Self-Mutual Distillation Learning for Continuous Sign Language Recognition

Aiming Hao[1,2], Yuecong Min[1,2], Xilin Chen[1,2]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
{aiming.hao,yuecong.min}@vipl.ict.ac.cn, xlchen@ict.ac.cn

Abstract

In recent years, deep learning moves video-based Continuous Sign Language Recognition (CSLR) significantly forward. Currently, a typical network combination for CSLR includes a visual module, which focuses on spatial and short-temporal information, followed by a contextual module, which focuses on long-temporal information, and the Connectionist Temporal Classification (CTC) loss is adopted to train the network. However, due to the limitation of chain rules in back-propagation, the visual module is hard to adjust for seeking optimized visual features. As a result, it enforces that the contextual module focuses on contextual information optimization only rather than balancing efficient visual and contextual information. In this paper, we propose a Self-Mutual Knowledge Distillation (SMKD) method, which enforces the visual and contextual modules to focus on short-term and long-term information and enhances the discriminative power of both modules simultaneously. Specifically, the visual and contextual modules share the weights of their corresponding classifiers, and train with CTC loss simultaneously. Moreover, the spike phenomenon widely exists with CTC loss. Although it can help us choose a few of the key frames of a gloss, it does drop other frames in a gloss and makes the visual feature saturation in the early stage. A gloss segmentation is developed to relieve the spike phenomenon and decrease saturation in the visual module. We conduct experiments on two CSLR benchmarks: PHOENIX14 and PHOENIX14-T. Experimental results demonstrate the effectiveness of the SMKD.

1. Introduction

As spoken language in speaking-hearing person's daily conversation, sign language plays the most important role for hearing-impaired person's communication. Sign language is used by millions of people all over the world. Different to spoken language, sign language conveys meaning by manual elements (e.g., hand configuration) and non-manual elements (e.g., facial expressions) [24], and has its own vocabulary and grammar. These characters make speaking-hearing person hard to understand sign language. Automatic Sign Language Recognition (SLR) provides a bridge to overcome this gap.

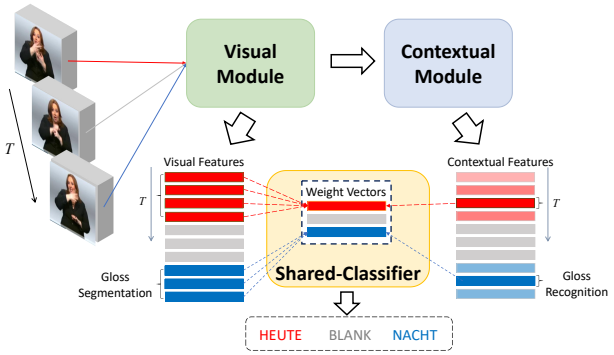


Figure 1. Overview of the proposed SMKD method. With the help of the shared-classifier, the visual and contextual modules are attempted to align the features at gloss level. This makes the two modules focus more on spatial-temporal information. To explore the short-term temporal information, a gloss segmentation is proposed into the visual module.

Different to the video-based isolated SLR, which recognizes a gloss-wise clip into its corresponding gloss (i.e., written words that represent signs), video-based Continuous Sign Language Recognition (CSLR) is a much more complicated task that aims to translate a sign language video into its corresponding sign gloss sequence. Due to the enormous cost of creating frame-level annotations, most CSLR dataset only has sentence-level annotations, and researchers often treat video-based CSLR as a weakly supervised problem [6, 3]. This further increases the difficulty of the task. To address these problems, some recent works [6, 26] adopt a deep network to deal with the video-based CSLR. The network consists of a visual module to extract the short-term spatial-temporal information of the input sequence and followed by a contextual module to encode the long-term contextual information. To train the designed network, the Connectionist Temporal Classification (CTC) [9] loss is used

to search the alignment between extracted features and the corresponding labeling.

However, end-to-end training makes the visual module hard to learn effective visual features as the penalty is hard to conduct from the contextual module [26, 34]. This makes the contextual module tends to over fit on the contextual information like sequential order of sign action instead of seeking optimize visual information [3]. Representation power of visual module isn't explored enough. Meanwhile, limited scale of the datasets makes network's performance decrease quickly on test set. To exploit the visual module's potential, some works [6, 35] propose to learn explicit visual features by optimizing the visual module directly with an auxiliary task, and these enhancements on the visual features optimization improve the generalization ability of the whole network. However, it is not a good choice to train visual module independently as that will lose the cooperation between visual and contextual modules as shown in [22].

In this study, we aim to enforce the contextual module to focus more on the visual information and strengthen the discriminative capability of the visual module to ensure powerful visual features. To achieve this, we propose a knowledge distillation method named Self-Mutual Knowledge Distillation (SMKD) that lets the visual module and contextual module share the weights of their corresponding classifiers and perform the CTC training simultaneously. The SMKD is inspired by two facts: 1) the CTC loss can be viewed as an iterative softmax loss, as shown in [19]; 2) according to [27], the classifier weight vector can be treated as a prototype of their respective class, and they can be used to guide the feature learning of the network with softmax loss. Based on these two facts, the weights of the visual and contextual modules are shared initially to enforce them to produce features as consistent as possible. With the feature alignment at the gloss level, the discriminative power of the visual features is enhanced, and the contextual module is enforced to focus more on the visual feature sequence.

