Visual Alignment Constraint for Continuous Sign Language Recognition

Yuecong Min[1,2], Aiming Hao[1,2], Xiujuan Chai[3], 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
[3]Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing, 100081, China
{yuecong.min,aiming.hao}@vipl.ict.ac.cn, chaixiujuan@caas.cn, xlchen@ict.ac.cn

Abstract

Vision-based Continuous Sign Language Recognition (CSLR) aims to recognize unsegmented signs from image streams. Overfitting is one of the most critical problems in CSLR training, and previous works show that the iterative training scheme can partially solve this problem while also costing more training time. In this study, we revisit the iterative training scheme in recent CSLR works and realize that sufficient training of the feature extractor is critical to solving the overfitting problem. Therefore, we propose a Visual Alignment Constraint (VAC) to enhance the feature extractor with alignment supervision. Specifically, the proposed VAC comprises two auxiliary losses: one focuses on visual features only, and the other enforces prediction alignment between the feature extractor and the alignment module. Moreover, we propose two metrics to reflect overfitting by measuring the prediction inconsistency between the feature extractor and the alignment module. Experimental results on two challenging CSLR datasets show that the proposed VAC makes CSLR networks end-to-end trainable and achieves competitive performance.

1. Introduction

Sign Language is a complete and natural language that conveys information through both manual components (hand/arm gestures) and non-manual components (facial expressions, head movements, and body postures) [10, 37] with its own grammar and lexicon [41]. Vision-based Continuous Sign Language Recognition (CSLR) aims to automatically recognize signs from image streams, which can bridge the communication gap between the Deaf and hearing people. It also provides more non-intrusive communication channel for sign language users.

Different from speech recognition, the data collection and annotation of sign language are costly, which poses a significant problem for recognition [2]. Therefore, most recent CSLR works solve this problem in a weakly supervised manner and adopt network architectures composed of the feature extractor and the alignment module. The feature extractor abstracts visual information from each frame, and the alignment module searches the possible alignments between visual features and the corresponding labeling. Different to those works [27, 29, 31] adopt HMMs to update frame-wise state labels for the feature extractor, Graves *et al*. [15] provide a more elegant solution so-called Connectionist Temporal Classification (CTC) to align the prediction and labeling by maximizing the sum of probability of all feasible alignments, which is adopted by following works [3, 6, 8, 9, 27, 36, 46].

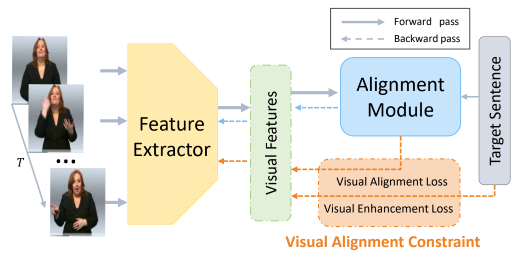


Figure 1. Overview of the proposed non-iterative CLSR approach with the visual alignment constraint. To solve the insufficient training of the feature extractor, the proposed VAC enhances the generalization ability of the visual extractor by constraining the feature space with the alignment supervision.

Although CTC-based CSLR methods provide convenience in training, previous studies [9, 39] show that end-to-end training limits the discriminative power of the feature extractor. They leverage the iterative training scheme to enhance the feature extractor, which significantly improves the performance. Nevertheless, it requires an additional fine-tuning process besides the end-to-end training and increases the training time. Several recent works [6, 36] try to accelerate this training scheme by adopting fully convolutional networks and fine-grained labels.

In this study, we revisit CTC-based CSLR model at different iterations and observe that only a few frames play key roles in training. The feature extractor abstracts visual information and provides initial localizations of key frames for the alignment module. The alignment module further refines the recognition results from the feature extractor and learns long-term relationships with its powerful temporal modeling ability. Due to the spike phenomenon of CTC [14, 34], the alignment module converges much faster than the feature extractor on CSLR datasets with limited samples and cannot provide enough feedback to the feature extractor. The overfitting of the alignment module leads to insufficient training of the feature extractor and deteriorates the generalization ability of the trained model. The iterative training scheme tries to solve this problem by enhancing the feature extractor with iteratively refined pseudo labels.

Based on above observations, we conclude that constraining the feature space is critical to efficiently train CSLR models. To solve this problem, we propose a Visual Alignment Constraint (VAC) to make CSLR networks end-to-end trainable. As shown in Fig. 1, the proposed VAC is composed of two auxiliary losses which provide extra supervision for the feature extractor. The visual enhancement loss enforces the feature extractor to make predictions based on visual features only and the visual alignment loss aligns the short-term visual predictions to long-term contextual predictions. With the combination of the two losses, the proposed method achieves competitive performance to the latest methods on PHOENIX14 [28] and CSL [23] datasets.

To better understand the performance gains, we present two metrics named Word Deterioration Rate (WDR) and Word Amelioration Rate (WAR) to evaluate the contributions of the feature extractor and the alignment module, which can also be used as indicators of overfitting. Comparing to the iterative training procedure, experimental results show that the proposed method can obtain a more powerful feature extractor and make better use of visual features.

The major contributions are summarized as follows:

- Revisiting the iterative training scheme in CSLR and showing that the overfitting of the alignment module leads to insufficient training of the feature extractor.
- Proposing a visual alignment constraint to make the network end-to-end trainable by enhancing the feature extractor and aligning visual and contextual features.
- Presenting two metrics to evaluate the contributions of the feature extractor and the alignment module, which verifies the effectiveness of the proposed method.

