C^{2}SLR: Consistency-enhanced Continuous Sign Language Recognition

Ronglai Zuo and Brian Mak
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
{rzuo,mak}@cse.ust.hk

Abstract

The backbone of most deep-learning-based continuous sign language recognition (CSLR) models consists of a visual module, a sequential module, and an alignment module. However, such CSLR backbones are hard to be trained sufficiently with a single connectionist temporal classification loss. In this work, we propose two auxiliary constraints to enhance the CSLR backbones from the perspective of consistency. The first constraint aims to enhance the visual module, which easily suffers from the insufficient training problem. Specifically, since sign languages convey information mainly with signers' faces and hands, we insert a keypoint-guided spatial attention module into the visual module to enforce it to focus on informative regions, i.e., spatial attention consistency. Nevertheless, only enhancing the visual module may not fully exploit the power of the backbone. Motivated by that both the output features of the visual and sequential modules represent the same sentence, we further impose a sentence embedding consistency constraint between them to enhance the representation power of both the features. Experimental results over three representative backbones validate the effectiveness of the two constraints. More remarkably, with a transformer-based backbone, our model achieves state-of-the-art or competitive performance on three benchmarks, PHOENIX-2014, PHOENIX-2014-T, and CSL.

1. Introduction

Hearing-impaired people usually use sign languages as their communication method. Video-based continuous sign language recognition (CSLR) aims to transcribe a sign language video into a sequence of glosses (basic lexical units in a sign language). In recent years, deep learning techniques dominate CSLR modeling because of their superiority over traditional methods [31, 32, 55]. According to [32], the backbone of most deep-learning-based CSLR models consists of three components: a visual module, a sequential (contextual) module, and an alignment module. Within this framework, the visual module first extracts visual features from input videos. Then the sequential module extracts sequential and contextual information from the visual features. Finally, the alignment module aligns the sequential features with the gloss label sequence and computes its probability.

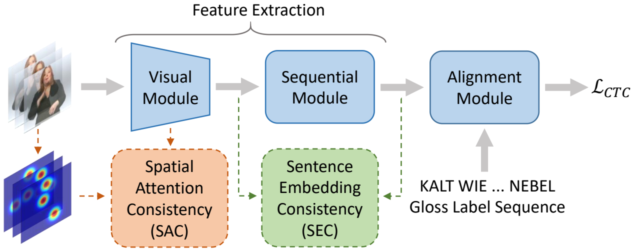

Figure 1. An overview of the CSLR backbone and the proposed consistency constraints. First, our SAC constraint leverages pose keypoints heatmaps to enforce the visual module to focus on informative regions. Second, our SEC constraint aligns the visual and sequential features at the sentence level, which can enhance the representation power of both the features simultaneously.

As a common practice, the connectionist temporal classification (CTC) [14] loss is adopted as the main objective function to train such CSLR backbones. However, only using the CTC loss may lead to the insufficient training problem that the extracted features are not representative enough to be used to yield accurate recognition results [10, 11, 17, 31, 37, 39, 55]. Two kinds of methods can relieve this issue. First, [11, 17, 37–39, 55] use a stage optimization strategy to iteratively refine the extracted features, which is time-consuming since the model needs to adapt to a different objective in a new stage [10]. As an alternative solution, auxiliary learning can keep the whole model end-to-end trainable by just adding one or more auxiliary tasks [10, 31]. In this work, we propose two novel auxiliary constraints from the perspective of information consistency.

Our first constraint aims to enhance the visual module, which plays a key role in feature extraction but easily suffers from the insufficient training problem [11, 31, 55]. Since the information of sign languages is mainly included in signers' faces and hands [23, 55], to enrich the visual features, some CSLR models [36, 55, 56] leverage an off-the-shelf pose detector [7, 43] to locate the face and hands and then crop the feature maps to form a multi-stream architecture. However, the multi-stream architecture will introduce many more parameters and the cropping operation cannot fully exploit the rich information contained in the pose keypoints heatmaps. As shown in Figure 1, we find that the heatmaps can reflect the importance of different spatial positions, which is quite similar to the attention mechanism. Thus, as shown in Figure 2, we insert a lightweight spatial attention module guided by keypoints heatmaps into the visual module to enforce it to focus on informative regions, which leads to our spatial attention consistency (SAC) constraint.

Only enhancing the visual module may not fully exploit the power of the backbone. Some works [17, 31] show that explicitly enforcing the consistency between the visual and sequential modules can strengthen their cooperation, and give better performance. VAC [31] treats the visual and sequential modules as the student and teacher, respectively, and achieves knowledge distillation between them. Similarly, SMKD [17] achieves knowledge transfer by sharing classifiers. Knowledge distillation can be seen as a kind of consistency since the KL-divergence loss used in [31] is a measurement of the distance between two probability distributions. However, the above two methods share the same deficiency that the consistency is measured at the frame level, *i.e.*, the probability distribution is computed for each frame independently. We think that there should be differences between the distributions of the visual and sequential modules at the frame level since the sequential module gathers contextual information for each frame; otherwise, the sequential module may be removed. Motivated by that both the visual and sequential features represent the same sentence, we impose a sentence embedding consistency (SEC) constraint between them. As shown in Figure 2, we build a sentence embedding extractor that can be co-trained with the CSLR backbone, and then minimize the distance between the sentence embeddings of visual and sequential features but maximize the distance between the sentence embeddings of visual and negative sequential features.

In summary, our main contributions are:

- We propose a spatial attention consistency constraint, which can enhance the visual module by leveraging pose keypoints heatmaps to guide an inner spatial attention module.
- We propose a sentence embedding consistency constraint, which can align the visual and sequential features at the sentence level and enhance the representation power of both the features simultaneously.
- Extensive experiments are conducted to validate that both consistency constraints can enhance the performance of three representative CSLR backbones with negligible extra cost. More remarkably, with a transformer-based backbone, our consistency-enhanced CSLR (C^{2}SLR) model can achieve state-of-the-art (SOTA) or competitive performance on three benchmarks, while the whole model is trained in an end-to-end manner.