In addition, as CTC loss will bring the spike phenomenon [19, 8], which causes only a few key frames contribute to final result and makes the visual module lose its discriminative power for the other frames. To explore the short-term temporal information which is depressed during CTC constrained training, we further propose to add a gloss segmentation into the visual module, where the pseudo gloss segment label is produced by a proposed Gloss Segment Boundary Assignment (GSBA) algorithm. Benefiting from the above mechanism, the weight vector of each gloss can feed more spatial-temporal information into the contextual module for better generalization capability. An overview of our proposed method is shown in Fig. 1.

Notably, we will decouple the weight matrix sharing between the two modules during the final training stage to relax the constraints on the contextual module, and make it focuses on long-term temporal information. We conduct extensive experiments on two CSLR benchmarks to demonstrate the effectiveness of the SMKD. In summary, the major contributions of our work are as follows:

- A SMKD method is proposed to enforce the visual and contextual module to focus more on spatial-temporal information, and it also can strengthen the discriminative power of both modules simultaneously.
- A gloss segmentation is developed to relief the spike phenomenon caused by CTC constraint and decrease saturation in visual module during the model training.

2. Related work

2.1. Continuous Sign Language Recognition

The learning process of recent works [6, 3, 23] can be summarized on three aspects: feature extraction, recognition, and alignment. Most of feature extraction of recent CSLR systems are composed of the visual module (Conv2D [23], Conv2D+Conv1D [6, 3] or Conv3D [34, 26]) and contextual module (RNN [6] or Transformer [23, 2]). For each input sequence, the visual module encodes short-term spatial-temporal information into visual features. Then taking the visual features as input, the contextual module encodes long-term context information into context features. Based on the extracted features, the classifier can get a posterior probability for each frame for the recognition. Since the video streams are continuously in CSLR, the alignment module is required to find the proper alignment between clips and glosses to ensure the training procedure. Methods like [17, 18, 15] align the video frames to glosses by applying Viterbi search on the Hidden Markov Models (HMMs). While some others [6, 3] adopt the CTC constraint, where a soft full-sum alignment is calculated as the final training objective.

However, just as some works have discovered, end-to-end training cannot fully exploit the deep neural network of high complexity [6, 23]. Some works address this problem by adding an auxiliary loss. For example, Cui *et al*. [6] uses the pseudo labels produced by the contextual module to supervise the visual module, and Cheng *et al*. [3] proposes a Gloss Feature Enhancement (GFE) module to improve the quality of visual features. Benefiting from the enhancement of the visual features, the whole module's generalization ability is also improved. Different to the above methods, we propose a SMKD method to strengthen the discriminative power of both modules by sharing weights between visual and contextual modules for better feature extraction.

2.2. Knowledge Distillation

Knowledge distillation (KD) is an effective learning method to transfer the knowledge from teacher model to

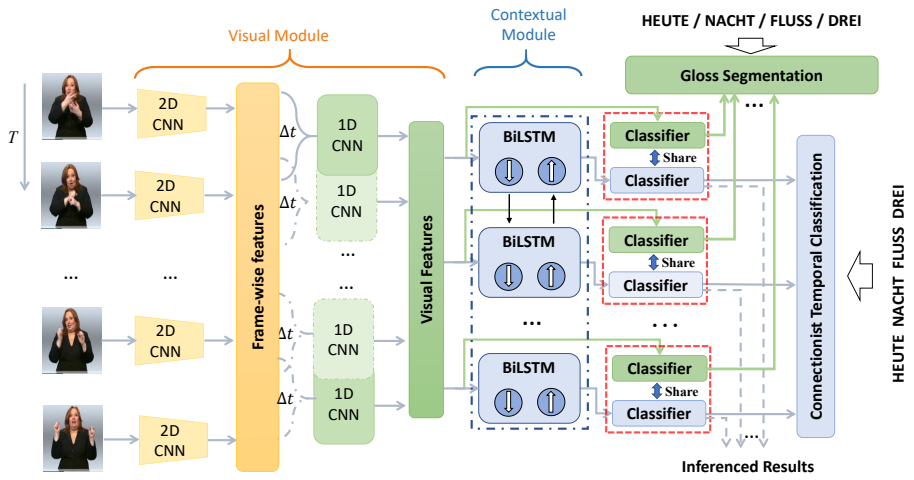


Figure 2. Flowchart of proposed SMKD. The procedure consists of visual extraction with a 2D-CNN+1D-CNN (visual module) and contextual integration with a BiLSTM (contextual module). The visual and contextual modules share the weights of their corresponding classifiers, and train with CTC and gloss segmentation simultaneously. Note that, only the predicted sequence from the contextual module contributes to recognition during the inference stage, as shown in the dashed line blocks.

a student model, which is achieved by providing a soft target [11] or directly inheriting the teacher's classifier [7, 28]. However, as shown in [32], there are two problems in naive knowledge distillation: low efficiency on knowledge transfer and difficulty in designing a proper teacher model. To solve these problems, Zhang *et al*. [32] proposed the idea of self-distillation that distill the knowledge from a model itself to enhance the generalization performance. Besides, some works [33, 31] propose to dispense with the teacher altogether, and allow an ensemble of students to teach each other, which is called mutual-distillation. In this work, we propose treating the different modules of the model as multiple student networks and achieving knowledge transfer by sharing their corresponding weight matrix.

3. Self-Mutual Knowledge Distillation

A SMKD method is proposed to exploit the power of visual module and relieve the overfitting problem. We first present the framework and formulation of the proposed method (Sect. 3.1). Then, we revisit the CTC loss (Sect. 3.2) and give details of weight sharing which promote visual module's contribution (Sect. 3.3). After that, we show the spike phenomenon and its drawback in Sect. 3.4, followed by a proposed solution of adding gloss segmentation to enforce the visual module provide visual features in more frames (Sect. 3.5). At last, we propose a three-stage optimization approach for the network's training (Sect. 3.6).

