNet) to explicitly capture and leverage body trajectories across frames to identify signs. In specific, a correlation module is first proposed to dynamically compute correlation maps between the current frame and adjacent frames to identify trajectories of all spatial patches. An identification module is then presented to dynamically emphasize the body trajectories within these correlation maps. As a result, the generated features are able to gain an overview of local temporal movements to identify a sign. Thanks to its special attention on body trajectories, CorrNet achieves new state-of-the-art accuracy on four large-scale datasets, i.e., PHOENIX14, PHOENIX14-T, CSL-Daily, and CSL. A comprehensive comparison with previous spatial-temporal reasoning methods verifies the effectiveness of CorrNet. Visualizations demonstrate the effects of CorrNet on emphasizing human body trajectories across adjacent frames.

## 1. Introduction

Sign language is one of the most widely-used communication tools for the deaf community in their daily life. However, mastering this language is rather difficult and time-consuming for the hearing people, thus hindering direct communications between two groups. To relieve this problem, isolated sign language recognition tries to classify a video segment into an independent gloss<sup>1</sup>. Continuous sign language recognition (CSLR) progresses by sequentially translating images into a series of glosses to express a sentence, more prospective toward real-life deployment.

<sup>1</sup>Gloss is the atomic lexical unit to annotate sign languages.



Figure 1. Visualization of correlation maps with Grad-CAM [39]. It's observed that without extra supervision, our method could well attend to informative regions in adjacent left/right frames to identify human body trajectories.

Human body trajectories are a salient cue to identify actions in human-centric video understanding [44]. In sign language, such trajectories are mainly conveyed by both manual components (hand/arm gestures), and non-manual components (facial expressions, head movements, and body postures) [10,35]. Especially, both hands move horizontally and vertically across consecutive frames quickly, with finger twisting and facial expressions to express a sign. To track and leverage such body trajectories is of great importance to understanding sign language.

However, current CSLR methods [5, 6, 16, 33, 34, 36, 54] usually process each frame separately, thus failing to exploit such critical cues in the early stage. Especially, they usually adopt a shared 2D CNN to capture spatial features for each frame independently. In this sense, frames are processed individually without interactions with adjacent neighbors, thus inhibited to identify and leverage cross-frame trajectories to express a sign. The generated features are thus not aware of local temporal patterns and fail to perceive the hand/face movements in expressing a sign. To handle this limitation, well-known 3D convolution [4] or its (2+1)D variants [42, 49] are potential candidates to capture short-term temporal information to identify body trajectories. Other temporal methods like temporal shift [30] or temporal convolutions [31] can also attend to short-term temporal movements. However, it's hard for them to aggregate beneficial information from distant informative spatial regions due to their limited spatial-temporal receptive field.

Besides, as their structures are fixed for each sample during inference, they may fail to dynamically deal with different samples to identify informative regions. To tackle these problems, we propose to explicitly compute correlation maps between adjacent frames to capture body trajectories, referred to as CorrNet. As shown in fig. 1, our approach dynamically attends to informative regions in adjacent left/right frames to capture body trajectories, without relying on extra supervision.

In specific, our CorrNet first employs a correlation module to compute correlation maps between the current frame and its adjacent frames to identify trajectories of all spatial patches. An identification module is then presented to dynamically identify and emphasize the body trajectories embodied within these correlation maps. This procedure doesn't rely on extra expensive supervision like body keypoints [53] or heatmaps [54], which could be end-to-end trained in a lightweight way. The resulting features are thus able to gain an overview of local temporal movements to identify a sign. Remarkably, CorrNet achieves new state-of-the-art accuracy on four large-scale datasets, i.e., PHOENIX14 [26], PHOENIX14-T [2], CSL-Daily [52], and CSL [23], thanks to its special attention on body trajectories. A comprehensive comparison with other spatial-temporal reasoning methods demonstrates the superiority of our method. Visualizations hopefully verify the effects of CorrNet on emphasizing human body trajectories across adjacent frames.

## 2. Related Work

### 2.1. Continuous Sign Language Recognition

Sign language recognition methods can be roughly categorized into isolated sign language recognition [18, 19, 43] and continuous sign language recognition [5, 6, 33, 34, 37] (CSLR), and we focus on the latter in this paper. CSLR tries to translate image frames into corresponding glosses in a weakly-supervised way: only sentence-level label is provided. Earlier methods [12, 13] in CSLR always employ hand-crafted features or HMM-based systems [15, 26–28] to perform temporal modeling and translate sentences step by step. HMM-based systems first employ a feature extractor to capture visual features and then adopt an HMM to perform long-term temporal modeling.

The recent success of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) brings huge progress for CSLR. The widely used CTC loss [14] in recent CSLR methods [5, 6, 33, 34, 36, 37] enables training deep networks in an end-to-end manner by sequentially aligning target sentences with input frames. These CTC-based methods first rely on a feature extractor, i.e., 3D or 2D&1D CNN hybrids, to extract frame-wise features, and then adopt a LSTM for capturing long-term temporal de-

pendencies. However, several methods [6, 37] found in such conditions the feature extractor is not well-trained and then present an iterative training strategy to relieve this problem, but consume much more computations. Some recent studies [5, 16, 33] try to directly enhance the feature extractor by adding alignment losses [16, 33] or adopt pseudo labels [5] in a lightweight way, alleviating the heavy computational burden. More recent works enhance CSLR by squeezing more representative temporal features [21] or dynamically emphasizing informative spatial regions [22].