2. Related Work

2.1. Continuous Sign Language Recognition

Sign Language Recognition (SLR) methods can be roughly categorized into isolated SLR [25, 32, 33] and continuous SLR [9, 27]. Different to isolated SLR, most CSLR approaches model sequence recognition in a weakly supervised manner: only sentence-level labeling is provided. Some early CSLR methods [12, 18, 37] adopt a divide-and-conquer paradigm that splits sign video into several subunits with HMM-based recognition systems to work with limited data. Hand-crafted features [11, 28, 43] are carefully selected to provide better visual information.

The recent successes of CNNs in computer vision [20, 42, 44] provide powerful tools for visual features representation. However, CNNs need frame-wise annotations contrary to the weakly supervised nature of CSLR. To solve this problem, Koller *et al.* [29] propose an iterative expectation-maximization approach by adding a hand shape classifier to the GMM-HMM model as an intermediate task to provide frame-level supervision. A few studies extend this work by proposing CNN+LSTM+HMM framework [30], incorporating more clues [27] and improving the iterative alignment approach [31]. This iterative CNN-LSTM-HMM setup provides robust visual features that are adopted by many recent works [4, 7].

Although the CNN-LSTM-HMM hybrid approaches achieve great results, they still need HMMs to provide frame-wise labels. Graves *et al.* [15] propose the CTC loss to maximize probabilities of all feasible alignments, which is widely used in many sequence problems [17, 16]. Several recent works [3, 8] use CTC loss to achieve the end-to-end training of CSLR. However, some works [8, 9, 39] find that such an end-to-end approach cannot train feature extractor properly and bring the iterative training back in use. Until very recently, some works [6, 36] try to solve this problem in an end-to-end way. Cheng *et al.* [6] propose a gloss feature enhancement module to learn better visual features. Niu and Mak [36] propose a multiple states approach and several operations to alleviate the overfitting problem. In this work, we try to explore the nature of iterative training and propose a more efficient method to train CSLR models.

2.2. Auxiliary Learning

Different from the conventional Multi-Task Learning [5], which aims to improve the generalization of all tasks, auxiliary learning chooses proper auxiliary tasks to assist in the generalization of the primary task. One straightforward way is to combine multiple tasks at the output stage. Follow this idea, Kim *et al.* [26] use CTC to speed up the training process and provide a monotonic alignment constraint. Pu *et al.* [39] propose an iteratively alignment network that jointly optimizes the CTC decoder and the LSTM decoder, additionally with a soft-DTW alignment constraint. Goyal *et al.* [13] propose an auxiliary loss to alleviate the posterior collapsing phenomenon in autoregressive decoder [1]. Another idea is to use different supervision at different stages. Sanabria *et al.* [40] use several

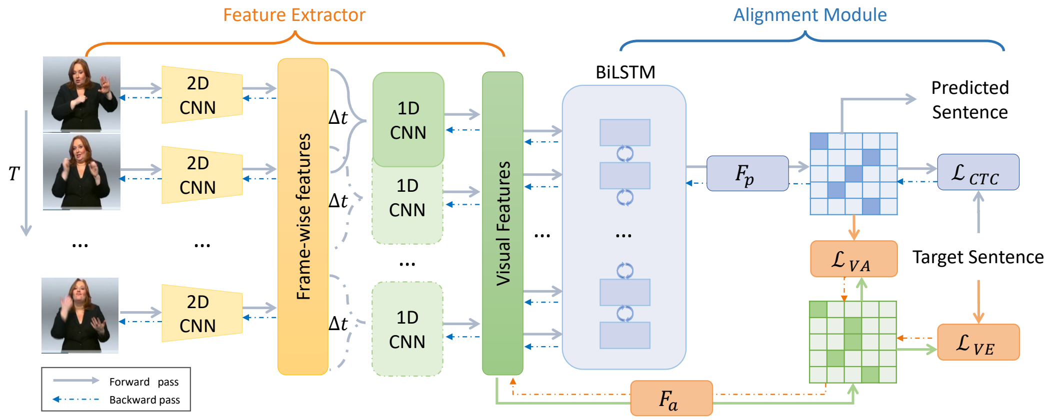


Figure 2. The proposed framework consists of three components: a feature extractor, an alignment module, and an auxiliary classifier F_a. The feature extractor first takes image sequence to abstract frame-wise features, and then applies 1D-CNN to extract the local visual information with Δt temporal receptive field. The outputs of 1D-CNN noted as visual features are sent to the alignment model and the auxiliary classifier. Two auxiliary losses are adopted during training: the visual enhancement loss ($\mathcal{L}_{VE}$) aligns visual features and the target sequence, and the visual alignment loss ($\mathcal{L}_{VA}$) aligns short-term visual predictions and long-term context predictions through knowledge distillation.

lower-level tasks, such as phoneme recognition, to constrain intermediate representations for speech recognition. In this study, we adopt the auxiliary learning strategy to provide the visual alignment constraint for the feature extractor.