2. Related Works

2.1. Deep-learning-based CSLR

According to [32], most deep-learning-based CSLR backbones can be separated into a visual module (2D-CNNs [17, 31, 55] or 3D-CNNs [39, 54]), a sequential module (RNNs [17, 31, 37, 39, 55], 1D-CNNs [10, 15], or Transformer [5, 32]), and an alignment module (hidden Markov models [24, 26] or CTC [17, 31, 55]). For sufficient training, [11] proposes a stage optimization strategy that uses pseudo labels to iteratively refine the extracted features, which is widely adopted in [17, 37, 39, 55]. On top of it, CMA [37] proposes a cross-modality constraint to help the training. SMKD [17] proposes a three-stage optimization approach, which is so time-consuming that the model needs to be trained for 100 epochs. Recently, VAC [31] proposes two auxiliary constraints over the frame-level probability distributions to enhance the visual module and to enforce the consistency between the visual and sequential modules, which enables the whole model end-to-end trainable. In this work, we enhance the visual module from a novel view, *i.e.*, spatial attention consistency, and align the two modules at the sentence level, *i.e.*, sentence embedding consistency.

2.2. Spatial Attention

Spatial attention has been proven to be effective on many computer vision tasks, including image classification [6, 19, 28, 46, 49], semantics segmentation [12], and object detection [6, 49]. However, training the spatial attention module with a task-specific loss function only may lead to sub-optimal solutions. Some works propose to leverage external information to guide the spatial attention module. [9] proposes to guide the spatial attention module with motion information for video captioning. Mask guidance and relation guidance are proposed in [27, 35] for occluded pedestrian detection and person re-identification, respectively. GALA [28] leverages click maps collected in a game to supervise the spatial attention module for image classification. In this work, we leverage pose keypoints heatmaps to guide the spatial attention module to enforce the visual module to focus on informative regions.

2.3. Sentence Embedding

A common practice to extract sentence embedding is feeding the word sequence into an LSTM or a bidirectional

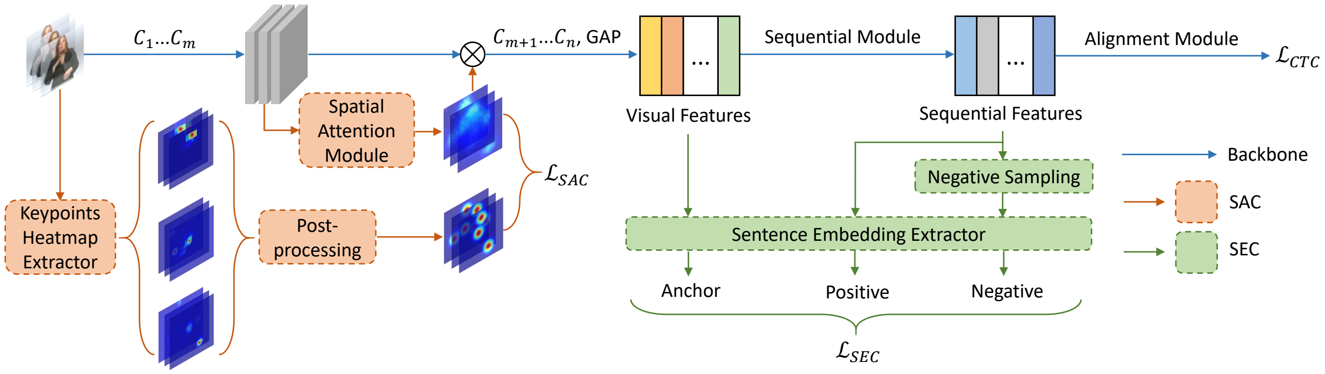

Figure 2. An overview of our C^{2}SLR. For spatial attention consistency (SAC), we first insert a spatial attention module after the m-th convolution layer, C_m, of the visual module, and then guide it by pre-extracted pose keypoints heatmaps. For sentence embedding consistency (SEC), we extract the sentence embeddings of visual features, sequential features, and negative sequential features, respectively, and adopt a triplet loss to train the sentence embedding extractor along with the CSLR backbone. (GAP: global average pooling.)

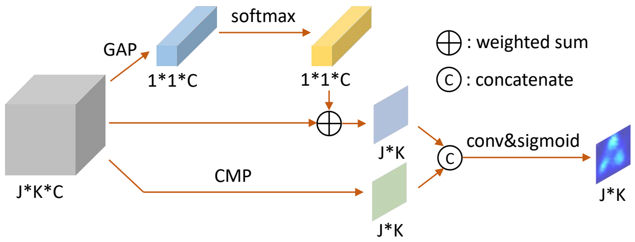

Figure 3. The architecture of our spatial attention module. ($J \times K \times C$: the size of the input feature maps, GAP: global average pooling, CMP: channel-wise max pooling.)

LSTM (BiLSTM) and taking the last one or two hidden state(s) as the sentence embedding [29, 34]. Recently, some powerful sentence embedding extractors [8, 13, 40] are built based on the BERT architecture [21]. However, it is difficult for these methods to fit our work because (1) they are too large to be co-trained along with the backbone; (2) they are pretrained on spoken languages, which are quite different to sign languages. In this work, we build a lightweight sentence embedding extractor that can be easily co-trained with the CSLR backbone.

3. Our Proposed Method

3.1. Framework Overview

As shown in Figure 2, the backbone of CSLR models is composed of a visual module, a sequential module, and an alignment module. Given a sign language video with T RGB frames $\mathbf{x} = \{\mathbf{x}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times 3\times H\times W}$, the visual module, which is composed of a stack of 2D-CNN[1] layers $(C_1, \ldots, C_n)$ and a global average pooling (GAP) layer, first extracts visual features $\mathbf{v} = \{\mathbf{v}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times d}$. After that, the sequential module further extracts sequential features $\mathbf{s} = \{\mathbf{s}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times d}$. Finally, the alignment module utilizes CTC [14] to compute the probability of the gloss label sequence $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y} = \{y_i\}_{i=1}^{N}$ and N is the length of the label sequence.

