Self-Emphasizing Network for Continuous Sign Language Recognition

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

College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
hly2021,lqgao,lzk100953@tju.edu.cn,wfeng@ieee.org

Abstract

Hand and face play an important role in expressing sign language. Their features are usually especially leveraged to improve system performance. However, to effectively extract visual representations and capture trajectories for hands and face, previous methods always come at high computations with increased training complexity. They usually employ extra heavy pose-estimation networks to locate human body keypoints or rely on additional pre-extracted heatmaps for supervision. To relieve this problem, we propose a self-emphasizing network (SEN) to emphasize informative spatial regions in a self-motivated way, with few extra computations and without additional expensive supervision. Specifically, SEN first employs a lightweight subnetwork to incorporate local spatial-temporal features to identify informative regions, and then dynamically augment original features via attention maps. It's also observed that not all frames contribute equally to recognition. We present a temporal self-emphasizing module to adaptively emphasize those discriminative frames and suppress redundant ones. A comprehensive comparison with previous methods equipped with hand and face features demonstrates the superiority of our method, even though they always require huge computations and rely on expensive extra supervision. Remarkably, with few extra computations, SEN achieves new state-of-the-art accuracy on four large-scale datasets, PHOENIX14, PHOENIX14-T, CSL-Daily, and CSL. Visualizations verify the effects of SEN on emphasizing informative spatial and temporal features. Code is available at https://github.com/hulianyuyy/SEN_CSLR

Introduction

Sign language is one of the most commonly-used communication tools for the deaf community in their daily life. It mainly conveys information by both manual components (hand/arm gestures), and non-manual components (facial expressions, head movements, and body postures) (Dreuw et al. 2007; Ong and Ranganath 2005). 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[1]. Continuous sign language recognition (CSLR) progresses by sequentially translating image streams into a series of glosses to express a complete sentence, more prospective towards bridging the communication gap.

Figure 1: Visualization of class activation maps with Grad-CAM (Selvaraju et al. 2017) for VAC (Min et al. 2021) (baseline). Top: Original frames. Bottom: activation maps. It's observed that without extra supervision, it fails to locate discriminative face and hand regions precisely.

In sign language, the left hand, right hand, and face play the most important role in expressing glosses. Mostly, they convey the information through horizontal/vertical hand movements, finger activities, and static gestures, assisted with facial expressions and mouth shapes to holistically deliver messages (Dreuw et al. 2007; Ong and Ranganath 2005). As a result, hand and face, are always especially leveraged and incorporated in sign language systems. In isolated sign language recognition, early methods (Freeman and Roth 1995; Sun et al. 2013) leveraged hand-crafted features to describe the gestures and motion of both hands. Recent methods either choose to build a pure pose-based system (Tunga, Nuthalapati, and Wachs 2021; Hu et al. 2021) based on detected keypoints for both hands and face, or construct appearance-based systems (Hu, Zhou, and Li 2021; Boukhayma, Bem, and Torr 2019) with cropped patches for hands and face as collaborative inputs. In CSLR, CNN-LSTM-HMM (Koller et al. 2019) builds a multi-stream (hands and face) Hidden-Markov-Model (HMM) to integrate multiple visual inputs to boost recognition accuracy. STMC (Zhou et al. 2020) explicitly inserts a pose-estimation

*Corresponding author

[1]Gloss is the atomic lexical unit to annotate sign languages.

network and uses the detected regions (hand and face) as multiple cues to perform recognition. More recently, C^2SLR (Zuo and Mak 2022) leverages the pre-extracted pose keypoints heatmaps as additional supervision to guide models to focus on hand and face areas.

Although it has been proven effective to incorporate hand and face features to improve recognition performance for sign language systems, previous methods usually come at huge computations with increased training complexity, and rely on additional pose estimation networks or extra expensive supervision (e.g., heatmaps). However, without these supervision signals, we find current methods (Min et al. 2021; Hao, Min, and Chen 2021; Cheng et al. 2020) in CSLR fail to precisely locate the hand and face regions (Fig. 1). To more effectively excavate these key cues but avoid relying on expensive supervision, we propose a self-emphasizing network (SEN) to explicitly emphasize informative spatial regions in a self-motivated way. Specifically, SEN first employs a lightweight subnetwork to incorporate local spatial-temporal features to identify informative regions, and then dynamically emphasizes or suppresses input features via attention maps.