3.1. Framework and Formulation

Given an image sequence $\boldsymbol{X} = \left\{\boldsymbol{x}_t \in \mathbb{R}^{h\times w\times c}\right\}_{t=1}^{T}$ with T images, the CSLR aims to learn a mapping that transform the image sequence to its corresponding gloss label sequence $\boldsymbol{l} = \{l_i \in \mathbb{G}\}_{i=1}^{N}$ with N glosses, where $\mathbb{G}$ is the gloss vocabulary. To model the mapping, the proposed method contains three components as mentioned in Sect. 2.1. The structure of proposed method is presented in Fig. 2 and the details are as follows.

Feature Extraction. The visual module E_v is formed by a 2D-CNN and 1D-CNN, which encode spatial and short-term temporal information, respectively. After that, we get the Local Visual Features (LVFs):

$$\boldsymbol{V} = \left\{\boldsymbol{v}_t \in \mathbb{R}^d\right\}_{t=1}^{\hat{T}} \Rightarrow \boldsymbol{v}_t = E_v\left(\left[\boldsymbol{x}_{t-r/2}, ..., \boldsymbol{x}_{t+r/2}\right]\right), \tag{1}$$

where $\hat{T} = T/\delta$ represents the temporal duration of LVFs, δ is the downsampling rate and r denotes the temporal receptive filed of the visual module. For the contextual module E_g, a two layers BiLSTM is utilized to encode the visual information provided by the visual module and store the long-term context information with an internal state. Then, the Global Contextual Features (GCFs) are obtained:

$$\boldsymbol{G} = \left\{\boldsymbol{g}_t \in \mathbb{R}^d\right\}_{t=1}^{\hat{T}} \Rightarrow \boldsymbol{g}_t = E_g\left(\left[\boldsymbol{v}_1, ..., \boldsymbol{v}_{\hat{T}}\right]\right). \tag{2}$$

Recognition. Similar to the A-softmax loss [20], we normalize the classifier's weights $\boldsymbol{W}$ and ignore its bias term (i.e., $\|\boldsymbol{w}_i\| = 1,\ b_i = 0$). Given a learned feature vector $\boldsymbol{f}_t$, the logit of $\boldsymbol{f}_t$ at class c is got as:

$$z_t^c = \boldsymbol{w}_c \cdot \boldsymbol{f}_t = \|\boldsymbol{f}_t\| \cos\theta_t^c, \tag{3}$$

where $\boldsymbol{Z} = \left\{\boldsymbol{z}_t \in \mathbb{R}^{|\mathbb{G}|+1}\right\}_{t=1}^{\hat{T}}$ is the logits before the softmax activation function and θ_t^c denotes the angle between $\boldsymbol{w}_c$ and $\boldsymbol{f}_t$. Based on the similarity between extracted fea-

ture and weight vector, the network get the predict probability of glosses as:

$$\hat{\boldsymbol{Y}} = \text{softmax}\left(\boldsymbol{z}\right) = \left\{\hat{\boldsymbol{y}}_t \in \mathbb{R}^{|\mathbb{G}|+1}\right\}_{t=1}^{\hat{T}}. \tag{4}$$

Alignment. To align the predicted gloss sequence with the target gloss sequence, we adopt the CTC loss for alignment. The following parts of this section will elaborate on the CTC and its training process in detail.

3.2. Revisiting the CTC Loss

CTC is a popular sequence learning algorithm, which decodes the sign gloss sequence from the probability distribution $\hat{\boldsymbol{Y}}$ by introducing a blank label as an assistant token. Define a path $\pi = \{\pi_t\}_{t=1}^{\hat{T}}, \pi_t \in \mathbb{G} \cup \{blank\}$. Given the length $\hat{T}$ feature sequence $\boldsymbol{F}$, the conditional probability of observing a particular path π is calculated as:

$$p\left(\pi|\boldsymbol{F}\right) = \prod_{t=1}^{\hat{T}} p\left(\pi_t|\boldsymbol{f}_t\right) = \prod_{t=1}^{\hat{T}} \hat{y}_t^{\pi_t}. \tag{5}$$

To get the final decoded sequence without blanks, CTC defines a many-to-one function $\mathcal{B} : \left(\mathbb{G} \cup \{blank\}\right)^{\hat{T}} \rightarrow \mathbb{G}^{\leq \hat{T}}$, which removes the repeated labels and blanks. The probability of the sign gloss sentence $\boldsymbol{l}$ decoded by CTC is the summation of the probabilities for all possible paths as:

$$p\left(\boldsymbol{l}|\boldsymbol{F}\right) = \sum_{\pi \in \mathcal{B}^{-1}(\boldsymbol{l})} p\left(\pi|\boldsymbol{F}\right). \tag{6}$$

The CTC loss function is calculated as the negative log probability of correctly labelling the sequence:

$$\mathcal{L}_{\text{CTC}}\left(\boldsymbol{l}, \hat{\boldsymbol{Y}}\right) = -\ln p\left(\boldsymbol{l}|\hat{\boldsymbol{Y}}\right). \tag{7}$$

