Connectionist Temporal Modeling of Video and Language: a Joint Model for Translation and Sign Labeling

Dan Guo , **Shengeng Tang** and **Meng Wang**
School of Computer Science and Information Engineering, Hefei University of Technology
guodan@hfut.edu.cn, tsg1995@hfut.edu.cn, eric.mengwang@gmail.com

Abstract

Online sign interpretation suffers from challenges presented by hybrid semantics learning among sequential variations of visual representations, sign linguistics, and textual grammars. This paper proposes a Connectionist Temporal Modeling (CTM) network for sentence translation and sign labeling. To acquire short-term temporal correlations, a Temporal Convolution Pyramid (TCP) module is performed on 2D CNN features to realize (2D+1D)=pseudo 3D′ CNN features. CTM aligns the pseudo 3D′ with the original 3D CNN clip features and fuses them. Next, we implement a connectionist decoding scheme for long-term sequential learning. Here, we embed dynamic programming into the decoding scheme, which learns temporal mapping among features, sign labels, and the generated sentence directly. The solution using dynamic programming to sign labeling is considered as pseudo labels. Finally, we utilize the pseudo supervision cues in an end-to-end framework. A joint objective function is designed to measure feature correlation, entropy regularization on sign labeling, and probability maximization on sentence decoding. The experimental results using the RWTH-PHOENIX-Weather and USTC-CSL datasets demonstrate the effectiveness of the proposed approach.

1 Introduction

This paper addresses problems associated with sign video interpretation, which is related to topics in the computer vision and machine learning fields, *i.e.*, gesture recognition [Joshi *et al.*, 2017], action detection and location [Nguyen *et al.*, 2018], human behavior analysis [Kacem *et al.*, 2018], and video understanding [Jelodar *et al.*, 2018]. Vision-based sign interpretation originates from isolated sign recognition [Wu *et al.*, 2016]; however, researchers are paying more and more attention to continuous Sign Language Translation (SLT) [Koller *et al.*, 2016a; Cui *et al.*, 2017]. Essentially, SLT aims to bridge the semantic gap between vision and language under complicated sign linguistics. Thus, we consider the SLT task, with a particular focus on online SLT.

SLT is challenging due to the following. (1) Visual hints under sign linguistics are latent and obscure, *i.e.*, facial expressions, lip languages, flexible signs of locality, and even some specific adjectives and adverbs. For example, the adverb "fast" is represented by increasing the speed of signing[Neverova *et al.*, 2016]. (2) SLT involves additional challenges related to hybrid semantics learning under vision understanding, sign recognition, and natural language translation. How to jointly learn an excellent visual representation, encode complicated sign linguistics, and decode grammatical sentences in a unified model framework remains a difficult problem to solve [Graves *et al.*, 2006]. (3) Sign videos have sentence-level annotations, rather than the exact temporal location of each sign action. It is a weakly-supervised sequence-to-sequence problem [Pu *et al.*, 2018]. (4) Accurate online video translation is also difficult. Some previous studies first encode the entire video and decode textual semantics, *e.g.*, for video captioning [Venugopalan *et al.*, 2015]. To improve the translation performance, offline algorithms have been used to fine-tune the model parameters [Koller *et al.*, 2016a; Cui *et al.*, 2017]. Our objective is end-to-end training for online translation with no additional supervision.

In this paper, we propose a joint model for online translation and sign labeling, which we refer to as a Connectionist Temporal Modeling (CTM) network. As shown in Figure 1, the pre-trained deep models ResNet-18 [He *et al.*, 2016] and ResNet-3D [Hara *et al.*, 2017] are used to extract 2D and 3D CNN features, respectively. Here a 3D CNN learns the spatiotemporal hints in the short-term clips, while 2D CNN features remain visual details at the frame-level. To align and fuse these features under different granularities, a Temporal Convolution Pyramid (TCP) module is designed to compress the 2D features by convoluting several adjacent features. As shown in Figure 2, there are three-stage convolutional operations, and each temporal convolution span involves two non-overlapping time steps in the TCP. Thus, the long frame-level 2D features (eight time steps) are transformed into a compact clip-level 3D feature (one time step). In fact, the TCP is a "pseudo" 3D feature extraction operation, *i.e.*, 2D spatial +1D temporal convolutional operations. Finally, we adopt Multi-Layer Perceptron (MLP) to fuse pseudo 3D and original 3D features.