3.2. Spatial Attention Consistency (SAC)

Sign languages convey information mainly by signers' faces and hands [23, 55]. Thus, we hope the visual module can focus on these informative regions (IRs). Motivated by this idea, we insert a spatial attention module guided by pre-extracted pose keypoints heatmaps into the visual module. Since SAC is applied to all frames in the same way, we will omit the time steps in the formulation below.

Spatial Attention Module. To build our spatial attention module, we borrow the idea of CBAM [49] due to its simplicity. As shown in Figure 3, we first apply a channel-wise max pooling (CMP) operation to pick the most informative channel:

$$\mathbf{M}_1 = f_{CMP}(\mathbf{F}) \in \mathbb{R}^{J\times K\times 1}, \tag{1}$$

where $\mathbf{M}_1$ is the squeezed feature map by CMP, and $\mathbf{F} \in \mathbb{R}^{J\times K\times C}$ is the input feature maps.

Different from the spatial attention module in CBAM which uses an average pooling operation along the channel dimension, we propose to dynamically assign a weight to each channel to measure its importance. As shown in Figure 3, we first conduct global average pooling (GAP) over the input feature maps $\mathbf{F}$ to gather global spatial information. Then the channel weights $\mathbf{E} \in (0,1)^{1\times 1\times C}$ are simply generated by a channel-wise softmax layer. After that, we can generate another squeezed feature map $\mathbf{M}_2$ by a weighted sum operation along the channel dimension:

$$\mathbf{M}_2 = \mathbf{F} \oplus \mathbf{E} = \sum_{i=1}^{C} \mathbf{F}_i \cdot \mathbf{E}_i \in \mathbb{R}^{J\times K\times 1}, \tag{2}$$

Finally, the spatial attention mask $\mathbf{M}$ is generated as:

$$\mathbf{M} = \sigma(f_{conv}(cat(\mathbf{M}_1, \mathbf{M}_2))) \in (0,1)^{J\times K}, \tag{3}$$

[1] We only consider visual modules which are based on 2D-CNNs since a recent survey [1] shows that 3D-CNNs cannot provide as precise gloss boundaries as 2D-CNNs, and lead to lower accuracy.

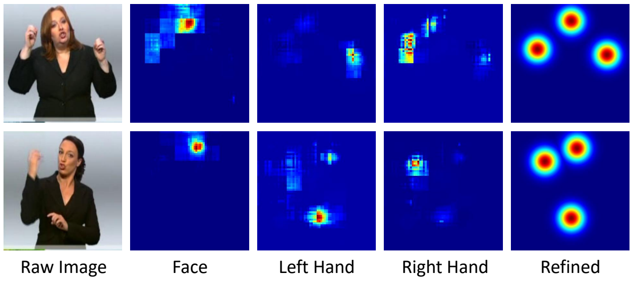

Figure 4. Two examples of the original and refined heatmaps.

where $\sigma(\cdot)$ is the sigmoid function, $f_{conv}(\cdot)$ is a 2D-CNN layer with a kernel size of 7×7, and $cat(\cdot, \cdot)$ is a channel-wise concatenation operation. The mask will be multiplied with the feature maps **F** to highlight important positions but suppress trivial ones.

It should be noted that the operation of assigning a weight to each channel shares a similar idea with the channel attention module in CBAM. But ours introduces no extra parameters and can even outperform the vanilla CBAM according to our ablation studies.

Keypoints Heatmap Extractor. The spatial attention module can be trained along with the backbone. However, since we have the prior knowledge that signers' faces and hands are more informative than other regions, we leverage an off-the-shelf pose extractor, HRNet [43] pretrained on MPII [2], to extract keypoints heatmaps to guide the spatial attention module. Specifically, we first normalize the raw outputs of HRNet to get the original heatmaps as:

$$\mathbf{H}_o^i = \frac{f_H^i(\mathbf{I}) - \min f_H^i(\mathbf{I})}{\max f_H^i(\mathbf{I}) - \min f_H^i(\mathbf{I})} \in [0,1]^{H\times W}, \quad (4)$$

where $\mathbf{I}$ is the raw RGB frame, $f_H(\cdot)$ is the HRNet, and $i \in \{1,2,3\}$ denotes the face, left hand, and right hand, respectively.

Post-processing. Although the original heatmaps can roughly reflect the positions of IRs, there are still some defects. First, the original heatmaps are not quite accurate. As shown in Figure 4, some unwanted regions may even get high activation values, *e.g.*, the top boundary of the face in the first row and the middle part of the left hand in the second row. Second, some bright regions may not be large enough to cover the whole IRs, *e.g.*, both of the faces in Figure 4. Third, the heatmap resolution of the pretrained HRNet is fixed to 64 × 64, which usually mismatches the resolution of the spatial attention mask. Thus, we propose a post-processing module to refine the original heatmaps.

Given the original heatmaps, we first generate the center for each IR via a simple argmax operation: $(x_i, y_i) = \operatorname{argmax} \mathbf{H}_o^i$. After that, to adapt to various resolutions of spatial attention masks, we further normalize the center as $(\hat{x}_i, \hat{y}_i) = (\frac{x_i}{H-1}, \frac{y_i}{W-1})$.

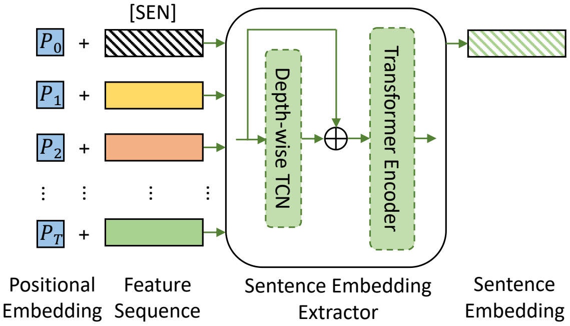

Figure 5. The workflow of sentence embedding extraction. We omit LayerNorm [3] for simplicity.