Our method is designed to explicitly incorporate body trajectories to identify a sign, especially those from hands and face. Some previous methods have also explicitly leveraged the hand and face features for better recognition. For example, CNN-LSTM-HMM [25] employs a multi-stream HMM (including hands and face) to integrate multiple visual inputs to improve recognition accuracy. STMC [53] first utilizes a pose-estimation network to estimate human body keypoints and then sends cropped appearance regions (including hands and face) for information integration. More recently, C<sup>2</sup>SLR [54] leverages the pre-extracted pose keypoints as supervision to guide the model to explicitly focus on hand and face regions. Our method doesn't rely on additional cues like pre-extracted body keypoints [54] or multiple streams [25], which consume much more computations to leverage hand and face information. Instead, our model could be end-to-end trained to dynamically attend to body trajectories in a self-motivated way.

### 2.2. Applications of Correlation Operation

Correlation operation has been widely used in various domains, especially optical flow estimation and video action recognition. Rocco et al. [38] used it to estimate the geometric transformation between two images, and Feichtenhofer et al. [11] applied it to capture object co-occurrences across time in tracking. For optical flow estimation, Deep matching [47] computes the correlation maps between image patches to find their dense correspondences. CNN-based methods like FlowNet [9] and PWC-Net [40] design a correlation layer to help perform multiplicative patch comparisons between two feature maps. For video action recognition, Zhao et al. [51] firstly employ a correlation layer to compute a cost volume to estimate the motion information. STCNet [8] considers spatial correlations and temporal correlations, respectively, inspired by SENet [20]. MFNet [29] explicitly estimates the approximation of optical flow based on fixed motion filters. Wang et al. [44] design a learnable correlation filter and replace 3D convolutions with the proposed filter to capture spatial-temporal information. Different from these methods that explicitly or implicitly estimate optical flow, the correlation operator in our method is used in combination with other operations to identify and track body trajectories across frames.

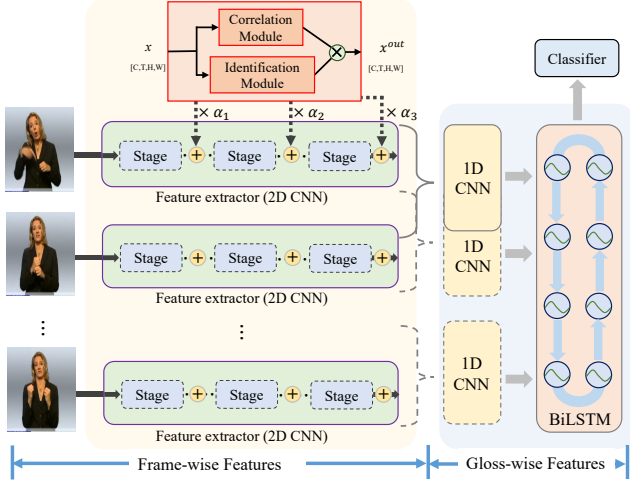


Figure 2. An overview for our CorrNet. It first employs a feature extractor (2D CNN) to capture frame-wise features, and then adopts a 1D CNN and a BiLSTM to perform short-term and long-term temporal modeling, respectively, followed by a classifier to predict sentences. We place our proposed identification module and correlation module after each stage of the feature extractor to identify body trajectories across adjacent frames.

### 3. Method

#### 3.1. Overview

As shown in fig. 2, the backbone of CSLR models consists of a feature extractor (2D CNN<sup>2</sup>), a 1D CNN, a BiLSTM, and a classifier (a fully connected layer) to perform prediction. Given a sign language video with  $T$  input frames  $x = \{x_t\}_{t=1}^T \in \mathcal{R}^{T \times 3 \times H_0 \times W_0}$ , a CSLR model aims to translate the input video into a series of glosses  $y = \{y_i\}_{i=1}^N$  to express a sentence, with  $N$  denoting the length of the label sequence. Specifically, the feature extractor first processes input frames into frame-wise features  $v = \{v_t\}_{t=1}^T \in \mathcal{R}^{T \times d}$ . Then the 1D CNN and BiLSTM perform short-term and long-term temporal modeling based on these extracted visual representations, respectively. Finally, the classifier employs widely-used CTC loss [14] to predict the probability of target gloss sequence  $p(y|x)$ .

The CSLR model processes input frames independently, failing to incorporate interactions between consecutive frames. We present a correlation module and an identification module to identify body trajectories across adjacent frames. Fig. 2 shows an example of a common feature extractor consisting of multiple stages. The proposed two modules are placed after each stage, whose outputs are element-wisely multiplied and added into the original features via a learnable coefficient  $\alpha$ .  $\alpha$  controls the contribu-

<sup>2</sup>Here we only consider the feature extractor based on 2D CNN, because recent findings [1, 54] show 3D CNN can not provide as precise gloss boundaries as 2D CNN, and lead to lower accuracy.

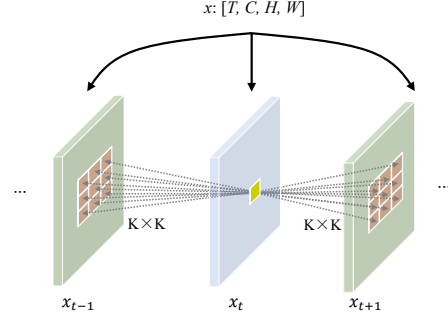


Figure 3. Illustration for the correlation operator. It computes affinities between a feature patch  $p(i, j)$  in  $x_t$  and patches  $p_{t+1}(i', j')/p_{t-1}(i', j')$  in adjacent frame  $x_{t+1}/x_{t-1}$ .

tions of the proposed modules, and is initialized as zero to make the whole model keep its original behaviors. The correlation module computes correlation maps between consecutive frames to capture trajectories of all spatial patches. The identification module dynamically locates and emphasizes body trajectories embedded within these correlation maps. The outputs of correlation and identification modules are multiplied to enhance inter-frame correlations.