It's also observed that not all frames contribute equally to recognition. For example, frames with hand/arm movements of the signer are usually more important than those transitional frames. We present a temporal self-emphasizing module to emphasize those discriminative frames and suppress redundant ones dynamically. Remarkably, SEN yields new state-of-the-art accuracy upon four large-scale CSLR datasets, especially outperforming previous methods equipped with hand and face features, even though they always come at huge computations and rely on expensive supervision. Visualizations verify the effects of SEN in emphasizing spatial and temporal features. Code is available at https://github.com/hulianyuyy/SEN_CSLR

Related Work

Continuous Sign Language Recognition

Sign language recognition methods can be roughly categorized into isolated sign language recognition (Tunga, Nuthalapati, and Wachs 2021; Hu et al. 2021; Hu, Zhou, and Li 2021) and continuous sign language recognition (Pu, Zhou, and Li 2019; Cheng et al. 2020; Cui, Liu, and Zhang 2019; Niu and Mak 2020; Min et al. 2021) (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. Early methods in CSLR usually depend on hand-crafted features (Gao et al. 2004; Freeman and Roth 1995) to provide visual information, especially body gestures, hands, and face, or rely on HMM-based systems (Koller et al. 2016; Han, Awad, and Sutherland 2009; Koller, Zargaran, and Ney 2017; Koller, Forster, and Ney 2015) to perform temporal modeling and then translate sentences step by step. The HMM-based methods typically first employ a feature extractor to capture visual representations and then adopt an HMM to perform long-term temporal modeling. The recent success of convolutional neural networks (CNNs) and recurrent neural networks brings huge progress for CSLR. The widely-used CTC loss (Graves et al. 2006) enables end-to-end training for recent methods by aligning target glosses with inputs.

Especially, hands and face are paid close attention to by recent methods. For example, CNN-LSTM-HMM (Koller et al. 2019) employs a multi-stream HMM (including hands and face) to integrate multiple visual inputs to improve recognition accuracy. STMC (Zhou et al. 2020) utilizes a pose-estimation network to estimate human body keypoints and then sends cropped patches (including hands and face) for integration. More recently, C^2SLR (Zuo and Mak 2022) leverages the pre-extracted pose keypoints as supervision to guide the model. Despite high accuracy, they consume huge additional computations and training complexity.

Practically, recent methods (Pu, Zhou, and Li 2019; Pu et al. 2020; Cheng et al. 2020; Cui, Liu, and Zhang 2019; Niu and Mak 2020; Min et al. 2021) usually first employ a feature extractor to capture frame-wise visual representations for each frame, and then adopt 1D CNN and BiLSTM to perform short-term and long-term temporal modeling, respectively. However, several methods (Pu, Zhou, and Li 2019; Cui, Liu, and Zhang 2019) found in such conditions the feature extractor is not well trained and propose the iterative training strategy to refine the feature extractor, but consume much more computations. More recent methods try to directly enhance the feature extractor by adding visual alignment losses (Min et al. 2021) or adopt pseudo label (Cheng et al. 2020; Hao, Min, and Chen 2021) for supervision. We propose the self-emphasizing network to emphasize informative spatial features, which can be viewed to enhance the feature extractor in a self-motivated way.

Spatial Attention

Spatial attention has been proven to be effective in many fields including image classification (Cao et al. 2019; Hu et al. 2018; Woo et al. 2018; Hu, Shen, and Sun 2018), scene segmentation (Fu et al. 2019) and video classification (Wang et al. 2018). SENet (Hu, Shen, and Sun 2018), CBAM (Woo et al. 2018), SKNet (Li et al. 2019) and ECA-Net (Wang et al. 2020) devise lightweight channel attention modules for image classification. The widely used self-attention operator (Wang et al. 2018) employs dot-product feature similarities to build attention maps and aggregate long-term dependencies. However, the calculation complexity of the self-attention operator is quadratic to the incorporated pixels, incurring a heavy burden for video-based tasks (Wang et al. 2018). Instead of feature similarities, our SEN employs a learnable subnetwork to aggregate local spatial-temporal representations and generates spatial attention maps for each frame, much more lightweight than self-attention operators. Some works also propose to leverage external supervision to guide the spatial attention module. For example, GALA (Linsley et al. 2018) collects click maps from games to supervise the spatial attention for image classification. A relation-guided spatial attention module (Li et al. 2020) is designed to explore the discriminative regions globally for Video-Based Person Re-Identification. MGAN (Pang et al. 2019) introduces an attention network to emphasize visible pedestrian regions by modulating full body features. In con-

trast to external supervision, our self-emphasizing network strengthens informative spatial regions in a self-motivated way, thus greatly lowering required computations and training complexity.

Method

Framework Overview

As shown in fig. 2, the backbone of CSLR models is consisted of a feature extractor (2D CNN[2]), 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 to predict the probability of target gloss sequence $p(y|x)$.

To emphasize the informative spatial and temporal features for CSLR models, we present a spatial self-emphasizing module (SSEM) and a temporal self-emphasizing module (TSEM). Specifically, we incorporate them into the feature extractor to operate on each frame. Fig. 2 shows an example of a common feature extractor consisting of multiple stages with several blocks in each. We place the SSEM and TSEM in parallel before the 3×3 spatial convolution in each block to emphasize informative spatial and temporal features, respectively. When designing the architecture, efficiency is our core consideration, to avoid heavy computational burdens like previous methods (Zhou et al. 2020; Zuo and Mak 2022) based on heavy pose-estimation networks or expensive heatmaps. We next introduce our SSEM and TSEM, respectively.