Suppose the spatial attention module yields a spatial attention mask with a resolution of $J \times K$. Then a Gaussian-like refined keypoints heatmap is generated for each IR:

$$\mathbf{H}_r^i(a,b) = \exp\left(-\frac{1}{2}\left(\frac{(a-\hat{c}_i^x)^2}{(J/\gamma_x)^2} + \frac{(b-\hat{c}_i^y)^2}{(K/\gamma_y)^2}\right)\right), \quad (5)$$

where $0 \le a < J$, $0 \le b < K$. $(\hat{c}_i^x, \hat{c}_i^y) = (\hat{x}_i(J-1), \hat{y}_i(K-1))$, which denotes the transformed center under the resolution $J \times K$ for each IR, respectively. γ_x and γ_y are two scalars to control the scale of the highlighted regions. Finally, we merge the three refined IR heatmaps into a single one: $\mathbf{H}_r = \max_i \mathbf{H}_r^i \in (0,1)^{J\times K}$.

SAC Loss. We leverage the refined keypoints heatmaps to guide the spatial attention module via the SAC loss[2]:

$$\mathcal{L}_{SAC} = \frac{1}{J\times K}\|\mathbf{M} - \mathbf{H}_r\|_2^2. \quad (6)$$

3.3. Sentence Embedding Consistency (SEC)

Enforcing the consistency between the visual and sequential features can enhance their representation power [17, 31]. Motivated by that both the features are representing the same sentence, we impose a sentence embedding consistency between them.

Sentence Embedding Extractor (SEE). Within a sign language video, each gloss only consists of a few frames, which implies the importance of local contexts. Motivated by this, we build our SEE based on QANet [53], which consists of a depth-wise temporal convolution network (TCN) layer and a transformer encoder layer as shown in Figure 5. The depth-wise TCN aims to first extract local contextual information from the frame-level feature sequence, then the transformer encoder models global contexts by the self-attention mechanism.

Similar to the [CLS] token in [21], we first prepend a learnable sentence embedding token, [SEN], to the sequential feature $\mathbf{s} \in \mathbb{R}^{T\times d}$:

$$\mathbf{s}' = cat([\text{SEN}], \mathbf{s}) \in \mathbb{R}^{(T+1)\times d}. \quad (7)$$

[2] For implementation, we further compute the average of $\mathcal{L}_{SAC}$ over all time steps.

The input of the SEE is the summation of the feature sequence and the positional embeddings [45]; *i.e.*, $\mathbf{s}'' = \mathbf{s}' + \mathbf{P}$, where $\mathbf{P} \in \mathbb{R}^{(T+1)\times d}$.

Within the SEE, a simple depth-wise TCN (1D depth-wise CNN) [50] layer first models local contexts with a residual connection: $\mathbf{s}''_l = f_{TCN}(\mathbf{s}'') + \mathbf{s}''$. Then a transformer encoder layer gathers information of all time steps via a set of attention weights to get the sentence embedding:

$$\mathbf{s}_{se} = f_{TF}(\mathbf{s}''_l) = \sum_{i=0}^{T} w_i \mathbf{s}''_{l_i} \in \mathbb{R}^d, \tag{8}$$

where the weights w_i are learned by the self-attention module in the transformer encoder. The sentence embedding of visual features, $\mathbf{v}_{se}$, can also be obtained in the same way.

Negative Sampling. Directly minimizing the distance between $\mathbf{s}_{se}$ and $\mathbf{v}_{se}$ may lead to trivial solutions. Suppose the parameters of SEE are all-zero, then whatever the input is, the output will be the same. To avoid this, negative samples are needed. In this work, we follow the common practice [18, 33, 41, 52] that sampling another video from the mini-batch and taking its sequential features as the negative sample. We also notice that most CSLR models [17, 31, 55, 56] are trained with a batch size of 2, thus our negative sampling strategy degenerates to swapping under this setting:

$$(\mathbf{B}^n[0], \mathbf{B}^n[1]) = (\mathbf{B}[1], \mathbf{B}[0]), \tag{9}$$

where $\mathbf{B} \in \mathbb{R}^{2\times T\times d}$ is a mini-batch of the sequential features, and $\mathbf{B}^n[\cdot]$ denotes the corresponding negative sample.

SEC Loss. To minimize the distance between the sentence embeddings of visual and sequential features of the same sentence and maximize those from different sentences, we implement SEC loss as a triplet loss [41]:

$$\mathcal{L}_{SEC} = \max\{d(\mathbf{v}_{se}, \mathbf{s}_{se}) - d(\mathbf{v}_{se}, \mathbf{s}^n_{se}) + \alpha, 0\}, \tag{10}$$

where $d(\cdot,\cdot) = 1 - cos(\cdot,\cdot)$; $\{\mathbf{v}_{se}, \mathbf{s}_{se}\}$ are sentence embeddings of visual and sequential features from the same sentence; $\{\mathbf{v}_{se}, \mathbf{s}^n_{se}\}$ are sentence embeddings of visual and sequential features from different sentences, and we call the sentence embedding of the sequential features from a different sentence as the negative sample $\mathbf{s}^n_{se}$; α is the margin.