The CTC loss can be reinterpreted as an iterative softmax loss, which produces the pseudo ground truth probability distribution $\boldsymbol{Y} = \left\{\boldsymbol{y}_t \in \mathbb{R}^k\right\}_{t=1}^{\hat{T}}$ and pseudo labels $\boldsymbol{c} = \left\{c_t = \arg\max\left(y_t\right)\right\}_{t=1}^{\hat{T}}$ for input sequence iteratively. After getting the pseudo ground truth $\boldsymbol{y}$, the CTC loss will behave the same gradient backpropagation with the softmax loss. Per [19] results, we have:

$$\frac{\partial \mathcal{L}_{\text{CTC}}\left(\boldsymbol{l}, \hat{\boldsymbol{Y}}\right)}{\partial \boldsymbol{z}} = \frac{\partial \mathcal{L}_{\text{CE}}\left(\boldsymbol{Y}, \hat{\boldsymbol{Y}}\right)}{\partial \boldsymbol{z}} = \frac{\partial \sum_{t=1}^{\hat{T}} \boldsymbol{y}_t \log \hat{\boldsymbol{y}}_t}{\partial \boldsymbol{z}}. \tag{8}$$

where $\mathcal{L}_{\text{CE}}$ represents the cross-entropy loss. The same gradient backpropagation makes some observations about the softmax loss [27, 29] are also fit for the CTC loss. Specifically, the learned classifier's weights can be treated as classwise prototypes. Given the pseudo ground truth $\boldsymbol{Y}$, the CTC loss will optimize the similarity between training samples and classwise prototypes, maximizing the intra-class similarity as well as minimizing the inter-class similarity.

According to the observations in [19] and the relationship between features and weights mentioned above, the training process of the network with the CTC loss can be summarized as: Given a frame's feature $\boldsymbol{f}_t$, the CTC loss will produce a pseudo ground truth probability distribution $\boldsymbol{y}_t$ and assign a pseudo label c for it initially. In such a way, the $\boldsymbol{f}_t$ will be treated as a class-relevant feature of class c, so the similarity between $\boldsymbol{f}_t$ and $\boldsymbol{w}_c$ will increase. With the update of $\boldsymbol{W}$ and $\boldsymbol{F}$, the network will update the assignment for each frame until convergence. After that, the probability of pseudo label y_t^c for each frame will be increased gradually, and the training of the network will carry on with the convergent assignment.

3.3. Sharing the Weight Matrix

As the contextual module learns from all the frame information, it tends to "remember" all the signing sequences in the training set instead of learning the glosses independently. This leads to an overfitting issue during training as shown in [3]. Based on the above analysis of the network's training process with CTC loss, we propose an Weight Sharing (WS) method that shares the same classifier's weights to instruct the feature learning of the visual module and contextual module and relief the overfitting issue. There are two insights for this operation:

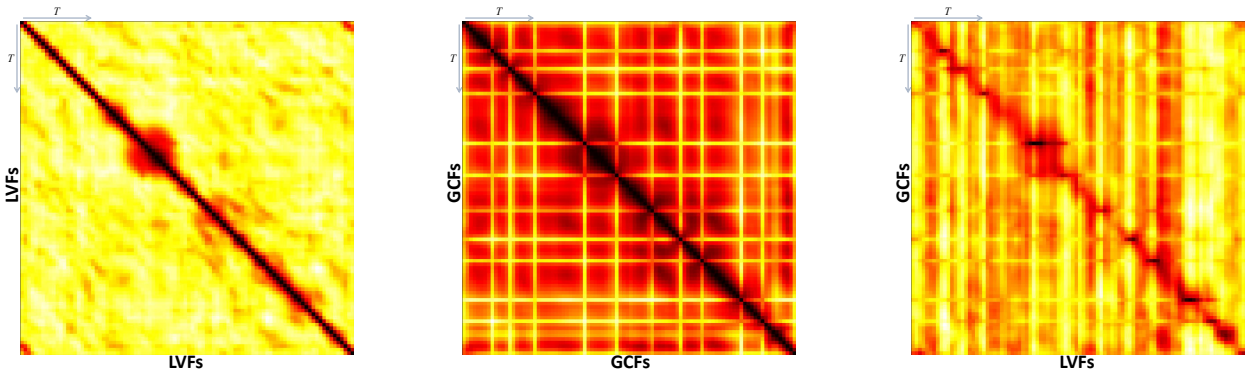
1) Given a gloss c, assume the LVF $\boldsymbol{v}^c$ and GCF $\boldsymbol{g}^c$ corresponding to it are passing two virtual classifiers C_v and C_g to produce prediction. Although the two classifiers most likely are different, they should keep the same tendency, i.e., $\|C_g\left(\boldsymbol{g}^c\right) - C_v\left(\boldsymbol{v}^c\right)\| \rightarrow 0$. Once, we release the model to unified classifier on these two features, then we have an equivalent $\|\boldsymbol{g}^c - \boldsymbol{v}^c\| \rightarrow 0$. With the feature alignment at the gloss level, this will enforce the visual module to enhance visual feature extraction, and restrict the contextual module to focus more on the short-term spatial-temporal information.

2) The visual and contextual modules can be treated as two student networks, they construct a general weight matrix, which balances the contributions from these two modules, for their feature learning.

Specifically, the LVFs $\boldsymbol{V}$ and the GCFs $\boldsymbol{G}$ will pass the same classifier, and get the predict probability distributions $\hat{\boldsymbol{Y}}_v$ and $\hat{\boldsymbol{Y}}_g$, respectively. Then, we use the CTC loss to train the whole network. The overall objective $\mathcal{L}_{\text{WS}}$ becomes:

$$\mathcal{L}_{\text{WS}} = \mathcal{L}_{\text{CTC}}\left(\boldsymbol{l}, \hat{\boldsymbol{Y}}_g; \boldsymbol{W}\right) + \alpha \cdot \mathcal{L}_{\text{CTC}}\left(\boldsymbol{l}, \hat{\boldsymbol{Y}}_v; \boldsymbol{W}\right), \tag{9}$$

where α is the tunable hyper-parameter that balances the contributions between the visual and contextual modules.