#### 3.2. Correlation Module

Sign language is mainly conveyed by both manual components (hand/arm gestures), and non-manual components (facial expressions, head movements, and body postures) [10, 35]. However, these informative body parts, e.g., hands or face, are misaligned in adjacent frames. We propose to compute correlation maps between adjacent frames to identify body trajectories.

Each frame could be represented as a 3D tensor  $C \times H \times W$ , where  $C$  is the number of channels and  $H \times W$  denotes spatial size. Given a feature patch  $p_t(i, j)$  in current frame  $x_t$ , we compute the affinity between patch  $p(i, j)$  and another patch  $p_{t+1}(i', j')$  in adjacent frame  $x_{t+1}$ , where  $(i, j)$  is the spatial location of the patch. To restrict the computation, the size of the feature patch could be reduced to a minimum, i.e., a pixel. The affinity between  $p(i, j)$  and  $p_{t+1}(i', j')$  is computed in a dot-product way as:

$$A(i, j, i', j') = \frac{1}{C} \sum_{c=1}^C (p_t^c(i, j) \cdot p_{t+1}^c(i', j')). \quad (1)$$

For the spatial location  $(i, j)$  in  $x_t$ ,  $(i', j')$  is often restricted within a  $K \times K$  neighborhood in  $x_{t+1}$  to relieve spatial misalignment. A visualization is given in fig. 3. Thus, for all pixels in  $x_t$ , the correlation maps are a tensor of size  $H \times W \times K \times K$ .  $K$  could be set as a smaller value to keep semantic consistency or as a bigger value to attend to distant informative regions.

Given the correlation map between a pixel and its neighbors in adjacent frame  $x_{t+1}$ , we constrain its range

into (0,1) to measure their semantic similarity by passing  $A(i, j, i', j')$  through a sigmoid function. We further subtract 0.5 from the results, to emphasize informative regions with positive values, and suppress redundant areas with negative values as:

$$A'(i, j, i', j') = \text{Sigmoid}(A(i, j, i', j')) - 0.5 \quad (2)$$

After identifying the trajectories between adjacent frames, we incorporate these local temporal movements into the current frame  $x_t$ . Specifically, for a pixel in  $x_t$ , its trajectories are aggregated from its  $K \times K$  neighbors in adjacent frame  $x_{t+1}$ , by multiplying their features with the corresponding affinities as :

$$T(i, j) = \sum_{i', j'} A'(i, j, i', j') * x_{t+1}(i', j'). \quad (3)$$

In this sense, each pixel is able to be aware of its trajectories across consecutive frames. We aggregate bidirectional trajectories from both  $x_{t-1}$  and  $x_{t+1}$ , and attach a learnable coefficient  $\beta$  to measure the importance of bi-directions. Thus, eq. 3 could be updated as :

$$T(i, j) = \beta_1 \cdot \sum_{i', j'} A'_{t+1}(i, j, i', j') * x_{t+1}(i', j') + \beta_2 \cdot \sum_{i', j'} A'_{t-1}(i, j, i', j') * x_{t-1}(i', j') \quad (4)$$

where  $\beta_1$  and  $\beta_2$  are initialized as 0.5. This correlation calculation is repeated for each frame in a video to track body trajectories in videos.

### 3.3. Identification Module

The correlation module computes correlation maps between each pixel with its  $K \times K$  neighbors in adjacent frames  $x_{t-1}$  and  $x_{t+1}$ . However, as not all regions are critical for expressing a sign, only those informative regions carrying body trajectories should be emphasized in the current frame  $x_t$ . The trajectories of background or noise should be suppressed. We present an identification module to dynamically emphasize these informative spatial regions. Specifically, as informative regions like hand and face are misaligned in adjacent frames, the identification module leverages the closely correlated local spatial-temporal features to tackle the misalignment issue and locate informative regions.

As shown in fig. 4, the identification module first projects input features  $x \in \mathcal{R}^{T \times C \times H \times W}$  into  $x_r \in \mathcal{R}^{T \times C/r \times H \times W}$  with a  $1 \times 1 \times 1$  convolution to decrease the computations, by a channel reduction factor  $r$  as 16 by default.