3.4. Alignment Module and Loss Function

CTC [14] is widely adopted as the alignment module in recent works [17, 31, 37, 55]. It yields a label for each time step which may be a repeating label or a special blank symbol. With the assumption of conditional independence, given an input sequence $\mathbf{x}$, the conditional probability of a label sequence $\boldsymbol{\phi} = \{\phi_i\}_{i=1}^T$, where $\phi_i \in \mathcal{V} \cup \{blank\}$ and $\mathcal{V}$ is the gloss vocabulary, can be estimated by:

$$p(\boldsymbol{\phi}|\mathbf{x}) = \prod_{i=1}^{T} p(\phi_i|\mathbf{x}), \tag{11}$$

where $p(\phi_i|\mathbf{x})$ is the frame-level gloss probabilities generated by a classifier. Finally, the probability of yielding the true label sequence is a summation of all feasible alignments:

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\boldsymbol{\phi}=\mathcal{G}^{-1}(\mathbf{y})} p(\boldsymbol{\phi}|\mathbf{x}), \tag{12}$$

where $\mathcal{G}$ is a mapping function to remove repeats and blank symbols in $\boldsymbol{\phi}$. Then the CTC loss is defined as:

$$\mathcal{L}_{CTC} = -\log p(\mathbf{y}|\mathbf{x}). \tag{13}$$

Finally, the overall loss function of C^{2}SLR is a combination of the CTC, SAC, and SEC loss:

$$\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_{SAC} + \mathcal{L}_{SEC}. \tag{14}$$

4. Experiments

4.1. Datasets and Evaluation Metric

PHOENIX-2014 [25] is a German CSLR dataset with a vocabulary size of 1081. There are 5672, 540, and 629 samples in the training, development (dev), and test set, respectively. **PHOENIX-2014-T** [4] is an extension of PHOENIX-2014 with a vocabulary size of 1085. There are 7096, 519, and 642 samples in the training, dev, and test set, respectively. **CSL** [20, 39, 54] is a Chinese CSLR dataset consisting of 4000 and 1000 samples in the training and test set, respectively, with a vocabulary size of 178.

Evaluation Metric. We use word error rate (WER) to measure the dissimilarity between two sequences.

$$\text{WER} = \frac{\#\text{deletions} + \#\text{substitutions} + \#\text{insertions}}{\#\text{glosses in label}} \tag{15}$$

The evaluation scripts are provided by each dataset.

4.2. Implementation Details

Data Augmentation. RGB frames are resized to 256 × 256 before cropping to 224 × 224. Stochastic frame dropping [32] with a dropping ratio of 0.5 is used for both the PHOENIX datasets. Since videos in CSL are much longer, we adopt a *seg-and-drop* strategy that first split the videos into segments consisting of only two frames, and then one frame is randomly chosen from each segment. Finally, we randomly drop 40% frames from these processed videos.

Backbones and Hyper-parameters. Since there is no consensus on the architecture of CSLR backbones, we choose the following three representative backbones to validate the effectiveness of our method.

- VGG11+TCN+BiLSTM (VTB). It is the backbone adopted in the SOTA work [55].
- CNN+TCN (CT). This lightweight backbone only consists of a 9-layer 2D-CNN and a 3-layer TCN, which is adopted in [10].

- VGG11+Local Transformer (VLT). VGG11 [42] is adopted as the visual module, which is the same as the SOTA work [55]. The sequential module is a 2-layer local transformer encoder, which is similar to our sentence embedding extractor. The main difference is that we further leverage Gaussian bias [30, 51] to emphasize local contexts. More details are available in the supplementary materials.

To match the channel dimensions of the visual and sequential features, we set the number of the output channels of the TCN layers in CT and VTB to 512 and the number of hidden units of BiLSTM in VTB to 2×256, which leads to comparable WERs with those reported in the original papers [10,55]. We insert the spatial attention module after the 5th convolution layer, which is a trade-off between heatmap resolution and GPU memory limitation. In terms of post-processing, we set $\gamma_x = \gamma_y = 14$, which is a trade-off between covering informative regions and avoiding trivial regions. The kernel size of the depth-wise TCN layer in our SEE and VLT backbone is set to 5, which is the same as [53]. The margin α in Eq. 10 is set to 2, which is the maximum difference between the negative and positive distance with a cosine distance function.

Training. All models are trained with a batch size of 2 following recent works [17, 31, 55]. Model parameters are optimized by Adam optimizer [22] with an initial learning rate of 1×10^{-4} and a weight decay factor of 1×10^{-4}. We empirically find that $\mathcal{L}_{SEC}$ decreases much faster than $\mathcal{L}_{CTC}$, thus we multiply the learning rate of the SEE with a factor of 0.1/0.01/0.1 for the three backbones, respectively. We decrease the learning rate by a factor of 0.7 according to the performance on the dev set as [5]. But since CSL doesn't have an official dev split, we decrease the learning rate after the 15th and 25th epoch and per 5 epochs after the 30th epoch. Training will terminate if either it reaches the 60th/50th epoch for the PHOENIX/CSL datasets, respectively, or the learning rate is smaller than 1×10^{-5}.

Inference and Decoding. Following [32], to match the training condition, we evenly select every $\frac{1}{p_d}$-th frame to drop, where p_d is the dropping ratio. We adopt the beam search algorithm with a beam size of 10 for decoding.

4.3. Ablation Studies

We conduct ablation studies on PHOENIX-2014 following previous works [17, 31, 37, 55].

Effectiveness of SAC and SEC. As shown in Table 1, it is clear to see that both SAC and SEC can improve the performance of the three backbones which are only trained by the CTC loss. However, if we only insert the spatial attention module into the backbones, *i.e.*, SAC$^-$, the performance can only be improved slightly, which demonstrates the necessity of $\mathcal{L}_{SAC}$. Also, using SAC and SEC simultaneously can achieve better results than using either one of them. The superiority of SAC+SEC over SAC suggests that explicitly enforcing the consistency between the visual and sequential modules can strengthen the cooperation between the two modules, which can further improve the performance of the model. Besides, since VLT performs the best among the three backbones, we will use it as the default backbone for the following experiments.