Spatial Self-Emphasizing Module (SSEM)

From fig. 1, we argue current CSLR models fail to effectively leverage the informative spatial features, e.g., hands and face. We try to enhance the capacity of the feature extractor of CSLR models to incorporate such discriminative features without affecting its original spatial modeling ability. Practically, our SSEM is designed to first leverage the closely correlated local spatial-temporal features to identify the informative regions for each frame, and then augment original representations in the form of attention maps.

As shown in fig. 3, SSEM first projects the input features $s = \{s_t\}_{t=1}^{T} \in \mathcal{R}^{T\times C\times H\times W}$ into $s_r \in \mathcal{R}^{T\times C/r\times H\times W}$ to decrease the computational costs brought by SSEM, with r the reduction factor as 16 by default.

The frame-wise features s in the feature extractor are independently extracted for each frame by 2D convolutions, failing to incorporate local spatial-temporal features

[2]Here we only consider the feature extractor based on 2D CNN, because recent findings (Adaloglou et al. 2021; Zuo and Mak 2022) show 3D CNN can not provide as precise gloss boundaries as 2D CNN, and lead to lower accuracy.

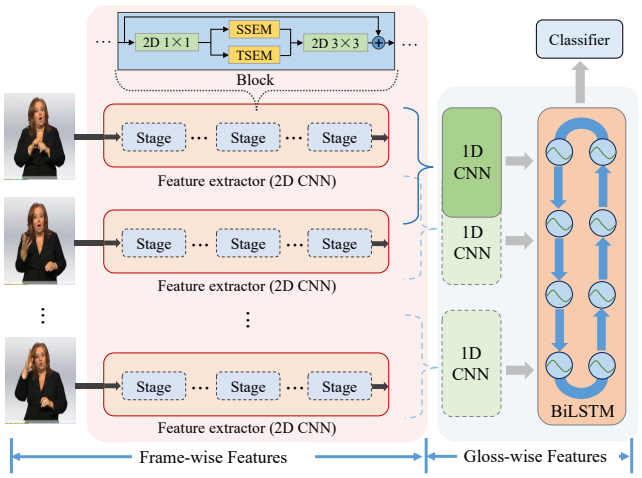

Figure 2: A overview for our SEN. It first employs a feature extractor (2D CNN) to capture frame-wise features, and then adopts a 1D CNN and a BiLSTM to perform short-term and long-term temporal modeling, respectively, followed by a classifier to predict sentences. We place our proposed spatial self-emphasizing module (SSTM) and temporal self-emphasizing module (TSEM) into each block of the feature extractor to emphasize the spatial and temporal features, respectively.

to distinguish the informative spatial regions. Besides, as the signer has to throw his/her arms and hands to express glosses, the informative regions in adjacent frames are always misaligned. Thus, we devise a multi-scale architecture to perceive spatial-temporal features in a large neighborhood to help identify informative regions.

Instead of a large spatial-temporal convolution kernel, we employ N parallel factorized branches with group-wise convolutions of progressive dilation rates to lower computations and increase the model capacity. As shown in fig. 3, these N branches own the same spatial-temporal kernel size $K_t \times K_s \times K_s$, with different spatial dilation rates $[1 \cdots N]$. Features from different branches are multiplied with learnable factors $\{\sigma_1 \dots \sigma_k\}$ to control the importance of different branches via gradient-based backward propagation, and are then added to mix information from different receptive fields. This multi-scale architecture is expressed as:

$$s_m = \sum_{i=1}^{N} \sigma_i \times \mathrm{Conv}_i(s_r) \tag{1}$$

where the group-wise convolution Conv_i at different levels captures spatial-temporal features from different receptive fields, with dilation rate $(1, i, i)$.

Especially, as the channels are downsized by r times in SSEM and we employ group-wise convolutions with small spatial-temporal kernels to capture multi-scale features, the overall architecture is rather lightweight with few ($<$0.1%) extra computations compared to the original model, as demonstrated in our ablative experiments.

Next, s_m is sent into a $1 \times 1 \times 1$ convolution to project

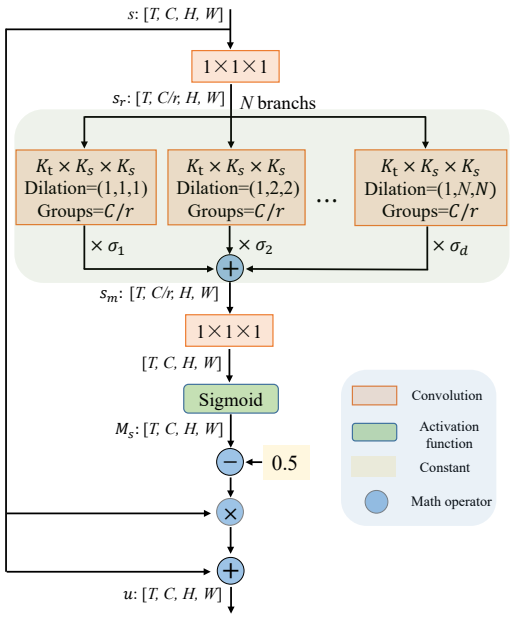

Figure 3: Illustration for our spatial self-emphasizing module (SSEM).