As the informative regions, e.g., hands and face, are not exactly aligned in adjacent frames, it's necessary to consider a large spatial-temporal neighborhood to identify

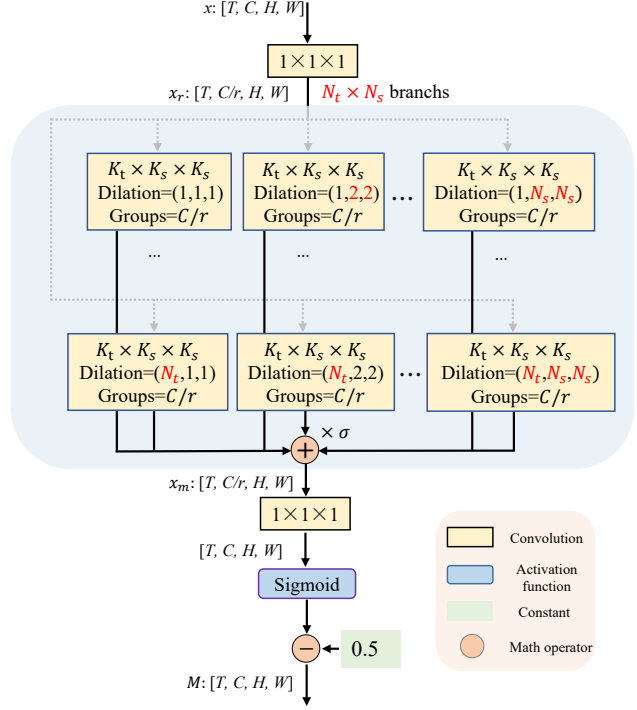


Figure 4. Illustration for our identification module.

these features. Instead of directly employing a large 3D spatial-temporal kernel, we present a multi-scale paradigm by decomposing it into parallel branches of progressive dilation rates to reduce required computations and increase the model capacity.

Specifically, as shown in fig. 4, with a same small base convolution kernel of  $K_t \times K_s \times K_s$ , we employ multiple convolutions with their dilation rates increasing along spatial and temporal dimensions concurrently. The spatial and temporal dilation rate range within  $(1, N_s)$  and  $(1, N_t)$ , respectively, resulting in total  $N_s \times N_t$  branches. Group convolutions are employed for each branch to reduce parameters and computations. Features from different branches are multiplied with learnable coefficients  $\{\sigma_1, \dots, \sigma_{N_s \times N_t}\}$  to control their importance, and then added to mix information from branches of various spatial-temporal receptive fields as:

$$x_m = \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \sigma_{i,j} \cdot \text{Conv}_{i,j}(x_r) \quad (5)$$

where the group-wise convolution  $\text{Conv}_{i,j}$  of different branches receives features of different spatial-temporal neighborhoods, with dilation rate  $(j, i, i)$ .

After receiving features from a large spatial-temporal neighborhood,  $x_m$  is sent into a  $1 \times 1 \times 1$  convolution to project its channels back into  $C$ . It then passes through a sigmoid function to generate attention maps  $M \in \mathcal{R}^{T \times C \times H \times W}$  with its values ranging within (0,1). Spe-

cially,  $M$  is further subtracted from a constant value of 0.5 to emphasize informative regions with positive values, and suppress redundant areas with negative values as:

$$M = \text{Sigmoid}(\text{Conv}_{1 \times 1 \times 1}(x_m)) - 0.5. \quad (6)$$

Given the attention maps  $M$  to identify informative regions, it's multiplied with the aggregated trajectories  $T(x)$  by the correlation module to emphasize body trajectories and suppress others like background or noise. This refined trajectory information is finally incorporated into original spatial features  $x$  via a residual connection as:

$$x^{out} = x + \alpha T(x) \cdot M. \quad (7)$$

As stated before,  $\alpha$  is initialized as zero to keep the original spatial features.

## 4. Experiments

### 4.1. Experimental Setup

#### 4.1.1 Datasets.

**PHOENIX14** [26] is recorded from a German weather forecast broadcast with nine actors before a clean background with a resolution of  $210 \times 260$ . It contains 6841 sentences with a vocabulary of 1295 signs, divided into 5672 training samples, 540 development (Dev) samples and 629 testing (Test) samples.

**PHOENIX14-T** [2] is available for both CSLR and sign language translation tasks. It contains 8247 sentences with a vocabulary of 1085 signs, split into 7096 training instances, 519 development (Dev) instances and 642 testing (Test) instances.

**CSL-Daily** [52] revolves the daily life, recorded indoor at 30fps by 10 signers. It contains 20654 sentences, divided into 18401 training samples, 1077 development (Dev) samples and 1176 testing (Test) samples.

**CSL** [23] is collected in the laboratory environment by fifty signers with a vocabulary size of 178 with 100 sentences. It contains 25000 videos, divided into training and testing sets by a ratio of 8:2.

#### 4.1.2 Training details.

For fair comparisons, we follow the same setting as state-of-the-art methods [33, 54] to prepare our model. We adopt ResNet18 [17] as the 2D CNN backbone with ImageNet [7] pretrained weights. The 1D CNN of state-of-the-art methods is set as a sequence of {K5, P2, K5, P2} layers where  $K\sigma$  and  $P\sigma$  denotes a 1D convolutional layer and a pooling layer with kernel size of  $\sigma$ , respectively. A two-layer BiLSTM with hidden size 1024 is attached for long-term temporal modeling, followed by a fully connected layer for sentence prediction. We train our models for 40 epochs with

Configurations	Dev(%)	Test(%)
-	20.2	21.0
$N_t=4, N_s=1$	19.6	20.1
$N_t=4, N_s=2$	19.2	19.8
$N_t=4, N_s=3$	<b>18.8</b>	<b>19.4</b>
$N_t=4, N_s=4$	19.1	19.7
$N_t=2, N_s=3$	19.4	19.9
$N_t=3, N_s=3$	19.1	19.7
$N_t=4, N_s=3$	<b>18.8</b>	<b>19.4</b>
$N_t=5, N_s=3$	19.3	19.8
$K_t=9, K_s=7$	19.9	20.4

Table 1. Ablations for the multi-scale architecture of identification module on the PHOENIX14 dataset.

initial learning rate 0.001 which is divided by 5 at epoch 20 and 30. Adam [24] optimizer is adopted as default with weight decay 0.001 and batch size 2. All input frames are first resized to  $256 \times 256$ , and then randomly cropped to  $224 \times 224$  with 50% horizontal flipping and 20% temporal rescaling during training. During inference, a  $224 \times 224$  center crop is simply adopted. Following VAC [33], we employ the VE loss and VA loss for visual supervision, with weights 1.0 and 25.0, respectively. Our model is trained and evaluated upon a 3090 graphical card.