Backbone	SAC$^-$	SAC	SEC	WER%	Par.(M)	Sp.(s)
VTB				25.0	15.6359	0.169
	✓			24.6	+0.0001	+0.002
		✓		23.7	+0.0001	+0.002
			✓	24.3	+0.0000	+0.000
		✓	✓	**22.6**	+0.0001	+0.002
CT				26.1	8.7504	0.095
	✓			26.0	+0.0001	+0.001
		✓		25.1	+0.0001	+0.001
			✓	25.2	+0.0000	+0.000
		✓	✓	**24.5**	+0.0001	+0.001
VLT				21.5	16.1850	0.163
	✓			21.4	+0.0001	+0.002
		✓		20.8	+0.0001	+0.002
			✓	20.9	+0.0000	+0.000
		✓	✓	**20.4**	+0.0001	+0.002

Table 1. Ablation study for SAC and SEC. During inference, since our SEC can be removed, only the spatial attention module in SAC will introduce negligible parameters and affect inference speed. (SAC$^-$ denotes only inserting the spatial attention module but not guided by $\mathcal{L}_{SAC}$, Par.: number of parameters, Sp.: inference speed measured on the same TITAN RTX GPU in seconds per video.)

Visualization Results for SAC. We visualize the learned spatial attention masks of SAC (with $\mathcal{L}_{SAC}$) and SAC$^-$ (without $\mathcal{L}_{SAC}$) of five test samples as shown in Figure 6. It should be noted that during testing, we don't use the keypoints heatmaps to guide the spatial attention module, thus our comparison is fair. Generally, it is clear that the attention masks with the constraint of $\mathcal{L}_{SAC}$ are much better. Without $\mathcal{L}_{SAC}$, the attention masks are quite noisy: horizontal lines on the top and many highlights at trivial regions, *e.g.*, the right arm of s_2 and the waist of s_3, s_5. This can also explain why SAC$^-$ cannot clearly improve the performance of the backbones as shown in Table 1. Our SAC is so robust that it can capture the IRs (face and hands) accurately even when the frames are blurry (right columns of s_1 to s_5). It is also capable to deal with different hand positions; *e.g.*, one hand is near the face (s_1, s_2, s_4), two hands are lower than the face (s_1, s_3), and hand overlapping (s_5).

Channel Weights. Within our spatial attention module, we dynamically assign a weight to each channel to measure its importance before squeezing the feature maps. As shown in Table 2, removing the channel weights, which degenerates to a simple channel-wise average pooling, can lead to a performance drop by 0.5%. Our channel weights are similar to the channel attention module of CBAM [49], but no

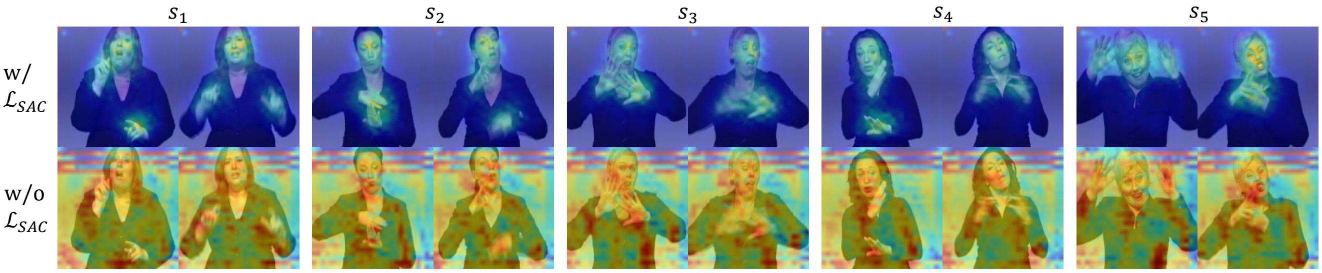

Figure 6. Visualization results for learned spatial attention masks with or without the guidance of $\mathcal{L}_{SAC}$. We randomly select five samples ($s_1, \ldots, s_5$) from the **test** set, and for each sample, we select one clear frame and one blurry frame. It is clear that the guidance of $\mathcal{L}_{SAC}$ can help the spatial attention module capture the IRs (face and hands) more accurately.

Method	WER%	#Param(M)
VLT + SAC	**20.8**	16.1851
- channel weights	21.3	-0.0000
+ channel attention [49]	21.2	+0.0335
- post-processing	21.7	-0.0000

Table 2. Ablation study for SAC.

Method	Extractor	Neg. Sam.	WER%
VLT + SEC	TF+DTCN	✓	**20.9**
	TF+DTCN	×	21.5
	TF	✓	21.1
	BiLSTM	✓	21.3

Table 3. Ablation study for the architecture of the sentence embedding extractor and negative sampling. (TF: Transformer, DTCN: depth-wise TCN, Neg. Sam.: negative sampling.)

Level	Constraint	WER%
Sentence	consistency	**20.9**
Frame	consistency	21.6
	visual enhancement (VE) [31]	22.3
	visual alignment (VA) [31]	21.9
	VE+VA [31]	22.8

Table 4. Ablation study for the constraint level. We fine-tuned the loss factor of VA as [31] on the VLT for fair comparisons.

extra parameters are introduced. To further validate their effectiveness, after removing the channel weights, we add the channel attention module as CBAM, however, it can only slightly improve the performance and cannot beat ours even with extra parameters.

Heatmap Refinement. As shown in Figure 4, original heatmaps may highlight unwanted regions and may not cover the IRs entirely. The results in Table 2 also imply the importance of the quality of the keypoints heatmaps, *i.e.*, only using original heatmaps without post-processing can drop the performance of SAC by nearly 1%.

Sentence Embedding Extractor and Negative Sampling. Within our sentence embedding extractor, a depth-wise TCN layer first models local contexts followed by a transformer encoder which gathers information of each frame. As shown in Table 3, dropping the TCN layer leads to worse performance, which suggests the importance of local contexts for sentence embedding extraction. We also compare our method with the common practice, *i.e.*, taking the concatenation of the last two hidden states of BiLSTM as the sentence embedding. However, it performs worse than transformer-based extractors, which validates the strength of the self-attention mechanism for sentence embedding extraction. Negative sampling plays a key role for our SEC. As shown in Table 3, removing negative sampling, *i.e.*, directly minimizing the distance between the sentence embeddings of the visual and sequential features, would render the SEC ineffective.