Next, three modules in the CTM framework, *i.e.*, Connectionist Temporal TRanslation (CTTR), Feature CLaSsi-

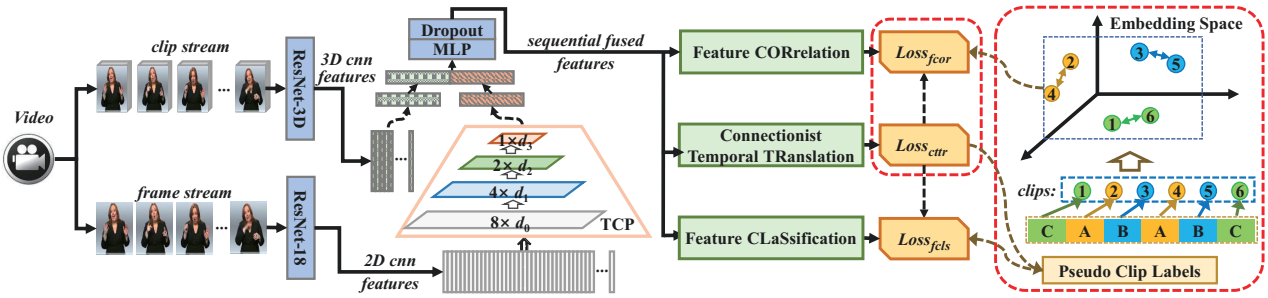

Figure 1: Overview of the proposed CTM framework for online SLT. Given a video, we extract 2D frame-level and 3D clip-level feature streams using the pre-trained models ResNet-18 and ResNet-3D, respectively. The TCP module is conducted on the 2D features to learn short-term temporal clues, and align them to the 3D features. The detailed implementation of TCP is shown in Figure 2. Then, the fused features are fed into three modules for long-term sequential learning, as shown in Figure 3. Finally, we utilize pseudo supervision cues in the online deep model. A joint loss optimization combining $\mathcal{L}_{fcor}$, $\mathcal{L}_{cttr}$, and $\mathcal{L}_{fcls}$, is designed to measure feature correlation, sentence decoding, and entropy regularization on sign labeling.

fication (FCLS), and Feature CORrelation (FCOR), receive aforementioned fused features for long-term sequential learning. As shown in Figure 3, the first CTTR module performs sentence translation, while the FCLS and FCOR modules measure sign labelling at the word-level. In particular, the CTTR module provides a translated sentence with the maximum probability under multiple decomposed paths of sign word labeling. It addresses the task in a weakly-supervised learning manner. Since videos are annotated at the sentence-level, CTTR is the primary contributive module in the long-term sequential learning.

To further address the challenges associated with weakly supervision learning, we integrate the pseudo-supervised learning into the deep CTM model. In our solution, $\mathcal{L}_{cttr}$, $\mathcal{L}_{fcls}$, and $\mathcal{L}_{fcor}$ involve the objective functions of above three deep modules, respectively. Using $\mathcal{L}_{cttr}$, we calculate a sign labeling path with maximum probability corresponding to the sentence label and consider it as sequential sign labels of features, *i.e.*, pseudo labels. Based on the pseudo labels, $\mathcal{L}_{fcls}$ measures the entropy regularization on feature classification, and $\mathcal{L}_{fcor}$ is a triple loss relative to feature correlation. $\mathcal{L}_{fcor}$ models the feature similarity under the same sign class or different sign classes, *i.e.*, similarity difference among positive or negative feature samples. Finally, $\mathcal{L}_{cttr}$, $\mathcal{L}_{fcls}$, and $\mathcal{L}_{fcor}$ are combined as the objective function of the overall framework. By minimizing the joint loss, the model is pushed to learn more temporal cues. The primary contributions of this study can be summarized as follows:

- Learning a good clip representation facilitates the acquisition of the short-term temporal correlation. The TCP is conducted on 2D CNN features to realize (2D+1D)=pseudo 3D′ CNN. We align the 3D′ features with the original 3D features and fuse them.
- We propose a connectionist decoding scheme for long-term sequential learning, where the decoder embeds the dynamic programming optimization into end-to-end deep learning. It learns the connectionist mapping among features, sign words, and the generated sentence.
- Pseudo supervision cues are utilized for online learning. We design a joint objective function to measure sentence translation, feature correlation, and entropy regularization based on pseudo labels, where pseudo labels denote the sign word labels obtained from previous connectionist decoding.

2 Related Work

2.1 Sequence-to-sequence Learning

Learning temporal cues in videos is very important. In this paper, we do not discuss traditional temporal models, such as Dynamic Time Warping (DTW) [Lin *et al.*, 2014] and Hidden Markov Models (HMM) [Guo *et al.*, 2017], in detail. For SLT, both DTW and HMM require significant training time. DTW implements template matching on the entire dataset and HMM represents the transformation among a large number of latent states. With the rapid development of deep learning, various RNN deformations have become more prevalent and effective in sequential learning. Liu et al. proposed an LSTM-based end-to-end neural network that was effective for isolated sign recognition [Liu *et al.*, 2016]. Given the ability of CNNs to extract features and RNNs to perform sequential learning, many hybrid models were emerged, *i.e.*, temporal convolution and bidirectional RNNs [Pigou *et al.*, 2018], Recurrent 3D CNNs [Lefebvre *et al.*, 2015], and depth DNN embedded with HMM [Wu *et al.*, 2016]. These hybrid methods can model the spatiotemporal variations simultaneously. Note that we also exploit the merits of CNNs and RNNs in our proposed model.

2.2 Weakly-supervised Learning in Videos

As SLT is a typical weakly-supervised task, some researchers have focused on action location and word alignment. To address word alignment, Koller et al. embedded a deep CNN into the hybrid HMM framework [Koller *et al.*, 2017]. Cui et al. proposed an LSTM and Connectionist Temporal Classification (CTC) framework to address gloss-level classification [Cui *et al.*, 2017]. In contrast, the encoder-decoder framework, which is widely used in neural machine translation and video captioning, is prevalent for weakly-supervised sequential learning. Guo *et al.* proposed an encoder-decoder framework with variable-length clip mining for a Chinese

sign language translation [Guo *et al.*, 2018]. A similar study [Cihan Camgoz *et al.*, 2018] proposed an attention-based encoder-decoder network comprising two specialized RNN modules. However, these proposed methods decoded word by word after encoding all visual content. They do not apply to online SLT. Online sequential action recognition [Liu *et al.*, 2018] is closely related to our study. However, the task assumed strict supervision cues, *i.e.*, exact start and end time labels for each action. Our task differs in that it involves sentence-level labels without exact temporal cues at the word-level. Compared to the problem that Liu *et al.* [Liu *et al.*, 2018] addressed, online SLT has fewer temporal cues.

In order to enhance temporal cues, some studies have adopted multiple iterations of the EM algorithm to perform offline fine-tuning. Koller *et al.* trained a frame-level classifier on extra sign language dataset by embedding a CNN in an iterative EM algorithm [Koller *et al.*, 2016a]. In [Koller *et al.*, 2016b], the proposed model embedded a CNN into a HMM, which feeds the output of the CNN to a Bayesian model in the HMM for continuous SLT. Cui *et al.* designed a three-stage optimization process, *i.e.*, feature extraction, word alignment, and sequence learning [Cui *et al.*, 2017]. In [Pu *et al.*, 2018], authors alternately optimized the feature extractor and the sequence learning model using dilated convolution operations. Differing from these offline iterations, the proposed model introduces pseudo supervision cues into an end-to-end framework for online sequential learning.