(a) Self similarity matrix of the LVFs. (b) Self similarity matrix of the GCFs. (c) Similarity matrix between the LVFs and GCFs.

Figure 3. The heatmap of the LVFs' and GCFs' self-similarity matrices and the similarity matrix between the LVFs and the GCFs (**the darker color represents the higher similarity**).

3.4. Visualizing the Similarity Matrix

To explore the property of the features after sharing the weights, we select an instance (01April_2010_Thursday_heute_default-0, more examples are shown in the supplement material) in PHOENIX14 training set and compute the frame-to-frame self-similarity (the cosine-similarity between the features) matrices of the LVFs and GCFs, and the similarity matrix between the LVFs and GCFs. Then the heatmaps of the obtained matrices are visualized and shown in Fig. 3.

In Fig. 3(c), a strong correlation between a GCF of the key frame and the LVFs of the adjacent frames can be observed. That is to say, the GCF of a key frame focuses the LVFs near it. Considering the visual module learns the glosses independently due to less sentence-level supervision, the contextual module attempts to learn the glosses by paying more attention to visual information instead of "remembering" all the signing sequences by using the full-frame information.

Both Fig. 3(a) and Fig. 3(b) show that, either the LVFs or the GCFs have the property of local similarity. Moreover, in Fig. 3(b) the GCFs of some frames have significant differences from other. Meanwhile, we find that all of these frames are predicted as non-blank classes, while others are predicted as the blank class. There are two reasons for this phenomenon:

1) The spike phenomenon, which widely exists during training with CTC loss, still happens when using WS.

2) As the visual and contextual modules have different receptive fields, the spike phenomenon influences the feature learning of the contextual module and visual module at different scales.

The contextual module has a larger receptive field which allows it to aggregate the global context information in one moment. Correspondingly, the visual module only has a local receptive field, which causes only a few key frames to contribute to the result and makes the visual module lose its discriminative power for the other frames.

3.5. Gloss Segment Boundary Assignment

To enhance the utilization of short-term spatial-temporal information, an effective way is to increase the proportion of the non-blank class-relevant features. So a gloss segmentation is added into the visual module, where the pseudo gloss segment label is produced by a proposed GSBA algorithm. Below the proposed GSBA algorithm will be described in detail.

The proposed GSBA algorithm is based on two assumptions: 1) For a given image sequence $\boldsymbol{X}$, each frame $\boldsymbol{x}_t$ respond to a class $c_t \in \boldsymbol{l}$, where $\boldsymbol{l}$ is its corresponding sign gloss sequence; 2) The pseudo label produced by the CTC can be viewed as a single-frame supervision [21] for gloss segmentation. Similar to the CTC, the GSBA will iteratively produce the gloss segmentation proposal during the training of the visual module. Specifically, each key frame is treated as an anchor frame. Given an anchor frame at time t, which corresponding class is c_a. An expanded radius d is set to limit the maximum expansion distance firstly (d increases with iteration). Then, we annotate the past frame from $t-1$ to $t-d$ frame and the future frame from $t+1$ to $t+d$ frame, separately. If the cosine similarity between the GCF of the current expanding frame and the weight vector of c_a is the largest among the classes $c_j \in \boldsymbol{l}$, this frame will be annotated with the label c_a. Otherwise, we will stop the expansion process (the pseudo-code is shown in the supplement material). This method's runtime is almost linear to different glosses in the whole dataset, which is the size of all gloss sequences. So it is fast to produce the gloss segment label. The visualization of pseudo gloss segment label produced by GSBA with $d = 1, 2, 3$ is shown in Fig. 4.

With the produced segment labels $\boldsymbol{Y}^{seg}$, a gloss segmentation is added for the visual module. And a softmax loss

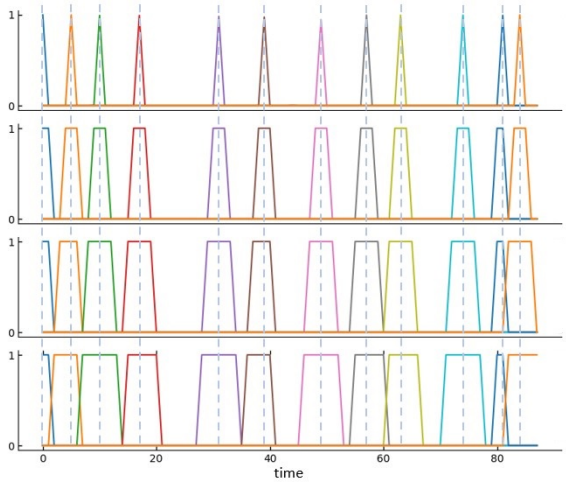


Figure 4. From top to bottom are the spike phenomenon and the pseudo gloss segment labels produced by GSBA with $d = 1, 2, 3$ (different colors represent different classes).

with label smoothing [30] is used for the gloss segmentation. The overall objective $\mathcal{L}$ becomes:

$$\mathcal{L}_{\text{GSBA}} = \mathcal{L}_{\text{CTC}}\left(\boldsymbol{l}, \hat{\boldsymbol{Y}}_g\right) + \alpha \mathcal{L}_{\text{CE-LS}}\left(\tilde{\boldsymbol{Y}}^{seg}, \hat{\boldsymbol{Y}}_v\right), \tag{10}$$

where $\mathcal{L}_{\text{CE-LS}}$ represent $\mathcal{L}_{\text{CE}}$ with label smoothing and $\tilde{\boldsymbol{y}}^{seg}$ denotes the smoothed labels.