Constraint Level. We compare our SEC with some frame-level constraints to further validate its effectiveness as shown in Table 4. For fairness, we first replace the sentence embeddings $\mathbf{v}_{se}$ and $\mathbf{s}_{se}$ in Eq. 10 by the corresponding frame-level features, *i.e.*, minimizing the positive distance and maximizing the negative distance at the frame level. However, it leads to a WER of 21.6%, which is much worse than our SEC. We also compare our SEC with VAC [31], which consists of two frame-level constraints. VAC first appends a classifier to the visual module to yield a probability distribution for each frame (visual distribution). Then another CTC loss, which is the same as the one used for training the backbone, is computed between the visual distribution and the gloss label, *i.e.*, visual enhancement (VE). Second, a KL-divergence loss is computed to minimize the distance between the visual distribution and the original probability distribution ($p(\phi_i|\mathbf{x})$ in Eq. 11), *i.e.*, visual alignment (VA). As shown in Table 4, both VE and VA perform much worse than our SEC, which implies that the SEC is a more appropriate way to measure the consistency between the visual and sequential modules.

Examples of Sentence Embedding Distances. As shown in Figure 7, we provide two examples of video-gloss

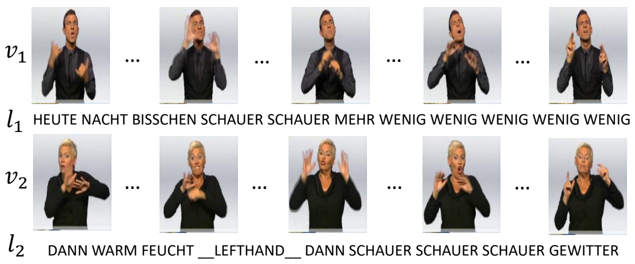
v_1
l_1
HEUTE NACHT BISSCHEN SCHAUER SCHAUER MEHR WENIG WENIG WENIG WENIG WENIG
v_2
l_2
DANN WARM FEUCHT __LEFTHAND__ DANN SCHAUER SCHAUER SCHAUER GEWITTER

Figure 7. Two examples of video-gloss pairs.

$d(\cdot,\cdot)$	$\mathbf{s}_{se}(v_1)$	$\mathbf{s}_{se}(v_2)$
$\mathbf{v}_{se}(v_1)$	0.01	1.99
$\mathbf{v}_{se}(v_2)$	1.76	0.37

Table 5. Examples of sentence embedding distances of the visual and sequential features. v_1 and v_2 are the videos in Figure 7.

Method	End-to-end	Dev	Test
CNN-LSTM-HMMs [24]	×	26.0	26.0
DNF (RGB) [11] + SBD-RL [48]	×	23.4	23.5
DNF [11]	×	23.1	22.9
CMA [37]	×	21.3	21.9
SMKD [17]	×	20.8	21.0
STMC [55]	×	21.1	20.7
SFL [32]	✓	24.9	25.3
FCN [10]	✓	23.7	23.9
VAC [31]	✓	21.2	22.3
C^{2}SLR (ours)	✓	**20.5**	**20.4**

Table 6. Comparison on PHOENIX-2014.

pairs denoted as (v_1, l_1) and (v_2, l_2). The sentence embedding distances of the visual and sequential features of v_1 and v_2 are shown in Table 5. It is clear that the distance for the same video (diagonal entries) can be very small. Otherwise (off-diagonal entries), the distance can be very large (the maximum value is 2.00).

4.4. Comparison with State-of-the-art Results

PHOENIX-2014. Table 6 shows a comprehensive comparison between other methods and ours on PHOENIX-2014. Although our C^{2}SLR is end-to-end trainable, it can outperform the SOTA work, STMC [55], which adopts the stage optimization strategy. To the best of our knowledge, this is the first time that an end-to-end method can outperform those using the stage optimization strategy.

PHOENIX-2014-T. We evaluate our method on PHOENIX-2014-T as shown in Table 7. Our method can outperform the SOTA one on the test set as well. Moreover, the performance of our proposed method is highly consistent on both dev and test sets. That means the CSLR model trained with the two proposed consistency constraints generalizes well for unseen data, and that makes it an advantage for real practice.

Method	End-to-end	Dev	Test
CNN-LSTM-HMMs [24]	×	22.1	24.1
SMKD [17]	×	20.8	22.4
STMC [55]	×	19.6	21.0
SFL [32]	✓	25.1	26.1
FCN [10]	✓	23.3	25.1
SLT [5]	✓	24.6	24.5
C^{2}SLR (ours)	✓	20.2	**20.4**

Table 7. Comparison on PHOENIX-2014-T.

Method	End-to-end	Test
LS-HAN [20]	×	17.3
Align-iOPT [39]	×	6.1
STMC [55]	×	2.1
CTF [47]	✓	11.2
HLSTM-attn [16]	✓	10.2
FCN [10]	✓	3.0
VAC [31]	✓	1.6
C^{2}SLR (ours)	✓	0.9
MSeqGraph* [44]	✓	**0.6**

Table 8. Comparison on CSL. *uses extra depth modality.

CSL. Finally, as shown in Table 8, we compare our method with others on CSL. It can achieve comparable performance to the SOTA work, MSeqGraph [44], which uses extra depth modality.

5. Conclusion

In this work, we propose two consistency constraints to enhance CSLR backbones. First, we insert a spatial attention module into the visual module and guide it by pre-extracted pose keypoints heatmaps, which can enforce the visual module to focus on informative regions. Second, we impose a sentence-level consistency constraint between the visual and sequential features, which can enhance the representation power of both the features. Extensive ablation studies validate the effectiveness of both the consistency constraints. More remarkably, our model can achieve SOTA or competitive performance on three benchmarks, while the whole model is trained in an end-to-end manner.