3 Proposed Method

The general framework of our model is shown in Figure 1. Given a video $\boldsymbol{\mathcal{V}} = \{v_n\}_{n=1}^{\mathcal{N}}$, two feature streams, *i.e.*, frame-level features $\boldsymbol{f_{2d}} = (f_1, \cdots, f_{\mathcal{N}})$ and clip-level features $\boldsymbol{f_{3d}} = (clip_1, \cdots, clip_N)$, are extracted, and then a sequence of gloss labels $\boldsymbol{\mathcal{Y}} = (y_1, \cdots, y_m)$ is output. We discuss each module in detail in the following.

3.1 Clip Feature Learning in Videos

Sequential learning always suffers gradient attenuation along temporal transitions. Learning a good clip representation is to acquire the compact short-term temporal correlation. In this section, we discuss the extraction of 2D and 3D CNN features, the alignment of 2D and 3D, and feature fusion. The advantage of this clip representation is that both 3D and temporal 2D CNNs are leveraged to learn discriminant features.

TCP on 2D Features

Motivated by the $\triangle t$-gram language model used in natural language processing tasks, in the Temporal Convolution Pyramid (TCP) module, each layer (temporal convolution layer; TCOV) implements the embedding of n-item adjacent features. It calculates local convolution in the short-term temporal view. In other words, the TCP is a pseudo 3D feature learning operation, *i.e.*, the combing 2D spatial and 1D temporal convolutions.

Given the original 2D frame features $\{\boldsymbol{f_{2d}}\} \in \mathbb{R}^{d_0 \times \mathcal{N}}$, an l-layer TCP transforms them to pseudo clip features $\{\boldsymbol{f'_{3d}}\} \in \mathbb{R}^{d_l \times \frac{\mathcal{N}}{\prod_{i=1}^{l}(s_i)}}$. The TCP calculation is expressed as follows:

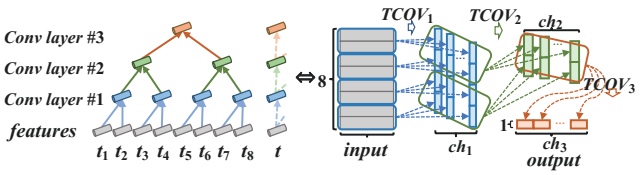

Figure 2: Temporal Convolution Pyramid (TCP) on 2D features.

$$\{\boldsymbol{f'^{n}_{3d}}\}_{n=1}^{N} = TCOV_{\Phi_l}\left\{\cdots\left[TCOV_{\Phi_1}(\boldsymbol{f^{n}_{2d}}|_{n=1}^{\mathcal{N}})\right]\right\} \quad (1)$$

where Φ_i indicates the parameter of the i-layer TCOV$_i$ in the TCP, and Φ_i =(ch_i, d_i, $\triangle t_i$, s_i, pad_i) denotes the format of a convolutional parameter (number of channels, height, width, stride, and padding). $d_i \times \triangle t_i$ denotes the convolutional kernel size, and s_i is the sliding window along the temporal dimension. Here, $d_{i+1} = ch_i$, *i.e.*, the kernel size for the (i+1)-th layer is set to the output dimension of the i-th layer.

As shown in Figure 2, there is a three-layer TCP with ($\triangle t_i = 2$)-gram and non-overlapping ($s_i = 2$). Thus, the frame-level 2D features (eight time steps) are transformed into a short clip-level 3D feature (one time step). The TCP gradually condenses temporal cues via contiguous $\triangle t$-items.

2D & 3D Feature Alignment and Fusion

Give the temporal dimension of the original 3D features, a compact pseudo 3D feature is obtained using the TCP; thus, the model can align pseudo 3D feature $\boldsymbol{f'}_{3d}$ to the original 3D features $\boldsymbol{f}_{3d}$, and fuse them. Here, the entire clip representation enforces feature fusion [(2D+1D $\approx$ 3D′)+3D]. The fusion scheme using the MLP is formulated as follows:

$$\boldsymbol{F}_{fus} = \{\boldsymbol{f}_n\}_{n=1}^{N} = MLP(\boldsymbol{f}'_{3d} \oplus \boldsymbol{f}_{3d}) \quad (2)$$

where $\oplus$ represents the concatenation operation on vectors.