Moreover, as the segment labels are produced from the contextual module's prediction, the GSBA can be treated as an extension of hard KD, as it can adjust the spiking ratio. In such a way, a many-to-one alignment between the LVFs and GCFs is built at the temporal level.

3.6. Optimization Approach

As the context information is also critical to the recognition task, the weight matrix sharing between two modules should be decoupled during the final training stage to relax the constraints on the contextual module. Therefore, a three-stage optimization approach, which contains synchronous training stage, gloss segmentation stage and decouple training stage, is proposed for training as shown in Fig. 5. The three-stage optimization approach includes 1) sharing the weight matrix of the visual module and contextual module to enforce them focus more on the short-term temporal information; 2) adding a gloss segmentation task for the visual module to enhance the utilization of the short-term temporal information; 3) decoupling the weight matrix to make the contextual module focuses on long-term temporal information.

4. Experimental Results

In this section, we evaluate the effectiveness of the proposed method on the two CSLR datasets. We first detail the experimental settings (Sect. 4.1). Then we perform ablation studies on Sect. 4.2. Finally, we compare SMKD with other state-of-the-art methods (Sect. 4.3).

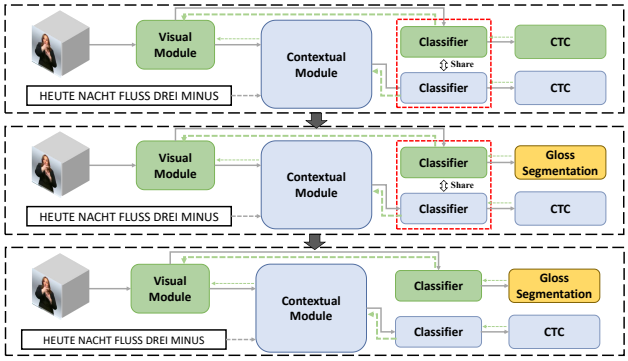


Figure 5. Illustration of the three-stage optimization approach we proposed, which consists of three stages: synchronous training stage, gloss segmentation stage, and decouple training stage.

4.1. Experiment Settings

Dataset. Two public datasets: RWTH-PHOENIX-Weather-2014 (PHOENIX14) and RWTH-PHOENIX-2014-Weather-T (PHOENIX14-T) are selected for this study.

PHOENIX14 [16] is a popular German sign language dataset collected from weather forecast broadcast, There are $6,841$ sentences signed by 9 signers (around $80,000$ glosses with a vocabulary of $1,295$ signs). All videos are of 25 frame per second with the resolution of 210×260. The dataset is divided into three parts: $5,672$ instance for training, 540 for development, and 629 for testing.

PHOENIX14-T [4] can be treated as an extension to the PHOENIX14. It contains parallel sign language, gloss annotations, and translations, making it available to evaluate both SLR and Sign Language Translation (SLT) tasks. The dataset has a vocabulary of $1,085$ signs and is also divided into three parts: $7,096$ instance for training, 519 for development, and 642 for testing.

Evaluation Metric. For video-based CSLR, we use Word Error Rate (WER) as the metric, which is defined as the minimal summation of the substitution (#sub), insertion (#ins), and deletion (#del) operations to convert the recognized sentence to the corresponding reference sentence (#reference):

$$\text{WER} = \frac{\#\text{sub} + \#\text{ins} + \#\text{del}}{\#\text{reference}}. \tag{11}$$

Implementation Details. For both datasets, the frames are resized to 256×256 and then cropped to 224×224. During training, we use random crop and horizontal flip (50%) for data augmentation. During testing, we only adopt center cropping. Let C_k, P_k denote a temporal convolutional layer with k ($= 1024$) filters, and a temporal max pooling layer with stride k, respectively. The visual module is

Table 1. Ablation studies on the effectiveness of the weight normalization (WN) on the PHOENIX14.

Methods	WN	Dev (%)	Test (%)
Baseline		24.3	25.4
Baseline	✓	24.1	24.7
Baseline+WS		21.4	21.9
Baseline+WS	✓	21.2	21.4

composed with a 2D Resnet18 [10] pre-trained on the ImageNet and a $C_5 - P_2 - C_5$ layer as a default setting. Batch normalization [13] is added after each convolutional layer to accelerate training. The contextual module contains two BiLSTM [12] layers with 2×512 dimensional hidden states. After feature extraction stage, the classifier cast the feature channel number to $|\mathbb{G}| + 1$. The model is trained with a batch size of 2 using the Adam optimizer [14] with an initial learning rate $\eta = 10^{-4}$. Each model is trained for 100 epochs and halves the learning rate at 40, 60, and 80. The gloss segmentation task is activated after epoch 30, and the gloss segmentation proposals are updated every 10 epoch. The initial d is set to 1, and then plus one every 20 epoch. The label smoothing rate is set as 0.2. For the decouple training stage, we train the network for 10 epochs and use the Adam optimizer with learning rate $\eta = 4 \times 10^{-6}$.

4.2. Ablation studies

In this section, ablation studies are conducted to demonstrate the effectiveness of SMKD. The network only with the CTC loss in contextual module is selected as the baseline. For a fair comparison, experiments are all performed on the PHOENIX14.

Effects of the weight normalization. As shown in Table 1, adding the weight normalization during the recognition can improve the performance in both baseline and baseline add WS, especially on the test set. As mentioned in [20], the norm can be treated as a kind of prior. Therefore, the normalization of weight $\boldsymbol{w}_i$ plays an important role in dealing with an unbalanced dataset, such as PHOENIX14. We adopt the weight normalization as the default setting for the following experiments.