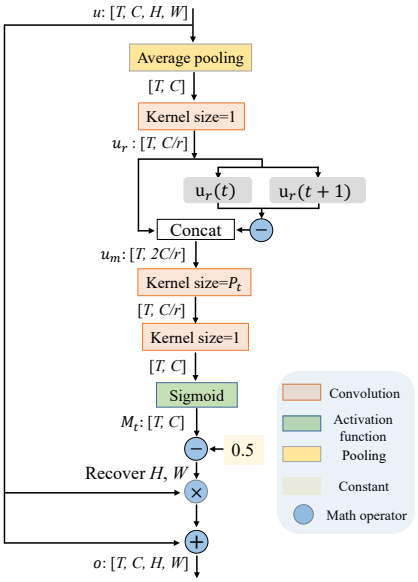

Figure 4: Illustration for our temporal self-emphasizing module (TSEM).

channels back into C, and then passed through a sigmoid activation function to generate attention maps $M_s \in \mathcal{R}^{T\times C\times H\times W}$ with values ranging between $[0, 1]$ as:

$$M_s = \text{Sigmoid}(\text{Conv}_{1\times1\times1}(s_m)) \tag{2}$$

Finally, the attention maps M_s are used to emphasize informative spatial regions for input features. To avoid hurting original representations and degrading accuracy, we propose to emphasize input features via a residual way as:

$$u = (M_s - 0.5 \times \mathbb{1}) \odot s + s \tag{3}$$

where $\odot$ denotes element-wise multiplication and u is the output.

In specific, we first subtract $0.5 \times \mathbb{1}$ from the attention maps M_s, with $\mathbb{1} \in \mathcal{R}^{T\times C\times H\times W}$ denoting an all-one matrix, to change the range of values in M_s into $[-0.5, 0.5]$. Then we element-wisely multiply the resulting attention maps with input features s to dynamically emphasize the informative regions and suppress unnecessary areas. Here, the values in M_s larger than 0 would strengthen the corresponding inputs features, otherwise they would weaken the input features. Finally, we add the modulated features with input features s to emphasize or suppress certain spatial features, but avoid hurting original representations.

Temporal Self-Emphasizing Module

We argue that not all frames in a video contribute equally to recognition, where some frames are more discriminative than others. For example, frames in which the signer moves his/her arms to express a sign are usually more important than those transitional frames or idle frames with meaningless contents. However, the feature extractor only employs 2D spatial convolutions to capture spatial features for each frame, equally treating frames without considering their temporal correlations. We propose a temporal self-emphasizing module (TSEM) to adaptively emphasize discriminative frames and suppress redundant ones.

As shown in fig. 4, input features $u \in \mathcal{R}^{T\times C\times H\times W}$ first undergo a global average pooling layer to eliminate the spatial dimension, i.e., H and W. Then these features pass through a convolution with kernel size of 1 to reduce channels by r times into $u_r \in \mathcal{R}^{T\times C/r}$ as:

$$u_r = \text{Conv}_{K=1}(\text{AvgPool}(u)) \tag{4}$$

where K denotes the kernel size. To better exploit local temporal movements to identify the discriminative frames, we leverage the temporal difference operator to incorporate motion information between adjacent frames to make decisions better. Specially, we calculate the difference between two adjacent frames for u_r as approximate motion information, and then concatenate it with appearance features u_r as :

$$u_m = \text{Concat}([u_r, u_r(t+1) - u_r]) \tag{5}$$

Next, we send u_m into a 1D temporal convolution with kernel size of P_t to capture the short-term temporal information. As the size of u_m is rather small, we here employ a normal temporal convolution instead of a multi-scale architecture. The features then undergo a convolution with kernel size of 1 to project channels back into C, and pass through a sigmoid activation function to generate attention maps $M_t \in \mathcal{R}^{T\times C}$ as:

$$M_t = \text{Sigmoid}(\text{Conv}_{K=1}(u_m)) \tag{6}$$

Finally, we employ M_t to emphasize the discriminative features for input u in a residual way as :

$$o = (M_t - 0.5 \times \mathbb{1}) \odot u + u \tag{7}$$

where $\odot$ denotes element-wise multiplication and o is the output.

Configurations	FLOPs	Dev(%)	Test(%)
-	3.64G	21.2	22.3
K_t=9, K_s=3, N=**1**	+0.4M	20.5	22.0
K_t=9, K_s=3, N=**2**	+0.6M	20.2	21.8
K_t=9, K_s=3, N=**3**	+0.8M	**19.9**	**21.4**
K_t=9, K_s=3, N=**4**	+1.0M	20.2	21.7
K_t=**7**, K_s=3, N=3	+0.7M	20.2	21.6
K_t=**11**, K_s=3, N=3	+1.0M	20.3	21.8
K_t=9, K_s=**7**, N=1	+2.9M	20.5	22.0

Table 1: Ablations for the multi-scale architecture of SSEM on the PHOENIX14 dataset.