3. Revisiting the Iterative Training in CSLR

The CSLR aims to predict the corresponding gloss label sequence $\boldsymbol{l} = (l_1, \cdots, l_N)$ based on a sequence of T frames $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$. The feature extractor plays an important role in CSLR, which extracts visual features $\boldsymbol{V} = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{T'})$ from image sequences. As shown in Fig. 2, we choose 2D-CNN to extract frame-wise features and 1D-CNN to extract local posture and motion information from neighboring frames as previous works did [9, 48]. The gloss-wise features are fed into a two-layer BiLSTM and the primary classifier F_p to combine long-term relationships and provide the predicted logits $\boldsymbol{Z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{T'})$. CTC loss is adopted to provide supervision by aligning the predictions and sequence labelings.

3.1. The Spike Phenomenon of CTC

The Connectionist Temporal Classification [15] is designed for end-to-end temporal classification tasks with unsegmented data. To provide more effective supervision, CTC introduces a 'blank' to represent unlabeled data (such as movement epenthesis or non-gesture segments in CSLR) and solves the alignment problem with dynamic programming. The blank class and gloss vocabulary $\mathbb{G}$ build the final extended gloss vocabulary $\mathbb{G}' = \mathbb{G} \cup \{blank\}$.

CTC defines a many-to-one function $\mathcal{B} : \mathbb{G}'^T \rightarrow \mathbb{G}^{\leq T}$ to align label sequence referred to as path $\pi \in \mathbb{G}'^T$ and labeling $\boldsymbol{l} \in \mathbb{G}^{\leq T}$ by sequentially removing the repeated labels and the blanks from the path. For example, $\mathcal{B}(\text{-}aaa\text{--}aabbb\text{-}) = \mathcal{B}(\text{-}a\text{-}ab\text{-}) = aab$. With the help of this function, CTC can provide supervision for parameters θ of the feature extractor and the alignment module by summing the probabilities of all feasible paths:

$$\begin{aligned}\mathcal{L}_{CTC} &= -\log p(\boldsymbol{l}|\boldsymbol{X};\theta) \\ &= -\log\Big(\sum_{\pi\in\mathcal{B}^{-1}(\boldsymbol{l})} p(\pi|\boldsymbol{X};\theta)\Big).\end{aligned} \tag{1}$$

The conditional probability $p(\pi|X)$ can be calculated according to the conditional independence assumption:

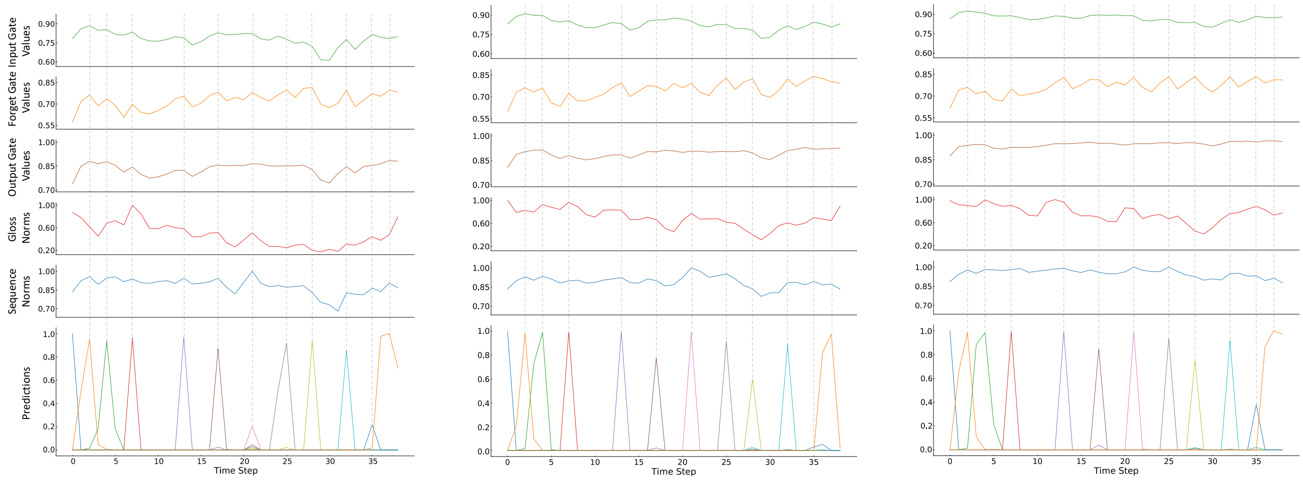
$$p(\pi|\boldsymbol{X}) = \prod_{t=1}^{T'} p(\pi_t|\boldsymbol{X};\theta), \tag{2}$$

where the probabilities are calculated by applying softmax fuction to the the network output logits: $P_\theta = \text{softmax}(\boldsymbol{Z})$.

As mentioned above, CTC aligns the path and the labeling by introducing a blank class and removing the repeat labels. When optimizing network with CTC, predictions tend to form a series of spike responses [15, 34]. The main reason for this is that predicting a blank label is a much safer choice for CTC when the network cannot confidently distinguish gloss boundaries. For example, both $\mathcal{B}(aaab)$ and $\mathcal{B}(a\text{--}b)$ are corresponding to the same labeling, but $\mathcal{B}(a\boldsymbol{b}ab)$ will bring larger loss even if there is only one mistake. Therefore, the CTC loss mainly focuses on key frames, and the final predictions are composed of a few non-blank key frames and many high-confidence blank frames.