Different way to construct the weight matrix. To study the impact of α in Equation (9) we test different α values and the result is shown in Fig. 6. The result shows that, as the α increases, the performance will improve first and then decline. We suppose that to achieve better performance, the visual module and contextual module need to reach a certain balance. Among the selected α, the optimal α is 0.5 for the Dev set and the Test set. In addition, we also test different ways to construct the general weight matrix, i.e., only use LVFs, GCFs (freeze $\boldsymbol{W}$ when backpropagating the gradient computed by the GCFs or LVFs) and use both of them. The results are shown in Table 2. The performance of only LVFs is superior to only GCFs, while using both of them achieves

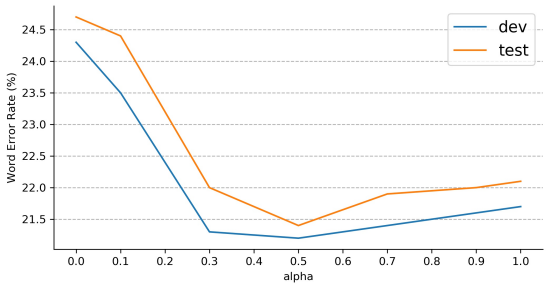


Figure 6. Ablation studies on the different α values on the PHOENIX14.

Table 2. Ablation studies on the different way to construct weight matrix on the PHOENIX14.

	Dev (%)	Test (%)
only GCFs	22.4	22.9
only LVFs	22.1	22.5
GCFs & LVFs	**21.2**	**21.4**

Table 3. Ablation studies on the different KD methods on the PHOENIX14.

Methods	Dev (%)	Test (%)
Baseline	24.1	24.7
Baseline+KD	23.2	23.6
Baseline+WS	**21.2**	**21.4**
Baseline+GSBA	21.5	22.0

relatively better performance. Unless stated, we select $\alpha = 0.5$ as the default setting in the next experiments.

Comparison with general KD. To show the effectiveness of the our proposed KD methods: WS and GSBA (extension of hard KD), we compare them with general KD method [11], i.e., using the soft probability distribution produced from the contextual module to guide the training of the visual module. Then, the overall objective $\mathcal{L}$ becomes:

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{CTC}}\left(\boldsymbol{l}, \hat{\boldsymbol{Y}}_g\right) + \mathcal{L}_{\text{CE}}\left(\hat{\boldsymbol{Y}}_g/\tau, \hat{\boldsymbol{Y}}_v/\tau\right), \tag{12}$$

where $\tau = 8$ denotes the temperature. The results are shown in Table 3, where both SMKD and GSBA perform better than KD. As discussed in Sect. 3.4, the spike phenomenon has a different influence on the visual and contextual modules, the soft target produced from the contextual module is composed of a lot of spiking and will mislead the visual module's training.

Comparison with different optimization approaches To clarify the effectiveness of our proposed optimization approach, we evaluate the performance in different stages in Table 4. Note that, baseline+dec_train means the visual and contextual modules have independent classifiers during total training process. We can observe that the performance of dec_train is inferior to the sync_train. This suggests that with the WS constraint, the contextual module can better utilize the visual information. The base-

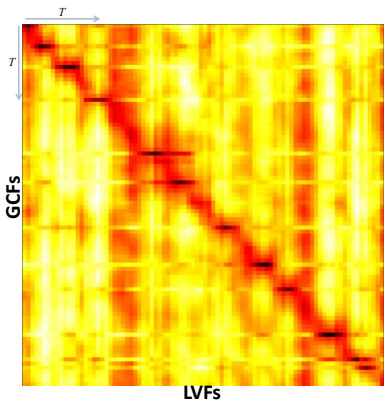


Figure 7. Similarity matrix between the LVFs and GCFs after the gloss segmentation stage.

line+sync_train+dec_train means the visual module will not perform the gloss segmentation task, the performance is not affected too much as the network has tended to be stable. Using the gloss segmentation task will introduce more visual information to improve the performance. Moreover, after the dec_train, the contextual module can select the useful visual information and then further promote the performance to achieve the best WER (Dev: 20.8, Test: 21.0). As shown in Fig. 7, we also visualize the similarity matrix between the LVFs and GCFs after performing the gloss segmentation stage. With the gloss segmentation, the GCF of a key frame will focus on more LVFs nearby it.

Table 4. Ablation studies on the optimization approaches on the PHOENIX14 (sync_train: synchronous training stage, gloss_segment: gloss segmentation stage, dec_train: decouple training stage).

baseline	sync_ train	gloss_ segment	dec_ train	Dev (%)	Test (%)
✓				24.1	24.7
✓			✓	22.0	22.4
✓	✓			21.2	21.4
✓	✓		✓	21.0	21.3
✓	✓	✓		20.9	21.3
✓	✓	✓	✓	**20.8**	**21.0**

4.3. Comparison with State-of-the-arts

In this section, we present thorough comparison with other state-of-the-art (SOTA) methods on the two datasets mentioned in Sect. 4.1.

Evaluation on PHOENIX14. Table 5 shows our approach compared to other methods on the PHOENIX14. The WER of our SMKD method on the development set and testing set are 20.8% and 21.0%, respectively, and achieves SOTA performance for RGB-based methods. Furthermore, while we do not use any extra clues, our model achieves comparable results among the models trained with extra clues.

Table 5. Performance comparison (%) on PHOENIX14 (symbol * represents using the extra clues), 'del' and 'ins' stand for deletion error and insertion error, respectively.