Experiments

Experimental Setup

Datasets. **PHOENIX14** (Koller, Forster, and Ney 2015) and **PHOENIX14-T** (Camgoz et al. 2018) are both recorded from a German weather forecast broadcast before a clean background with a resolution of 210 $\times$ 260. They contain 6841/8247 sentences with a vocabulary of 1295/1085 signs, divided into 5672/7096 training samples, 540/519 development (Dev) samples and 629/642 testing (Test) samples.

CSL-Daily (Zhou et al. 2021) is recorded indoor with 20654 sentences, divided into 18401 training samples, 1077 development (Dev) samples and 1176 testing (Test) samples.

CSL (Huang et al. 2018) 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.

Training details. We adopt ResNet18 (He et al. 2016) as the 2D CNN with ImageNet (Deng et al. 2009) pretrained weights. We place SSEM and TSEM before the second convolution in each block. The 1D CNN consists of a sequence of {K5, P2, K5, P2} layers where K and P denotes a 1D convolutional layer and a pooling layer with kernel size of 5 and 2, respectively. We then adopt a two-layer BiLSTM with 1024 hidden states and a fully connected layer for prediction. We train our model for 80 epochs with initial learning rate 0.0001 decayed by 5 after 40 and 60 epochs. Adam optimizer is adopted with weight decay 0.001 and batch size 2. All frames are first resized to 256×256 and then randomly cropped to 224×224, with 50% horizontal flip and ±20% random temporal scaling during training. During inference, a central 224×224 crop is simply selected. We use VE and VA losses from VAC (Min et al. 2021) for extra supervision.

Evaluation Metric. We use Word Error Rate (WER) as the evaluation metric, which is defined as the minimal summation of the **sub**stitution, **ins**ertion, and **del**etion operations to convert the predicted sentence to the reference sentence, as:

$$\mathrm{WER} = \frac{\#\mathrm{sub} + \#\mathrm{ins} + \#\mathrm{del}}{\#\mathrm{reference}}. \tag{8}$$

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

Ablation Study

We perform ablation studies on the PHOENIX14 dataset and report on both development (Dev) and testing (Test) sets.

Configurations	Dev(%)	Test(%)
-	21.2	22.3
$M_s \odot s$	22.3	23.4
$M_s \odot s + s$	20.6	21.7
$(M_s - 0.5 \times \mathbb{1}) \odot s$	20.2	21.5
$(M_s - 0.5 \times \mathbb{1}) \odot s + s$	**19.9**	**21.4**

Table 2: Ablations for the implementations of SSEM to augment input features on the PHOENIX14 dataset.

Configurations	Dev(%)	Test(%)
-	19.9	21.4
u_r	19.8	21.2
$\mathrm{Concat}([u_r, u_r(t+1) - u_r])$	**19.5**	**21.0**
P_t = 7	19.6	21.2
P_t = 9	**19.5**	**21.0**
P_t = 11	19.7	21.3

Table 3: Ablations for TSEM on the PHOENIX14 dataset.

Effects of the multi-scale architecture of SSEM. Tab. 1 ablates the implementations for the multi-scale architecture of SSEM. Our baseline achieves 21.2% and 22.3% WER on the Dev and Test Set. When fixing K_t=9, K_s=3 and varying the number of branches to expand spatial receptive fields, it's observed larger N consistently brings better performance. When N reaches 3, it brings no more performance gain. We set N as 3 by default and test the effects of K_t. One can see that either increasing K_t to 11 or decreasing K_t to 7 achieves worse performance. We thus adopt K_t as 9 by default. Notably, one can find SSEM brings few extra computations compared to our baseline. For example, the best-performing SSEM with K_t=9, K_s=3 and N=3 only owns 0.8M (<0.1%) extra FLOPs, which can be neglected compared to 364G FLOPs of our baseline model. Finally, we compare our proposed multi-scale architecture with a normal implementation of more computations. The receptive field of SSEM with K_t=9, K_s=3 and N=3 is identical to a normal convolution with K_t=9 and K_s=7. As shown in the bottom of tab. 1, a normal convolution not only brings more computations than SSEM, but also performs worse, verifying the effectiveness of our architecture.

Implementations of SSEM to augment inputs features. Tab. 2 ablates the implementations of SSEM to augment original features. It's first observed directly multiplying the attention maps M_s with input features s severely degrades performance, attributed to destroying input features distributions. Implemented in a residual way by adding s, $M_s \odot s + s$ could notably relieve such phenomenon and achieves +0.6% & +0.6% on the Dev and Test Sets. Further, we first subtract $0.5 \times \mathbb{1}$ from the attention maps M_s to emphasize or suppress certain positions, and then element-wisely multiply it with s. This implementation bring +1.0% & +0.8% performance boost. Finally, we update this implementation in a residual way by adding input features s as $(M_s - 0.5 \times \mathbb{1}) \odot s + s$, achieving notable performance boost by +1.3% & +0.9%.

Configurations	Dev(%)	Test(%)
-	21.2	22.3
SSEM	19.9	21.4
TSEM	20.5	21.7
SSEM + TSEM	19.8	21.4
TSEM + SSEM	19.6	21.2
Parallelled	**19.5**	**21.0**

Table 4: Ablations for the effectiveness of SSEM and TSEM on the PHOENIX14 dataset.