3.2. Visualization of LSTM Gates

Long Short-Term Memory [22] is widely used in sequence modeling, which excellently models long-term dependencies. The core component of LSTM is its memory



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure 3. Visualization of the gate values, the l_2 norm of features and the final prediction of a training sample among different iterations.

design: the input and forget gates control information from current inputs and the past memory to the current memory. The output gate controls what is expected to output from the current memory. The total update mechanism is as follows ($\odot$ denotes the Hadamard product):

$$\begin{aligned} \boldsymbol{i}_t &= \sigma(\boldsymbol{U}_i \boldsymbol{v}_t + \boldsymbol{W}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i), \\ \boldsymbol{f}_t &= \sigma(\boldsymbol{U}_f \boldsymbol{v}_t + \boldsymbol{W}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f), \\ \boldsymbol{o}_t &= \sigma(\boldsymbol{U}_o \boldsymbol{v}_t + \boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o), \\ \tilde{\boldsymbol{c}}_t &= \sigma(\boldsymbol{U}_c \boldsymbol{v}_t + \boldsymbol{W}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c), \\ \boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t, \\ \boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t). \end{aligned} \tag{3}$$

Here the i_t, f_t and o_t are corresponding to input, forget and output gates, respectively, the vector h_t and c_t are hidden and cell states. where $\boldsymbol{U}.$ and $\boldsymbol{W}.$ are the input-to-hidden and hidden-to-hidden weight matrices, and $\boldsymbol{b}.$ are bias vectors. Element-wise sigmoid is reprensented by σ.

Previous works [8, 9, 39] adopt iterative training to enhance the visual extractor. To explore how iterative training works and how LSTM makes predictions in CSLR, we begin by visualizing the averaged gate values of the last forward-direction LSTM and the network predictions at different iterations in Fig. 3. For the predictions, we only visualize non-blank classes that occur in the labeling. We can make some observations from the comparison of line charts:

1) The gate values and the predictions have positive correlations on the training set, and they reach the local maximum on similar frame subsets.

2) The correlations appear to be weakened as the iteration progresses, especially for the input and output gates, which become larger and smoother.

The above two observations are quite puzzling, as three gates are expected to play different roles in information flow. As shown in Equ. 3, three gates take the same inputs and have independent parameters. Therefore, we pinpoint the problem to the magnitude of input features and further visualize the l_2 norms of the activations before the first and the second BiLSTM layers, which are referred to as the gloss and sequence norms in Fig. 3.

3.3. A Magnitude Hypothesis

Fig. 3 presents an interesting observation that the l_2 norms of gloss and sequence features have similar tendencies with gates values and final predictions. Besides, the magnitudes variances of both gloss and sequence become smaller as the iteration progresses. Several recent papers [35, 45] found that well-separated features tend to have larger magnitudes, and we hypothesize the magnitudes variances are relevant to the importance of frames:

The l_2 norms of the features are effect indicators that reflect frame importance: the optimization algorithm will decrease the magnitudes of activations when suppressing the non-key frames due to the spike phenomenon of CTC.

With the above hypothesis, it is clear that frames with larger magnitudes in Fig. 3 play key roles compared to their neighbors. We further interpret the learning process of CTC-based CSLR model into two stages: 1) the feature extractor provides visual and initial localization information for the alignment module, and 2) the BiLSTM layers refine the localization and learn long-term relationships among key frames. Such a learning scheme can make efficient use of the data and accelerate the training process.

However, current CSLR datasets contain less data than other sequence learning tasks [17, 19], which means the BiLSTM layers can easily overfit the whole training set with partial visual information and other frames are decreasingly involved in the training progress. Although the network can achieve stable convergence, the power of feature extractor is not sufficiently explored. Therefore, the feature extractor cannot provide robust visual features during inference and deteriorate the generalization performance.

Based on these analyses, we attribute the success of iterative training to the reduction of the overfitting problem. With pseudo labels generated by the alignment module, the fine-tuning stage can enhance the feature extractor to make it generalize better. Although the pseudo labels can relieve the overfitting problem in some sense, it is still not enough. Therefore, we propose the visual alignment constraint on the visual feature space, which enforces the feature extractor to make predictions on its own and adopts the distillation loss to align both visual and contextual spike responses.

4. Visual Alignment Constraint

As mentioned above, the BiLSTM layers can easily overfit the training set with partial visual information. In this paper, we propose the Visual Alignment Constraint (VAC) to enhance the feature extractor with more alignment supervision. The proposed VAC is implemented by two simple auxiliary losses: the Visual Enhancement (VE) loss and the Visual Alignment (VA) loss. Besides, we propose two new evaluation metrics, Word Deterioration Rate (WDR) and Word Amelioration Rate (WAR), to evaluate the contributions of the feature extractor and the alignment module.

4.1. Loss Design of VAC

VE Loss. To enhance the feature extractor, we proposed to add an auxiliary classifier F_a on visual features $\boldsymbol{V}$ to get the auxiliary logits $\tilde{\boldsymbol{Z}} = (\tilde{z}_1, \cdots, \tilde{z}'_T) = F_a(\boldsymbol{V})$ and propose the VE loss that directly provides supervision for the feature extractor. This auxiliary loss enforces the feature extractor to make predictions based on local visual information only. Compared to previous gloss-wise supervision that needs to generate pseudo labels, we propose to add a CTC loss on the auxiliary classifier as the VE loss, which is compatible with the primary CTC loss and flexible to network designs. The VE loss only provides supervision for parameters θ^v of the feature extractor and the auxiliary classifier:

$$\mathcal{L}_{VE} = \mathcal{L}^v_{CTC} = -\log p(\boldsymbol{l}|\boldsymbol{X};\theta^v). \tag{4}$$