#### 4.1.3 Evaluation Metric.

We use Word Error Rate (WER) as the evaluation metric, which is defined as the minimal summation of the **substitution**, **insertion**, and **deletion** operations to convert the predicted sentence to the reference sentence, as:

$$\text{WER} = \frac{\#\text{sub} + \#\text{ins} + \#\text{del}}{\#\text{reference}}. \quad (8)$$

Note that the **lower** WER, the **better** accuracy.

## 4.2. Ablation Study

We report ablative results on both development (Dev) and testing (Test) sets of PHOENIX14 dataset.

**Study on the multi-scale architecture of identification module.** In tab. 1, without identification module, our baseline achieves 20.2% and 21.0% WER on the Dev and Test Set, respectively. The base kernel size is set as  $3 \times 3 \times 3$  for  $K_t \times K_s \times K_s$ . When fixing  $N_t=4$  and varying spatial dilation rates to expand spatial receptive fields, it's observed a larger  $N_s$  consistently brings better accuracy. When  $N_s$  reaches 3, it brings no more accuracy gain. We set  $N_s$  as 3 by default and test the effects of  $N_t$ . One can see that either increasing  $K_t$  to 5 or decreasing  $K_t$  to 2 and 3 achieves worse accuracy. We thus adopt  $N_t$  as 4 by default. We also compare our proposed multi-scale architecture with a normal implementation of more parameters. The receptive field

Configurations	Dev(%)	Test(%)
-	20.2	21.0
$K=3$	19.6	20.4
$K=5$	19.4	20.2
$K=7$	19.2	20.0
$K=9$	19.1	19.8
$K=H$ or $W$ (Full image)	<b>18.8</b>	<b>19.4</b>

Table 2. Ablations for the articulated area of correlation module on the PHOENIX14 dataset.

Correlation	Identification	Dev(%)	Test(%)
✗	✗	20.2	21.0
✓	✗	19.5	20.0
✗	✓	19.4	19.9
✓	✓	<b>18.8</b>	<b>19.4</b>

Table 3. Ablations for the effectiveness of correlation module and identification module on the PHOENIX14 dataset.

of the identification module with  $N_t=4$ ,  $N_s=3$  is identical to a normal convolution with  $K_t=9$  and  $K_s=7$ . As shown in the bottom of tab. 1, although a normal convolution owns more parameters and computations than our proposed architecture, it still performs worse, verifying the effectiveness of our architecture.

#### Study on the neighborhood $K$ of correlation module.

In tab. 2, when  $K$  is null, the correlation module is disabled. It’s observed that a larger  $K$ , i.e., more incorporated spatial-temporal neighbors, consistently brings better accuracy. The performance reaches the peak when  $K$  equals  $H$  or  $W$ , i.e., the full image is incorporated. In this case, distant informative objects could be interacted to provide discriminative information. We set  $K=H$  or  $W$  by default.

**Effectiveness of two proposed modules.** In tab. 3, we first notice that either only using the correlation module or identification module could already bring a notable accuracy boost, with 19.5% & 20.0% and 19.4% & 19.9% accuracy on the Dev and Test Sets, respectively. When combining both modules, the effectiveness is further activated with 18.8% & 19.4% accuracy on the Dev and Test Sets, respectively, which is adopted as the default setting.

**Effects of locations for CorrNet.** Tab 4 ablates the locations of our proposed modules, which are placed after Stage 2, 3 or 4. It’s observed that choosing any one of these locations could bring a notable accuracy boost, with 19.6% & 20.1%, 19.5% & 20.2% and 19.4% & 20.0% accuracy boost. When combining two or more locations, a larger accuracy gain is witnessed. The accuracy reaches the peak when proposed modules are placed after Stage 2, 3 and 4, with 18.8% & 19.4% accuracy, which is adopted by default.

**Generalizability of CorrNet.** We deploy CorrNet upon

Stage 2	Stage 3	Stage 4	Dev(%)	Test(%)
✗	✗	✗	20.2	21.0
✓	✗	✗	19.6	20.1
✗	✓	✗	19.5	20.2
✗	✗	✓	19.4	20.0
✓	✓	✗	19.2	19.9
✓	✓	✓	<b>18.8</b>	<b>19.4</b>

Table 4. Ablations for the locations of CorrNet on the PHOENIX14 dataset.

Configurations	Dev(%)	Test(%)
SqueezeNet [20]	22.2	22.6
w/ CorrNet	<b>20.2</b>	<b>20.4</b>
ShuffleNet V2 [32]	21.7	22.2
w/ CorrNet	<b>19.7</b>	<b>20.2</b>
GoogLeNet [41]	21.4	21.5
w/ CorrNet	<b>19.6</b>	<b>19.8</b>

Table 5. Ablations for the generalizability of CorrNet over multiple backbones on the PHOENIX14 dataset.