Methods	Dev(%)	Test(%)
-	21.2	22.3
w/ SENet (Hu, Shen, and Sun 2018)	20.7	21.6
w/ CBAM (Woo et al. 2018)	20.5	21.3
CNN+HMM+LSTM (Koller et al. 2019)	26.0	26.0
STMC (Zhou et al. 2020)	21.1	20.7
C^2SLR (Zuo and Mak 2022)	20.5	20.4
SEN	**19.5**	**21.0**

Table 5: Comparison with other methods of channel attention or hand and face features on the PHOENIX14 dataset.

Study on TSEM. Tab. 3 ablates the configurations for TSEM. We here adopt SSEM as our baseline and ablate the configurations for TSEM. It's first noticed that combining motion information by concatenating $u_r(t+1) - u_r$ with u_r slightly outperforms only using u_r to capture short-term temporal dependencies, verifying the effectiveness of local motion information. Next, when varying P_t, it's observed P_t=9 achieves the best performance among P_t=[7,9,11], which is adopted by default in the following.

Study on the effectiveness of SSEM and TSEM. Tab. 4 studies how to combine SSEM with TSEM. We first notice that only using SSEM or TSEM could already bring a notable performance boost, by +1.3& +0.9% and +0.7 & +0.6% on the Dev and Test Sets, respectively. When further combining SSEM with TSEM by sequentially placing SSEM before TSEM (SSEM+TSEM), placing TSEM before SSEM (TSEM+SSEM) or paralleling TSEM and TSEM, it's observed SSEM+TSEM performs best with +1.7% & +1.3% performance boost on the Dev and Test Sets, respectively, adopted as the default setting.

Comparison with other methods. We compare our SEN with related well-known channel attention methods like SENet (Hu, Shen, and Sun 2018) and CBAM (Woo et al. 2018), and previous CSLR methods equipped with hand and face features by extra pose-estimation networks or pre-extracted heatmaps. In the upper part of tab. 5, one can see SEN largely outperforms these channel attention methods, for its superior ability to emphasize informative hand and face features. In the bottom part of tab. 5, it's observed SEN greatly surpasses previous CSLR methods equipped with hand and face features, even though they employ extra heavy networks or expensive supervision. These results verify the effectiveness of our SEN in leveraging hand and face features.

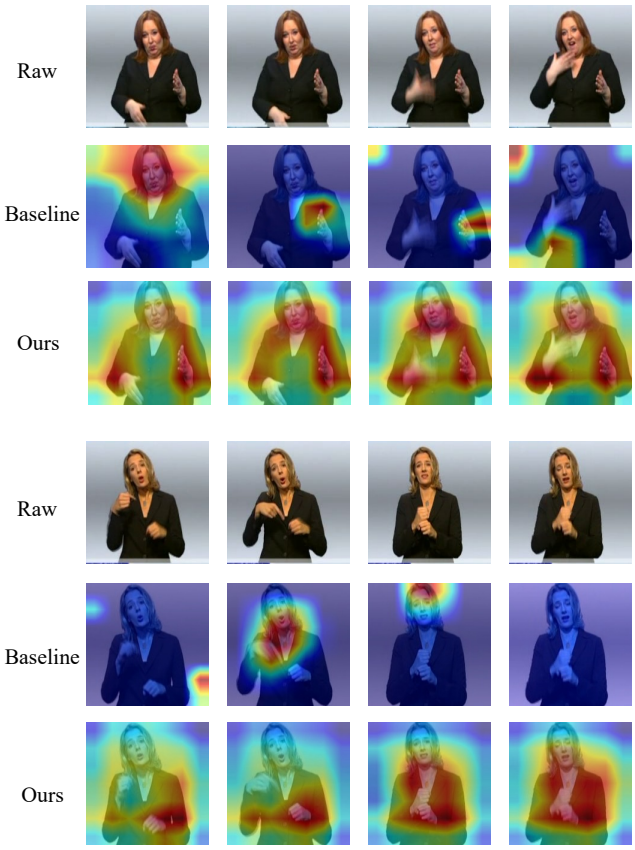

Figure 5: Visualizations of class activation maps by Grad-CAM (Selvaraju et al. 2017). Top: raw frames; Middle: class activation maps of our baseline; Bottom: class activation maps of our SEN. Our baseline usually focuses on nowhere or only attends to a single hand or face. Our SEN could generally focus on the human body (light yellow areas) and pays special attention to informative regions like hands and face (dark red areas).

Visualizations

Visualization for SSEM. We sample a few frames for expressing a gloss and plot the class activation maps for our baseline and SEN with Grad-CAM (Selvaraju et al. 2017) in fig. 5. The activation maps generated by our baseline usually focus on nowhere or only attend to a single hand or face, failing to fully focus on the informative regions (e.g., hands and face). Instead, our SEN could generally focus on the human body (light yellow areas), and pays special attention to those discriminative regions like hands and face (dark red areas). These visualizations show that without additional expensive supervision, our SEN could still effectively leverage the informative spatial features in a self-supervised way.