VA Loss. Because the VE loss lacks contextual information and is independent of the primary loss, which may lead to misalignment between two classifiers, we further propose the VA loss. The VA loss is implemented as a knowledge distillation loss [21], which regards the entire network and the visual feature extractor as the teacher and student models, respectively. A high temperature τ is adopted to "soften" probability distribution from spike responses. The distillation process is formulated as:

$$\mathcal{L}_{VA} = \mathrm{KL}\Big(\mathrm{softmax}(\frac{\boldsymbol{Z}}{\tau}), \mathrm{softmax}(\frac{\tilde{\boldsymbol{Z}}}{\tau})\Big). \tag{5}$$

In summary, to achieve the visual alignment goal, the VE loss enforces the feature extractor to provide more robust visual features for the alignment module, while the VA loss aligns the predictions of two classifiers by providing long-term supervision for the visual extractor. With the help of both losses, the feature extractor obtains more supervision which is compatible with the alignment module. The final objective function is composed of the primary CTC loss, the visual enhancement loss, and the visual alignment loss:

$$\mathcal{L} = \mathcal{L}_{CTC} + \mathcal{L}_{VE} + \alpha\mathcal{L}_{VA}. \tag{6}$$

4.2. Prediction Inconsistency Measurement

Word Error Rate (WER) is a widely-used metric to evaluate the performance of recognition algorithms in CSLR [28]. It is also referred to as the length normalized edit distance, which first aligns the recognized sequence with the reference sentence and then counts the number of operations, including substitution (sub), deletion (del), and insertion (ins), to transfer from the reference to the recognized sequence: WER = (#sub + #del + #ins) / #reference.

As shown in Fig. 4, both of the auxiliary and the primary recognized sentences (HYP$_a$ and HYP$_p$) have the same WER 22.22% (HYP$_a$ has two deletion errors, and HYP$_p$ has two insertion errors). The primary classifier corrects the misrecognized results of the auxiliary classifier but makes new mistakes, which can not be measured by WER. Therefore, we firstly align sentence triplet (REF*, HYP$^*_a$, HYP$^*_p$) and then calculate WDR and WAR: WDR measures the ratio that is correctly recognized by the auxiliary classifier but misrecognized by the primary classifier (two 'SUED' in HYP$^*_p$), and WAR does in the opposite direction ('MEHR' and 'KALT' in HYP$^*_p$). With the proposed metrics, we can connect the WER*[1] performance of two classifiers by:

$$\mathrm{WER}^*_p = \mathrm{WER}^*_a + \mathrm{WDR} - \mathrm{WAR}. \tag{7}$$

$\mathrm{WER}_p = \mathrm{WER}_a = 2/9 \approx 22.2\%$

REF$_p$:	__ON__	HEUTE	NACHT	MEHR	SCHNEE	NORD	****	SUEDOST	****	ABER	KALT
HYP$_p$:	__ON__	HEUTE	NACHT	MEHR	SCHNEE	NORD	SUED	SUEDOST	SUED	ABER	KALT
REF$_a$:	__ON__	HEUTE	NACHT	MEHR	SCHNEE	NORD		SUEDOST		ABER	KALT
HYP$_a$:	__ON__	HEUTE	NACHT	****	SCHNEE	NORD		SUEDOST		ABER	****
REF*:	__ON__	HEUTE	NACHT	MEHR	SCHNEE	NORD	****	SUEDOST	****	ABER	KALT
HYP$^*_a$:	__ON__	HEUTE	NACHT	****	SCHNEE	NORD	****	SUEDOST	****	ABER	****
HYP$^*_p$:	__ON__	HEUTE	NACHT	MEHR	SCHNEE	NORD	SUED	SUEDOST	SUED	ABER	KALT

$\mathrm{WAR} = 2/9 \approx 22.2\%$ $\quad$ $\mathrm{WDR} = 2/9 \approx 22.2\%$

Figure 4. Alignment results of the proposed metrics. We highlight wrong recognized glosses and the alignment results of the auxiliary classifier and the primary classifier.

In Equ. 7, the final result WER$^*_p$ come from three aspects: how well the visual extractor performs (related to WER$^*_a$), how much visual information is not fully utilized

[1]The adopted alignment approach leads to a little performance degradation than the general WER.

Table 1. Ablation results (WER, %) of iterative training and BN.

Iterations	w/o BN		w/ BN	
	Dev	Test	Dev	Test
1	32.7	33.0	27.2	28.0
2	28.9	29.8	25.5	26.3
3	28.3	28.9	**24.7**	**26.2**
None	30.4	32.1	25.4	26.6

(related to WDR) and how many predictions are made by contextual information only (related to WAR). More details are given in the supplementary material.

5. Experiments

5.1. Experimental Setup

Datasets. We evaluate the proposed method on two widely used datasets: RWTH-PHOENIX-Weather-2014 (PHOENIX14) [28] and Chinese Sign Language (CSL) dataset [23]. All ablations are performed on PHOENIX14.

The PHOENIX14 dataset is a widely used CSLR dataset recorded from the German TV weather forecasts and performed by nine hearing SL interpreters. It contains 6841 sentences with 1295 different glosses. The dataset is split into 5672 training sentences, 540 development (Dev) sentences, and 629 test sentences for the multi-signer setup.