Acknowledgements. The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. HKUST16200118).

References

[1] Nikolaos M. Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, Vassia Zacharopoulou, George Xydopoulos, Klimis Antzakas, Dimitris Papazachariou, and Petros Daras. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE TMM*, pages 1–1, 2021. 3

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 4

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[4] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *CVPR*, 2018. 5

[5] Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *CVPR*, pages 10020–10030, 2020. 2, 6, 8

[6] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *CVPRW*, pages 0–0, 2019. 2

[7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *TPAMI*, 43(1):172–186, 2019. 2

[8] Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. Semantic re-tuning with contrastive tension. In *ICLR*, 2020. 3

[9] Shaoxiang Chen and Yu-Gang Jiang. Motion guided spatial attention for video captioning. In *AAAI*, volume 33, pages 8191–8198, 2019. 2

[10] Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai. Fully convolutional networks for continuous sign language recognition. In *ECCV*, volume 12369, pages 697–714, 2020. 1, 2, 5, 6, 8

[11] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE TMM*, PP:1–1, 07 2019. 1, 2, 8

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 2

[13] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *EMNLP*, 2021. 3

[14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, page 369–376, 2006. 1, 3, 5

[15] Dan Guo, Shuo Wang, Qi Tian, and Meng Wang. Dense temporal convolution network for sign language translation. In *IJCAI*, pages 744–750, 2019. 2

[16] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical LSTM for sign language translation. In *AAAI*, pages 6845–6852, 2018. 8

[17] Aiming Hao, Yuecong Min, and Xilin Chen. Self-mutual distillation learning for continuous sign language recognition. In *ICCV*, pages 11303–11312, October 2021. 1, 2, 4, 5, 6, 8

[18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019. 5

[19] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *NeurIPS*, 31:9401–9411, 2018. 2

[20] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI*, pages 2257–2264, 2018. 5, 8

[21] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 3, 4

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[23] Oscar Koller. Quantitative survey of the state of the art in sign language recognition. *arXiv preprint arXiv:2008.09918*, 2020. 2, 3

[24] Oscar Koller, Necati Camgoz, Hermann Ney, and Richard Bowden. Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE TPAMI*, 42(9):2306–2320, 04 2019. 2, 8

[25] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *CVIU*, 141:108–125, Dec. 2015. 5

[26] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid cnn-hmms. *IJCV*, 126(12):1311–1325, 2018. 2

[27] Xingze Li, Wengang Zhou, Yun Zhou, and Houqiang Li. Relation-guided spatial attention and temporal refinement for video-based person re-identification. In *AAAI*, volume 34, pages 11434–11441, 2020. 2

[28] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. In *ICLR*, 2018. 2

[29] Yue Liu, Xin Wang, Yitian Yuan, and Wenwu Zhu. Cross-modal dual learning for sentence-to-video generation. In *ACM MM*, pages 1239–1247, 2019. 3

[30] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421, 2015. 6

[31] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. Visual alignment constraint for continuous sign language recognition. In *ICCV*, pages 11542–11551, October 2021. 1, 2, 4, 5, 6, 7, 8

[32] Zhe Niu and Brian Mak. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *ECCV*, pages 172–186, 2020. 1, 2, 5, 6, 8

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[34] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM TASLP*, 24(4):694–707, 2016. 3

[35] Yanwei Pang, Jin Xie, Muhammad Haris Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Mask-guided attention network for occluded pedestrian detection. In *CVPR*, pages 4967–4975, 2019. 2

[36] Katerina Papadimitriou and Gerasimos Potamianos. Multimodal Sign Language Recognition via Temporal Deformable Convolutional Sequence Learning. In *Interspeech*, pages 2752–2756, 2020. 2

[37] Junfu Pu, Wengang Zhou, Hezhen Hu, and Houqiang Li. Boosting continuous sign language recognition via cross modality augmentation. In *ACM MM*, pages 1497–1505, 2020. 1, 2, 5, 6, 8

[38] Junfu Pu, Wengang Zhou, and Houqiang Li. Dilated convolutional network with iterative optimization for continuous sign language recognition. In *IJCAI*, pages 885–891, 2018. 1

[39] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *CVPR*, pages 4165–4174, 2019. 1, 2, 5, 8

[40] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, pages 3982–3992, 2019. 3

[41] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 5

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[43] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019. 2, 4

[44] Shengeng Tang, Dan Guo, Richang Hong, and Meng Wang. Graph-based multimodal sequential embedding for sign language translation. *IEEE TMM*, 2021. 8

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 5

[46] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 3156–3164, 2017. 2

[47] Shuo Wang, Dan Guo, Wen-gang Zhou, Zheng-Jun Zha, and Meng Wang. Connectionist temporal fusion for sign language translation. In *ACM MM*, pages 1483–1491, 2018. 8

[48] Chengcheng Wei, Jian Zhao, Wengang Zhou, and Houqiang Li. Semantic boundary detection with reinforcement learning for continuous sign language recognition. *IEEE TCSVT*, 31(3):1138–1149, 2020. 8

[49] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. 2, 3, 6, 7

[50] Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *ICLR*, 2018. 5

[51] Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. Modeling localness for self-attention networks. In *EMNLP*, pages 4449–4458, 2018. 6

[52] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219, 2019. 5

[53] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*, 2018. 4, 6

[54] Hao Zhou, Wengang Zhou, and Houqiang Li. Dynamic pseudo label decoding for continuous sign language recognition. In *ICME*, pages 1282–1287, 2019. 2, 5

[55] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*, pages 13009–13016, 2020. 1, 2, 3, 5, 6, 8

[56] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-temporal multi-cue network for sign language recognition and translation. *IEEE TMM*, pages 1–1, 2021. 2, 5