Visualization for TSEM. We visualize the temporal attention maps of TSEM in fig 6. We sample several frames corresponding to an output gloss 'nord' as an example. The darker color, the higher weight. One can find that TSEM tends to allocate higher weights for frames with rapid move-

Methods	Backbone	PHOENIX14				PHOENIX14-T	
		Dev(%)		Test(%)		Dev(%)	Test(%)
		del/ins	WER	del/ins	WER		
Align-iOpt (Pu, Zhou, and Li 2019)	3D-ResNet	12.6/2	37.1	13.0/2.5	36.7	-	-
Re-Sign (Koller, Zargaran, and Ney 2017)	GoogLeNet	-	27.1	-	26.8	-	-
SFL (Niu and Mak 2020)	ResNet18	7.9/6.5	26.2	7.5/6.3	26.8	25.1	26.1
FCN (Cheng et al. 2020)	Custom	-	23.7	-	23.9	23.3	25.1
CMA (Pu et al. 2020)	GoogLeNet	7.3/2.7	21.3	7.3/2.4	21.9	-	-
VAC (Min et al. 2021)	ResNet18	7.9/2.5	21.2	8.4/2.6	22.3	-	-
SMKD (Hao, Min, and Chen 2021)	ResNet18	6.8/2.5	20.8	6.3/2.3	21.0	20.8	22.4
SLT* (Camgoz et al. 2018)	GoogLeNet	-	-	-	-	24.5	24.6
CNN+LSTM+HMM* (Koller et al. 2019)	GoogLeNet	-	26.0	-	26.0	22.1	24.1
DNF* (Cui, Liu, and Zhang 2019)	GoogLeNet	7.3/3.3	23.1	6.7/3.3	22.9	-	-
STMC* (Zhou et al. 2020)	VGG11	7.7/3.4	21.1	7.4/2.6	20.7	19.6	21.0
C^2SLR* (Zuo and Mak 2022)	ResNet18	-	20.5	-	20.4	20.2	20.4
Baseline	ResNet18	7.9/2.5	21.2	8.4/2.6	22.3	21.1	22.8
SEN (Ours)	ResNet18	5.8/2.6	**19.5**	7.3/4.0	**21.0**	**19.3**	**20.7**

Table 6: 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.

Figure 6: Visualizations of temporal attention maps for TSEM. One can find that TSEM highlight frames with rapid movements and suppress those static frames.

ments (the latter two frames in the first line; the middle three frames in the second line). TSEM assigns lower weights for static frames with few body movements. Such observation is consistent with our habits, as humans always pay more attention to those moving objects in the visual field to capture key movements. Those frames can also be considered conveying more important pattern for expressing a sign.

Comparison with State-of-the-Art Methods

PHOENIX14 and **PHOENIX14-T**. Tab. 6 shows a comprehensive comparison between our SEN and other state-of-the-art methods. We notice that with few extra computations, SEN could outperform other state-of-the-art methods upon both datasets. Especially, SEN 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, covering daily contents. Tab. 7 shows that our SEN achieves new state-of-the-art accuracy upon this challenging dataset with large progresses, which generalizes well upon real-world scenarios.

Methods	Dev(%)	Test(%)
LS-HAN (Huang et al. 2018)	39.0	39.4
TIN-Iterative (Cui, Liu, and Zhang 2019)	32.8	32.4
Joint-SLRT (Camgoz et al. 2020)	33.1	32.0
FCN (Cheng et al. 2020)	33.2	32.5
BN-TIN (Zhou et al. 2021)	33.6	33.1
Baseline	32.8	32.3
SEN(Ours)	**31.1**	**30.7**

Table 7: Comparison with state-of-the-art methods on the CSL-Daily dataset (Zhou et al. 2021).

Methods	WER(%)
SubUNet (Cihan Camgoz et al. 2017)	11.0
SF-Net (Yang et al. 2019)	3.8
FCN (Cheng et al. 2020)	3.0
STMC (Zhou et al. 2020)	2.1
VAC (Min et al. 2021)	1.6
C^2SLR (Zuo and Mak 2022)	0.9
Baseline	3.5
SEN(Ours)	**0.8**

Table 8: Comparison with state-of-the-art methods on the CSL dataset (Huang et al. 2018).

CSL. As shown in tab. 8, our SEN could achieve extreme superior accuracy (0.8% WER) upon this well-examined dataset, outperforming existing CSLR methods.

Conclusion

This paper proposes a self-motivated architecture, coined as SEN, to adaptively emphasize informative spatial and temporal features. Without extra expensive supervision, SEN outperforms existing CSLR methods upon four CSLR datasets. Visualizations confirm the effectiveness of SEN in leveraging discriminative hand and face features.

References

Adaloglou, N.; Chatzis, T.; Papastratis, I.; Stergioulas, A.; Papadopoulos, G. T.; Zacharopoulou, V.; Xydopoulos, G. J.; Atzakas, K.; Papazachariou, D.; and Daras, P. 2021. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, 24: 1750–1762.