The CSL dataset is collected under laboratory conditions with 100 sign language sentences with a vocabulary size of 178. Fifty signers perform each sentence five times (in 25000 videos with 100+ hours). We follow the previous setting [6] and split the dataset into training and test sets according to the ratio of 8:2.

Implementation Details. ResNet18 [20] is picked as the frame-wise feature extraction in considering its efficiency on the PHOENIX14 dataset. For the CSL dataset, we adopt VGG11 [42] as the backbone to reduce side effects of inconsistent statistics under the signer-independent setting. The gloss-wise temporal layer and two BiLSTM layers with 2×512 dimensional hidden states are adopted as the default setting. The weight α for $\mathcal{L}_{VA}$ is set to 25 and its temperature τ is set to 8 by default. We train all the models for 80 epochs for PHOENIX14 and 20 epochs for CSL with a mini-batch size of 2. Adam optimizer is used with an initial learning rate of 10^{-4}, divided by five after 40 and 60 epochs for PHOENIX14 and 10 and 15 for CSL. For iterative training, we reduce the learning rate by a factor of five after each iteration. All frames are resized to 256x256, and the training set is augmented with random crop (224x224), horizontal flip (50%), and random temporal scaling ($\pm$20%).

5.2. Quantitative Results

Ablation on iterative training and BN. Batch Normalization (BN) [24] is a widely-used tool to accelerate the training of deep networks by normalizing the activations. Although we adopt a small batch size, BN significantly improves the performance. As shown in Table 1, adding a BN layer after each temporal convolution layer brings 5.5%, 3.4%, and 3.6% performance gains at each iteration on the Dev set, which indicates the existence of insufficient training of the feature extractor. We can also observe that adopting iterative training can lead to noticeable performance gains compared to non-iterative training.

Table 2. Ablation results (WER, %) of Learning Rate (LR) ratios (LR of the feature extractor / LR of the alignment model).

LR Ratio	0.1	0.5	1	2	10
Dev	**25.0**	25.6	25.4	26.9	34.8
Test	**25.6**	26.5	26.6	27.5	35.1

Table 3. Ablation results (WER,%) of VAC design.

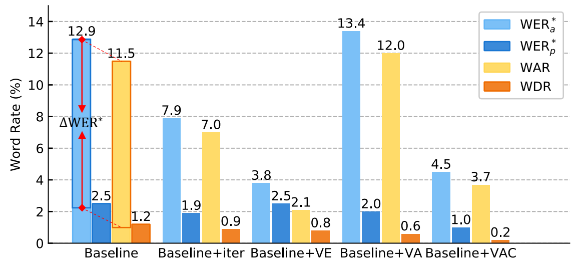
	$\mathcal{L}_{CTC}$	$\mathcal{L}_{VE}$	$\mathcal{L}_{VA}$	Dev	Test
Baseline	✓			25.4	26.6
Baseline+VE	✓	✓		23.3	23.8
Baseline+VA	✓		✓	24.5	25.1
Baseline+VAC	✓	✓	✓	**21.2**	**22.3**

Ablation on learning pace. A natural idea to solve the insufficient training problem is adjusting the learning paces of the feature extractor and the alignment module. In Table 2, we compare results under different learning rate ratios. Adopting a smaller learning rate for the feature extractor leads to comparable results with iterative training, which suggests the existence of insufficient training. However, it is hard to find an optimal learning setting. We adopt a non-iterative model with BN layers and the normal 1:1 learning rate ratio as our baseline.

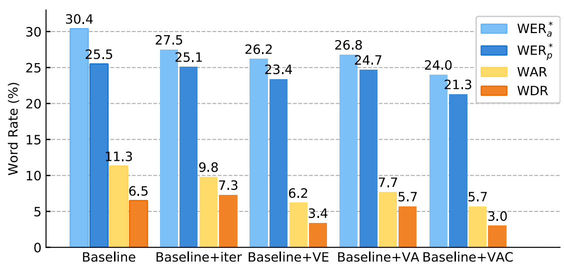
Ablation on VAC. Ablations on VAC are presented in Table 3. Constraining visual features with $\mathcal{L}_{VE}$ and $\mathcal{L}_{VA}$ improves the recognition results (2.1% and 0.9% on Dev set), which verifies the need to strengthen supervision on the feature extractor. It is also worth noting that although adopting the $\mathcal{L}_{VA}$ only leads to smaller gains than the $\mathcal{L}_{VE}$ only, adopting both losses can achieve further improvement. It suggests that aligning two spike responses provides more effective supervision than adopting independent supervision or distillation only.

Obeservations about the overfitting problem. Fig. 6 visualizes performance comparison with different evaluation metrics and we can draw some interesting observations about overfitting. First, the primary classifier can reach much lower WER on the training set than the auxiliary classifier in Fig. 6(a), which reflects its powerful temporal modeling ability. Second, there exists a significant performance gap between the training and Dev sets on WDR, which indicates that the BiLSTM layers do not fully incorporate the visual information although it successfully overfits the training set. Third, the actual performance gap is much larger than WER shows (ΔWER*). For example, the

Figure 5. Qualitative comparison among different settings with examples from training (the upper) and Dev (the lower) sets of PHOENIX14. Wrong recognized glosses (except del) are marked in red. The primary classifier and auxiliary classifier outputs are marked as (P) and (A).



(a) Results on PHOENIX14 **training** set.