Methods	Dev(%)	Test(%)
-	20.2	21.0
w/ SENet [20]	19.8	20.4
w/ CBAM [48]	19.7	20.2
w/ NLNet [46]	-	-
I3D [4]	22.6	22.9
R(2+1)D [42]	22.4	22.3
TSM [30]	19.9	20.5
CorrNet	<b>18.8</b>	<b>19.4</b>

Table 6. Comparison with other methods of spatial-temporal attention or temporal reasoning on the PHOENIX14 dataset.

multiple backbones, including SqueezeNet [20], ShuffleNet V2 [32] and GoogLeNet [41] to validate its generalizability in tab. 5. The proposed modules are placed after three spatial downsampling layers in SqueezeNet, ShuffleNet V2 and GoogLeNet, respectively. It’s observed that our proposed model generalizes well upon different backbones, bringing +2.0% & +2.2%, +2.0% & +2.0% and +1.8% & +1.7% accuracy boost on the Dev and Test Sets, respectively.

**Comparisons with other spatial-temporal reasoning methods.** Tab. 6 compares our approach with other methods of spatial-temporal reasoning ability. SENet [20] and CBAM [48] perform channel attention to emphasize key information. NLNet [46] employs non-local means to aggregate spatial-temporal information from other frames. I3D [4] and R(2+1)D [42] deploys 3D or 2D+1D convolutions to capture spatial-temporal features. TSM [30] adopts temporal shift operation to obtain features from ad-

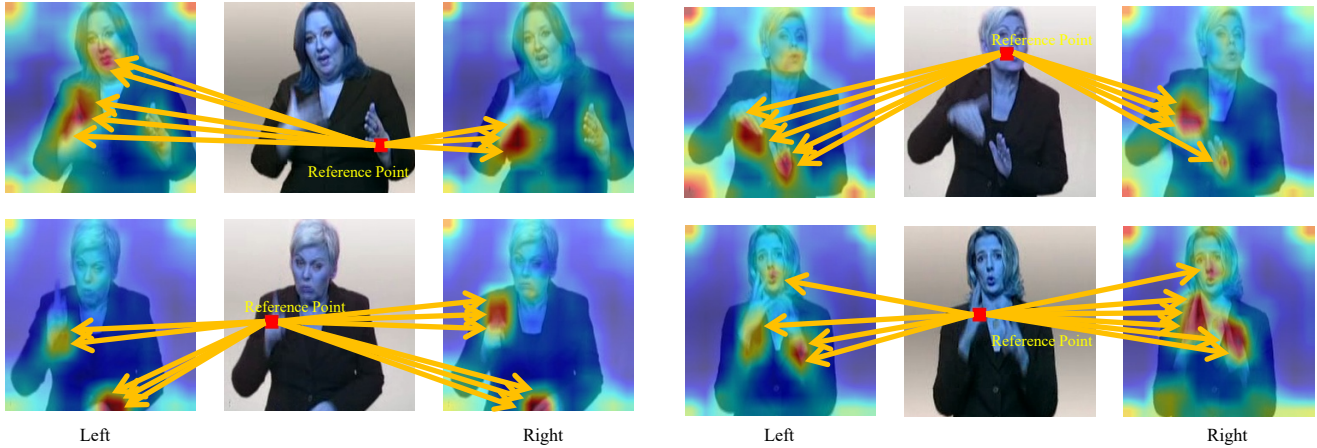


Figure 5. Visualizations of correlation maps for correlation module. Based on correlation operators, each frame could especially attend to informative regions in adjacent left/right frames like hands and face (dark red areas).

Methods	Dev(%)	Test(%)
CNN+HMM+LSTM [25]	26.0	26.0
DNF [6]	23.1	22.9
STMC [53]	21.1	20.7
C <sup>2</sup> SLR [54]	20.5	20.4
CorrNet	<b>18.8</b>	<b>19.4</b>

Table 7. Comparison with other methods that explicitly exploit hand and face features on the PHOENIX14 dataset.

adjacent frames. In the upper part of tab. 6, one can see CorrNet largely outperforms other attention-based methods, i.e., SENet, CBAM and NLNet, for its superior ability to identify and aggregate body trajectories. NLNet is out of memory due to its quadratic computational complexity with spatial-temporal size. In the bottom part of tab. 6, it’s observed that I3D and R(2+1)D even degrade accuracy, which may be attributed to their limited spatial-temporal receptive fields and increased training complexity. TSM slightly brings 0.3% & 0.3% accuracy boost. Our proposed approach surpasses these methods greatly, verifying its effectiveness in aggregating beneficial spatial-temporal information, from even distant spatial neighbors.

**Comparisons with previous methods equipped with hand or face features.** Many previous CSLR methods explicitly leverage hand and face features for better recognition, like multiple input streams [25], human body keypoints [53, 54] and pre-extracted hand patches [6]. They require extra expensive pose-estimation networks like HR-Net [45] or additional training stages. Our approach doesn’t rely on extra supervision and could be end-to-end trained to dynamically attend to body trajectories like hand and face in a self-motivated way. Tab. 7 shows that our method outperforms these methods by a large margin.