Methods	Dev (%)		Test (%)	
	del/ins	WER	del/ins	WER
SubUNet [1]	14.6/4.0	40.8	14.3/4.0	40.7
Staged-Opt [5]	13.7/7.3	39.4	12.2/7.5	38.7
Align-iOpt [26]	12.6/2.6	37.1	13.0/2.5	36.7
DPD+TEM [34]	9.5/3.2	35.6	9.3/3.1	34.5
Re-Sign [18]	-	27.1	-	26.8
SFL [23]	7.9/6.5	26.2	7.5/6.3	26.8
DNF [6]	7.8/3.5	23.8	7.8/3.4	24.4
FCN [3]	-	23.7	-	23.9
VAC [22]	7.9/2.5	21.2	8.4/2.6	22.3
CMA [25]	7.3/2.7	21.3	7.3/2.4	21.9
SFL [23]	10.3/4.1	24.9	10.4/3.6	25.3
DNF [6]*	7.3/3.3	23.1	6.7/3.3	22.9
STMC [35]*	7.7/3.4	21.1	7.4/2.6	**20.7**
SMKD (ours)	6.8/2.5	**20.8**	6.3/2.3	**21.0**

Table 6. Performance comparison (%) on PHOENIX14-T (v: video, m:mouth, h:hand, t:text, f:face, p:pose).

Methods	WER	
	Dev (%)	Test (%)
SFL (v) [23]	25.1	26.1
CNN+LSTM+HMM (v) [15]	24.5	26.5
SLT (v) [2]	24.9	24.6
FCN (v) [3]	23.3	25.1
CNN+LSTM+HMM (v+m) [15]	24.5	25.4
CNN+LSTM+HMM (v+m+h) [15]	22.1	24.1
SLT (v+t) [2]	24.6	24.5
STMC (v+h+f+p) [35]	**19.6**	**21.0**
SMKD (v)	**20.8**	**22.4**

Evaluation on PHOENIX14-T. In Table 6, we evaluate our approach on the PHOENIX14-T. We can observe that our approach also achieves the best performance (Dev: 20.8% and Test: 22.4%) with video information only.

5. Conclusion

In this paper, we propose a SMKD method to enforce the visual and contextual modules optimizing simultaneously in the initial stage of training to avoid the drawback of back-propagation, and decoupling the two modules in the late stage. To deal with spike phenomenon caused by CTC constraint, and utilize more visual information, we propose to add an additional gloss segmentation to enhance the visual module. To train the SMKD, we propose a three-stage optimization approach. Experimental results show that the proposed method achieve state-of-the-art performance on two benchmark datasets. Sign language recognition is a typical spatial-temporal sequence problem, and the proposed SMKD can further extend to other similar tasks.

References

[1] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3075–3084, 2017. 8

[2] 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 Conference on Computer Vision and Pattern Recognition*, pages 10023–10033, 2020. 2, 8

[3] Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai. Fully convolutional networks for continuous sign language recognition. In *Proceedings of the European Conference on Computer Vision*, pages 697–714. Springer, 2020. 1, 2, 4, 8

[4] 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. 6

[5] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7361–7369, 2017. 8

[6] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia*, 21(7):1880–1891, 2019. 1, 2, 8

[7] Jiankang Deng, Jia Guo, Debing Zhang, Yafeng Deng, Xiangju Lu, and Song Shi. Lightweight face recognition challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[8] Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer, 2012. 2

[9] 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. 1

[10] 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. 7

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 7

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 7

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 7

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[15] Oscar Koller, Cihan Camgoz, Hermann Ney, and Richard Bowden. Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 8

[16] 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. 6

[17] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. Deep sign: hybrid cnn-hmm for continuous sign language recognition. In *Proceedings of the British Machine Vision Conference*, 2016. 2

[18] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4297–4305, 2017. 2, 8

[19] Hongzhu Li and Weiqiang Wang. Reinterpreting ctc training as iterative fitting. *Pattern Recognition*, 105:107392, 2020. 2, 4

[20] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017. 3

[21] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *Proceedings of the European Conference on Computer Vision*, pages 420–437. Springer, 2020. 5

[22] Yuecong Min, Aiming Hao, Xiujuan Chai, and Xilin Chen. Visual alignment constraint for continuous sign language recognition. *arXiv preprint arXiv:2104.02330*, 2021. 2, 8

[23] Zhe Niu and Brian Mak. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *Proceedings of the European Conference on Computer Vision*, pages 172–186, 2020. 2, 8

[24] Sylvie CW Ong and Surendra Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):873–891, 2005. 1

[25] Junfu Pu, Wengang Zhou, Hezhen Hu, and Houqiang Li. Boosting continuous sign language recognition via cross modality augmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1497–1505, 2020. 8

[26] Junfu Pu, Wengang Zhou, and Houqiang Li. Iterative alignment network for continuous sign language recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4174, 2019. 1, 2, 8

[27] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 2, 4

[28] Weidong Shi, Guanghui Ren, Yunpeng Chen, and Shuicheng Yan. Proxylesskd: Direct knowledge distillation with inherited classifier for face recognition. *arXiv preprint arXiv:2011.00265*, 2020. 3

[29] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. 4

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[31] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020. 3

[32] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3713–3722, 2019. 3

[33] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 3

[34] Hao Zhou, Wengang Zhou, and Houqiang Li. Dynamic pseudo label decoding for continuous sign language recognition. In *Proceedings of the IEEE International Conference on Multimedia andFull Expo*, pages 1282–1287, 2019. 2, 8

[35] Hao Zhou, Wengang Zhou, Yun Zhou, and Houqiang Li. Spatial-temporal multi-cue network for continuous sign language recognition. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 13009–13016, 2020. 2, 8