(b) Results on PHOENIX14 **Dev** set.

Figure 6. Performance comparison with different metrics and settings (ΔWER^* = $\text{WER}^*_a - \text{WER}^*_p$ = WAR$-$WDR).

performance gap between two classifiers of Baseline on Dev set in Fig. 6(b) is only **4.9%** (=30.4%-25.5%), however, the primary classifier makes **11.3%** correct predictions based on contextual information only (WAR) and ignores **6.5%** correct visual information (WDR). The proposed inconsistent prediction metrics provide a helpful tool to understand and evaluate the overfitting problem.

Obeservations about the performance gap. Another interesting observation from Fig. 6(b) is that while the iterative training strengthens the visual extractor, it also increases the WDR. We assume that the pseudo-label-based approach is not well compatible with the primary CTC loss (previous work [6] adopts a balanced ratio to reduce the effects of "blank" labels). Therefore, we adopt an additional CTC loss as our $\mathcal{L}_{VE}$ and it significantly improves both WAR and WDR. The proposed $\mathcal{L}_{VA}$ has a limited effect on the visual extractor but it can narrow the performance gap between two classifiers. The combined use of both auxiliary losses achieves better performance with a smaller actual performance gap (WDR and WAR), which verifies the effectiveness of the proposed visual alignment constraint.

Table 4. Ablation results (WER, %) of temporal layer design. Cβ and Pβ correspond to 1D convolutional layer and max pooling layer with a kernel size of β, respectively.

	Temporal Layers	Δt	Dev / Test
Frame-wise	C1	1	25.2 / 26.5
	C3	3	24.4 / 25.4
Subgloss-wise	C5-P2	6	24.0 / 24.3
Gloss-wise	C5-P2-C5-P2	16	**21.2 / 22.3**

Ablation on temporal network design. Previous pseudo-label-based methods need to carefully design the temporal receptive field, which is set to approximate the average length of the isolated sign [6, 9]. Table 4 presents the performance comparison with different temporal receptive fields Δt to show the effectiveness and flexibility of the proposed VAC. To our surprise, the frame-wise feature extractor still achieves competitive results to other settings, and there is a small performance differences in the temporal layer design. The VAC provides more flexible supervision for the feature extractor and results show that it is superior to iterative training sceme [9].

5.3. Qualitative Results

Results Visualization. To better understand the learning process, we give some recognized examples in Fig. 5. The upper sample from **the training set** shows that the auxiliary classifier of the baseline does not correctly recognize some glosses (NACHT, loc-SUEDWEST, ORT-PLUSPLUS), but the primary classifier can still deliver the correct result. Although it is reasonable for the primary classifier to make predictions based on contextual information only, the lack of constraint on the feature space increases the risk of overfitting, which may lead to unpredictable predictions when context changes during inference. With the help of the VAC, both auxiliary and primary classifiers are sufficiently trained and make better predictions on the training set.

The lower sample from **the Dev set** shows a failure case

Table 5. Performance comparison on PHOENIX14 dataset. Results of the proposed method are based on ResNet18 and Gloss-wise temporal layer. The entries denoted by "*" used extra clues (such as keypoints and tracked face regions).

Methods	Backbone	Iteration	Dev(%)		Test(%)	
			del/ins	WER	del/ins	WER
SubUNet [3]	CaffeNet		14.6/4.0	40.8	14.3/4.0	40.7
Staged-Opt [8]	VGG-S/GoogLeNet	✓	13.7/7.3	39.4	12.2/7.5	38.7
Align-iOpt [39]	3D-ResNet	✓	12.6/2.6	37.1	13.0/2.5	36.7
Re-Sign [31]	GoogLeNet	✓	-	27.1	-	26.8
SFL [36]	ResNet18		7.9/6.5	26.2	7.5/6.3	26.8
STMC [48]	VGG11	✓	-	25.0	-	-
DNF [9]	GoogLeNet	✓	7.8/3.5	23.8	7.8/3.4	24.4
FCN [6]	Custom		-	23.7	-	23.9
CMA [38]	GoogLeNet	✓	7.3/2.7	**21.3**	7.3/2.4	**21.9**
CNN+LSTM+HMM [27]*	GoogLeNet	✓	-	26.0	-	26.0
DNF [9]*	GoogLeNet	✓	7.3/3.3	23.1	6.7/3.3	22.9
STMC [48]*	VGG11	✓	7.7/3.4	**21.1**	7.4/2.6	**20.7**
Baseline	ResNet18		8.3/3.1	25.4	8.8/3.2	26.6
Baseline+VAC	ResNet18		7.9/2.5	**21.2**	8.4/2.6	**22.3**

Table 6. Performance comparison (%) on CSL dataset. The entry denoted by "*" used extra clues (keypoints).

Methods	WER
LS-HAN [23]	17.3
SubUNet [3]	11.0
SF-Net [47]	3.8
FCN [6]	3.0
STMC [48]*	2.1
Baseline	3.5
Baseline+VAC	**1.6**

of the alignment module. The auxiliary classifier makes the correct predictions (HEUTE, OST and SCHON) based on visual features only. Nevertheless, the primary classifier neglects this information and gives a worse result, which is not mentioned in the WER metric but can be identified by the proposed metrics. More qualitative results can be found in the supplementary material.