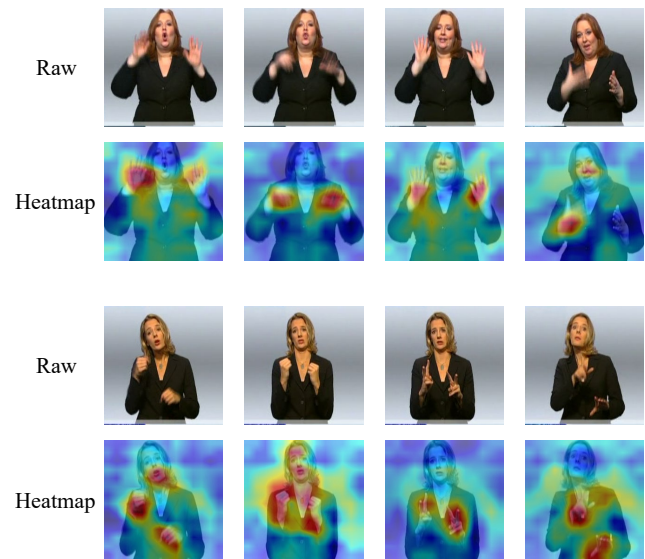


Figure 6. Visualizations of heatmaps by Grad-CAM [39]. Top: raw frames; Bottom: heatmaps of our identification module. Our identification module could generally focus on the human body (light yellow areas) and especially pays attention to informative regions like hands and face (dark red areas) to track body trajectories.

### 4.3. Visualizations

**Visualizations for correlation module.** Fig. 5 shows the correlation maps generated by our correlation module with adjacent frames. It’s observed that the reference point could well attend to informative regions in adjacent left/right frame, e.g., hands or face, to track body trajectories in expressing a sign. Especially, they always focus on the moving body parts that play a major role in expressing signs. For example, the reference point (left hand) in the up-

Methods	Backbone	PHOENIX14				PHOENIX14-T	
		Dev(%)		Test(%)		Dev(%)	Test(%)
		del/ins	WER	del/ins	WER		
SFL [34]	ResNet18	7.9/6.5	26.2	7.5/6.3	26.8	25.1	26.1
FCN [5]	Custom	-	23.7	-	23.9	23.3	25.1
CMA [36]	GoogLeNet	7.3/2.7	21.3	7.3/2.4	21.9	-	-
VAC [33]	ResNet18	7.9/2.5	21.2	8.4/2.6	22.3	-	-
SMKD [16]	ResNet18	6.8/2.5	20.8	6.3/2.3	21.0	20.8	22.4
TLP [21]	ResNet18	6.3/2.8	19.7	6.1/2.9	20.8	19.4	21.2
SEN [22]	ResNet18	5.8/2.6	19.5	7.3/4.0	21.0	19.3	20.7
SLT* [2]	GoogLeNet	-	-	-	-	24.5	24.6
CNN+LSTM+HMM* [25]	GoogLeNet	-	26.0	-	26.0	22.1	24.1
DNF* [6]	GoogLeNet	7.3/3.3	23.1	6.7/3.3	22.9	-	-
STMC* [53]	VGG11	7.7/3.4	21.1	7.4/2.6	20.7	19.6	21.0
C <sup>2</sup> SLR* [54]	ResNet18	-	20.5	-	20.4	20.2	20.4
<b>CorrNet</b>	ResNet18	5.6/2.8	<b>18.8</b>	5.7/2.3	<b>19.4</b>	<b>18.9</b>	<b>20.5</b>

Table 8. Comparison with state-of-the-art methods on the PHOENIX14 and PHOENIX14-T datasets. \* indicates extra clues such as face or hand features are included by additional networks or pre-extracted heatmaps.

per left figure specially attends to the quickly moving right hand to capture sign information.

**Visualizations for identification module.** Fig. 6 shows the heatmaps generated by our identification module. Our identification module could generally focus on the human body (light yellow areas). Especially, it pays major attention to regions like hands and face (dark red areas). These results show that our identification module could dynamically emphasize important areas in expressing a sign, e.g., hands and face, and suppress other regions.

#### 4.4. Comparison with State-of-the-Art Methods

**PHOENIX14 and PHOENIX14-T.** Tab. 8 shows a comprehensive comparison between our CorrNet and other state-of-the-art methods. The entries notated with \* indicate these methods utilize additional factors like face or hand features for better accuracy. We notice that CorrNet outperforms other state-of-the-art methods by a large margin upon both datasets, thanks to its special attention on body trajectories. Especially, CorrNet outperforms previous CSLR methods equipped with hand and faces acquired by heavy pose-estimation networks or pre-extracted heatmaps (notated with \*), without additional expensive supervision.

**CSL-Daily.** CSL-Daily is a recently released large-scale dataset with the largest vocabulary size (2k) among commonly-used CSLR datasets, with a wide content covering family life, social contact and so on. Tab. 9 shows that our CorrNet achieves new state-of-the-art accuracy upon this challenging dataset with notable progress, which generalizes well upon real-world scenarios.

**CSL.** As shown in tab. 10, our CorrNet could achieve extremely superior accuracy (0.8% WER) upon this well-examined dataset, outperforming existing CSLR methods.

Methods	Dev(%)	Test(%)
LS-HAN [23]	39.0	39.4
TIN-Iterative [6]	32.8	32.4
Joint-SLRT [3]	33.1	32.0
FCN [5]	33.2	32.5
BN-TIN [52]	33.6	33.1
<b>CorrNet</b>	<b>30.6</b>	<b>30.1</b>

Table 9. Comparison with other methods on the CSL-Daily dataset [52].

Methods	WER(%)
SF-Net [50]	3.8
FCN [5]	3.0
STMC [53]	2.1
VAC [33]	1.6
C <sup>2</sup> SLR [54]	0.9
<b>CorrNet</b>	<b>0.8</b>

Table 10. Comparison with other methods on the CSL dataset [23].