Boukhayma, A.; Bem, R. d.; and Torr, P. H. 2019. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10843–10852.

Camgoz, N. C.; Hadfield, S.; Koller, O.; Ney, H.; and Bowden, R. 2018. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7784–7793.

Camgoz, N. C.; Koller, O.; Hadfield, S.; and Bowden, R. 2020. 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*, 10023–10033.

Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 0–0.

Cheng, K. L.; Yang, Z.; Chen, Q.; and Tai, Y.-W. 2020. Fully convolutional networks for continuous sign language recognition. In *ECCV*.

Cihan Camgoz, N.; Hadfield, S.; Koller, O.; and Bowden, R. 2017. Subunets: End-to-end hand shape and continuous sign language recognition. In *ICCV*.

Cui, R.; Liu, H.; and Zhang, C. 2019. A deep neural framework for continuous sign language recognition by iterative training. *TMM*, 21(7): 1880–1891.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dreuw, P.; Rybach, D.; Deselaers, T.; Zahedi, M.; and Ney, H. 2007. Speech recognition techniques for a sign language recognition system. *hand*, 60: 80.

Freeman, W. T.; and Roth, M. 1995. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition*, volume 12, 296–301. Zurich, Switzerland.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3146–3154.

Gao, W.; Fang, G.; Zhao, D.; and Chen, Y. 2004. A Chinese sign language recognition system based on SOFM/SRN/HMM. *Pattern Recognition*, 37(12): 2389–2402.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376.

Han, J.; Awad, G.; and Sutherland, A. 2009. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6): 623–633.

Hao, A.; Min, Y.; and Chen, X. 2021. Self-Mutual Distillation Learning for Continuous Sign Language Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11303–11312.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, H.; Zhao, W.; Zhou, W.; Wang, Y.; and Li, H. 2021. SignBERT: Pre-Training of Hand-Model-Aware Representation for Sign Language Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11087–11096.

Hu, H.; Zhou, W.; and Li, H. 2021. Hand-model-aware sign language recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1558–1566.

Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Vedaldi, A. 2018. Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in neural information processing systems*, 31.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, J.; Zhou, W.; Zhang, Q.; Li, H.; and Li, W. 2018. Video-based sign language recognition without temporal segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Koller, O.; Camgoz, N. C.; Ney, H.; and Bowden, R. 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *PAMI*, 42(9): 2306–2320.

Koller, O.; Forster, J.; and Ney, H. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141: 108–125.

Koller, O.; Zargaran, O.; Ney, H.; and Bowden, R. 2016. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *Proceedings of the British Machine Vision Conference 2016*.

Koller, O.; Zargaran, S.; and Ney, H. 2017. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *CVPR*.

Li, X.; Wang, W.; Hu, X.; and Yang, J. 2019. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 510–519.

Li, X.; Zhou, W.; Zhou, Y.; and Li, H. 2020. Relation-guided spatial attention and temporal refinement for video-based person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11434–11441.

Linsley, D.; Shiebler, D.; Eberhardt, S.; and Serre, T. 2018. Learning what and where to attend. *arXiv preprint arXiv:1805.08819*.

Min, Y.; Hao, A.; Chai, X.; and Chen, X. 2021. Visual Alignment Constraint for Continuous Sign Language Recognition. In *ICCV*.

Niu, Z.; and Mak, B. 2020. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *ECCV*.

Ong, S. C.; and Ranganath, S. 2005. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(06): 873–891.

Pang, Y.; Xie, J.; Khan, M. H.; Anwer, R. M.; Khan, F. S.; and Shao, L. 2019. Mask-guided attention network for occluded pedestrian detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4967–4975.

Pu, J.; Zhou, W.; Hu, H.; and Li, H. 2020. Boosting continuous sign language recognition via cross modality augmentation. In *ACM MM*.

Pu, J.; Zhou, W.; and Li, H. 2019. Iterative alignment network for continuous sign language recognition. In *CVPR*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Sun, C.; Zhang, T.; Bao, B.-K.; Xu, C.; and Mei, T. 2013. Discriminative exemplar coding for sign language recognition with kinect. *IEEE Transactions on Cybernetics*, 43(5): 1418–1428.

Tunga, A.; Nuthalapati, S. V.; and Wachs, J. 2021. Pose-based sign language recognition using gcn and bert. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 31–40.

Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; and Hu, Q. 2020. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Seattle, WA, USA*, 13–19.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Yang, Z.; Shi, Z.; Shen, X.; and Tai, Y.-W. 2019. SF-Net: Structured feature network for continuous sign language recognition. *arXiv preprint arXiv:1908.01341*.

Zhou, H.; Zhou, W.; Qi, W.; Pu, J.; and Li, H. 2021. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1316–1325.

Zhou, H.; Zhou, W.; Zhou, Y.; and Li, H. 2020. Spatial-temporal multi-cue network for continuous sign language recognition. In *AAAI*.

Zuo, R.; and Mak, B. 2022. C2SLR: Consistency-Enhanced Continuous Sign Language Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5131–5140.