5.4. Comparison with the State-of-the-art.

We present the comparison results with several state-of-the-art approaches in Table 5 and Table 6. From Table 5 we can see that the proposed method with gloss-wise temporal layer and VAC achieves competitive results with previous iteration-based methods. We can also illustrate the success of STMC [48] and CMA [38] from the overfitting perspective: the former enforces the feature extractor to extract visual information from extra supervision and the latter weakens the contextual information with the data augmentation.

To examine the generalization of the proposed method, we also evaluate it on the CSL dataset. As no official split is given, the performance comparison among methods in Table 6 has limited practical value. The proposed method shows improvement than baseline and achieves better performance than recent work [6] under the same setting.

5.5. Discussion

We can roughly divide recent methods into two categories from the overfitting perspective: enhancing the feature extractor [6, 9, 39, 27, 47, 48] and weakening the alignment module [6, 31, 36]. The proposed VAC is an attempt to make better use of visual information, which provides a new perspective to solve this problem. How to better use visual features with a more powerful temporal model, which will be easier to overfit but can further improve WAR, is a challenging problem.

6. Conclusion

Overfitting is one of the major problems in CTC-based sign language recognition, which leads to insufficient training of the feature extractor. In this study, we propose the visual alignment constraint to make CSLR networks end-to-end trainable by enforcing the feature extractor to make predictions with more alignment supervision. Two metrics are proposed to measure the inconsistent predictions of the feature extractor and the alignment module. Experimental results show that the proposed VAC narrows the gap between predictions of the auxiliary and the primary classifiers. The proposed metrics and relevant experiments provide a new perspective on the relationship between visual and alignment modules, and we hope they can inspire future studies on CSLR and other sequence classification tasks.

Our source codes and trained models are available at `https://vipl.ict.ac.cn/resources/codes` or `https://github.com/ycmin95/VAC_CSLR`.

Acknowledgement. This study was partially supported by the Natural Science Foundation of China under contract No. 61976219.

References

[1] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016. 2

[2] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 16–31, 2019. 1

[3] 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. 1, 2, 8

[4] 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

[5] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 2

[6] 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, 6, 7, 8

[7] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793, 2018. 2

[8] 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. 1, 2, 4, 8

[9] 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, 3, 4, 7, 8

[10] Philippe Dreuw, David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. Speech recognition techniques for a sign language recognition system. In *Eighth Annual Conference of the International Speech Communication Association*, 2007. 1

[11] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *Proceedings of the International Workshop on Automatic Face and Gesture recognition*, volume 12, pages 296–301, 1995. 2

[12] Wen Gao, Gaolin Fang, Debin Zhao, and Yiqiang Chen. A chinese sign language recognition system based on sofm/srn/hmm. *Pattern Recognition*, 37(12):2389–2402, 2004. 2

[13] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[14] Alex Graves. Connectionist temporal classification. In *Supervised sequence labelling with recurrent neural networks*, volume 385 of *Studies in Computational Intelligence*, pages 61–62. Springer, 2012. 2

[15] 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, 2, 3

[16] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2008. 2

[17] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, 2013. 2, 4

[18] Junwei Han, George Awad, and Alistair Sutherland. Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623–633, 2009. 2

[19] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014. 4

[20] 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. 2, 6

[21] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *Neural Information Processing Systems Deep Learning and Representation Learning Workshop*, 2014. 5

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 3

[23] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 2018. 2, 6, 8

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456, 2015. 6

[25] Hamid Reza Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. In *Proceedings of the British Machine Vision Conference*, 2019. 2

[26] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-

task learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4835–4839, 2017. 2

[27] 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. 1, 2, 8

[28] Oscar Koller, Jens Forster, and Hermann Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, 2015. 2, 5, 6

[29] Oscar Koller, Hermann Ney, and Richard Bowden. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2016. 1, 2

[30] 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

[31] 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. 1, 2, 8

[32] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1459–1469, 2020. 2

[33] Dongxu Li, Xin Yu, Chenchen Xu, Lars Petersson, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6205–6214, 2020. 2

[34] Hongzhu Li and Weiqiang Wang. Reinterpreting ctc-based training as iterative fitting. *Pattern Recognition*, page 107392, 2020. 2, 3

[35] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10991–11000, 2020. 4

[36] 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. 1, 2, 8

[37] 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, 2

[38] 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

[39] 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, 4, 8

[40] Ramon Sanabria and Florian Metze. Hierarchical multitask learning with ctc. In *Proceedings of the 2018 IEEE Spoken Language Technology Workshop*, pages 485–490, 2018. 2

[41] Wendy Sandler and Diane Lillo-Martin. *Sign language and linguistic universals*. Cambridge University Press, 2006. 1

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2014. 2, 6

[43] Chao Sun, Tianzhu Zhang, Bing-Kun Bao, Changsheng Xu, and Tao Mei. Discriminative exemplar coding for sign language recognition with kinect. *IEEE Transactions on Cybernetics*, 43(5):1418–1428, 2013. 2

[44] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2

[45] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. 4

[46] Shuo Wang, Dan Guo, Wen-gang Zhou, Zheng-Jun Zha, and Meng Wang. Connectionist temporal fusion for sign language translation. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1483–1491, 2018. 1

[47] Zhaoyang Yang, Zhenmei Shi, Xiaoyong Shen, and Yu-Wing Tai. Sf-net: Structured feature network for continuous sign language recognition. *arXiv preprint arXiv:1908.01341*, 2019. 8

[48] 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. 3, 8