## 5. Conclusion

This paper introduces a correlation module to capture trajectories between adjacent frames and an identification module to locate body regions. Comparisons with previous CSLR methods with spatial-temporal reasoning or hand and face features demonstrate the superiority of CorrNet. Visualizations show that CorrNet could generally attend to hand and face regions to capture body trajectories.

### Acknowledgments.

The work is supported by the National Natural Science Foundation of China (Project No. 62072334).



## References

- [1] Nikolas Adaloglou, Theodoris Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, Vassia Zacharopoulou, George J Xydopoulos, Klimnis Atzakas, Dimitris Papazachariou, and Petros Daras. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, 24:1750–1762, 2021. [3](#)
- [2] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018. [2](#), [5](#), [8](#)
- [3] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033, 2020. [8](#)
- [4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. [1](#), [6](#)
- [5] Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai. Fully convolutional networks for continuous sign language recognition. In *ECCV*, 2020. [1](#), [2](#), [8](#)
- [6] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *TMM*, 21(7):1880–1891, 2019. [1](#), [2](#), [7](#), [8](#)
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [5](#)
- [8] Ali Diba, Mohsen Fayyaz, Vivek Sharma, M Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. Spatio-temporal channel correlation networks for action classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018. [2](#)
- [9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. [2](#)
- [10] Philippe Dreuw, David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. Speech recognition techniques for a sign language recognition system. *hand*, 60:80, 2007. [1](#), [3](#)
- [11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE international conference on computer vision*, pages 3038–3046, 2017. [2](#)
- [12] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition*, volume 12, pages 296–301. Zurich, Switzerland, 1995. [2](#)
- [13] Wen Gao, Gaolin Fang, Debin Zhao, and Yiqiang Chen. A chinese sign language recognition system based on sofm/srn/hmm. *Pattern Recognition*, 37(12):2389–2402, 2004. [2](#)
- [14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. [2](#), [3](#)
- [15] Junwei Han, George Awad, and Alistair Sutherland. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623–633, 2009. [2](#)
- [16] Aiming Hao, Yuecong Min, and Xilin Chen. Self-mutual distillation learning for continuous sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11303–11312, 2021. [1](#), [2](#), [8](#)
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [5](#)
- [18] Hezhen Hu, Weichao Zhao, Wengang Zhou, Yuechen Wang, and Houqiang Li. Signbert: Pre-training of hand-model-aware representation for sign language recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11087–11096, 2021. [2](#)
- [19] Hezhen Hu, Wengang Zhou, and Houqiang Li. Hand-model-aware sign language recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1558–1566, 2021. [2](#)
- [20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. [2](#), [6](#)
- [21] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. Temporal lift pooling for continuous sign language recognition. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 511–527. Springer, 2022. [2](#), [8](#)
- [22] Lianyu Hu, Liqing Gao, Zekang Liu, and Wei Feng. Self-emphasizing network for continuous sign language recognition. In *Thirty-seventh AAAI conference on artificial intelligence*, 2023. [2](#), [8](#)
- [23] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. [2](#), [5](#), [8](#)
- [24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#)
- [25] Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. Weakly supervised learning with multi-stream cnn-1stm-hmms to discover sequential parallelism in sign language videos. *PAMI*, 42(9):2306–2320, 2019. [2](#), [7](#), [8](#)
- [26] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015. [2](#), [5](#)
- [27] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Hybrid cnn-hmm for continuous sign lan-

- guage recognition. In *Proceedings of the British Machine Vision Conference 2016*, 2016. 2
- [28] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *CVPR*, 2017. 2
- [29] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403, 2018. 2
- [30] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019. 1, 6
- [31] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11669–11676, 2020. 1
- [32] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 6
- [33] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. Visual alignment constraint for continuous sign language recognition. In *ICCV*, 2021. 1, 2, 5, 8
- [34] Zhe Niu and Brian Mak. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *ECCV*, 2020. 1, 2, 8
- [35] Sylvie CW Ong and Surendra Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(06):873–891, 2005. 1, 3
- [36] Junfu Pu, Wengang Zhou, Hezhen Hu, and Houqiang Li. Boosting continuous sign language recognition via cross modality augmentation. In *ACM MM*, 2020. 1, 2, 8
- [37] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *CVPR*, 2019. 2
- [38] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017. 2
- [39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 7
- [40] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2
- [41] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 6
- [42] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1, 6
- [43] Anirudh Tunga, Sai Vidyaranya Nuthalapati, and Juan Wachs. Pose-based sign language recognition using gcn and bert. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 31–40, 2021. 2
- [44] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 352–361, 2020. 1, 2
- [45] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 7
- [46] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 6
- [47] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1385–1392, 2013. 2
- [48] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 6
- [49] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 1
- [50] Zhaoyang Yang, Zhenmei Shi, Xiaoyong Shen, and Yu-Wing Tai. Sf-net: Structured feature network for continuous sign language recognition. *arXiv preprint arXiv:1908.01341*, 2019. 8
- [51] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6566–6575, 2018. 2
- [52] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1325, 2021. 2, 5, 8
- [53] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*, 2020. 2, 7, 8
- [54] Ronglai Zuo and Brian Mak. C2slr: Consistency-enhanced continuous sign language recognition. In *Proceedings of*

*the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5131–5140, 2022. [1](#), [2](#), [3](#), [5](#), [7](#), [8](#)