3.2 Connectionist Temporal Translation

After feature fusion, we tackle long-term sequential learning. This subsection elaborates on the Connectionist Temporal TRanslation (CTTR) module in Figure 3. The other two modules are introduced in Section 3.3.

Temporal Encoding

The BGRU excels at modeling forward and backward contexts; thus, it is robust and effective for sequential action recognition. We employ the BGRU, expressed as follows, as the basic encoding RNN unit.

$$\boldsymbol{\mathcal{H}} = \{\boldsymbol{h_n}\}_{n=1}^{N} = \{BGRU\,(\boldsymbol{f_n})\}_{n=1}^{N} \quad (3)$$

Then, based on the output of BGRU, we employ a fully-connected layer FC to embed the output into non-normalized categorical probabilities with K sign classes as follows:

$$\boldsymbol{\mathcal{P}} = \{p_n\}_{n=1}^{N} = \varphi_{softmax}\left[FC(\{\boldsymbol{h}_n\}|_{n=1}^{N})\right] \quad (4)$$

where $\varphi_{softmax}$ is the softmax function, $\boldsymbol{\mathcal{P}}$ is the probability score matric, $p_n \in \mathbb{R}^K$ is the categorical probability vector of the n-th clip, and K equals the vocabulary size plus 1 (a new introduced blank symbol '_').

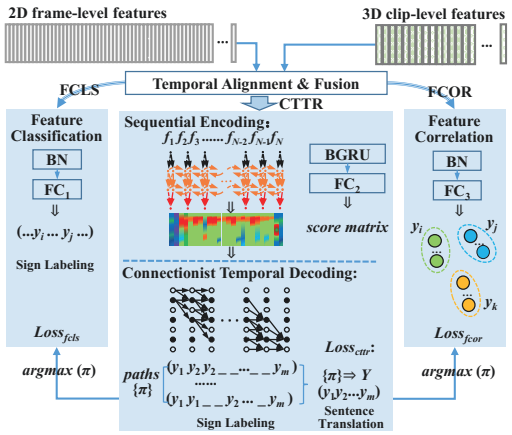

Figure 3: Architecture of the proposed approach for online SLT. The middle CTTR module decodes the connectionist mapping among features, words, and the generated sentence. Pseudo supervision cue π is utilized on both two side modules (FCLS and FCOR).

Decoding Optimization

Let word alignment π be a sequence of words including ordinary words and the blank symbol '_', *i.e.*, $\pi = \{\pi_n\}_{n=1}^{N}$, where π_n denotes the n-th word of π. The probability of π is given by the product of probabilities as follows:

$$P^{\pi} = Prob(\pi) = \prod_{n=1}^{N} p_n^{\pi_n} \tag{5}$$

where $p_n^{\pi_n}$ denotes the probability of word π_n at time step n.

Motivated by dynamic programming optimization [Graves *et al.*, 2006], we define a two-stage greedy strategy on a long word label sequence. It outputs a short sentence by removing the blank label in the first stage and deletes continuous repetitions in the second stage. Note that a target sequence $\mathcal{Y}$ may correspond to multiple different alignments $\{\pi\}$; thus, the CTTR module defines a many-to-one mapping as $\mathcal{B}$, which removes all blanks and repeated labels from the alignments. As shwon in Figure 3, $\mathcal{B}$ implements the transformation between sign labeling and sentence translation. Therefore, in the training process, the CTTR decoder objective function is defined as follows:

$$\mathcal{L}_{cttr} = \sum_{\pi=\mathcal{B}^{-1}(\mathcal{Y})} -logP^{\pi} = -\sum_{\pi=\mathcal{B}^{-1}(\mathcal{Y})} \sum_{n=1}^{N} p_n^{\pi_n} \tag{6}$$

where $\mathcal{B}^{-1}(\mathcal{Y}) = \{\pi \mid \mathcal{B}(\pi) = \mathcal{Y}\}$ is the set of all alignments. In the testing stage, we output the word alignment using the maximum function on score matrix $\mathcal{P} = \{p_n\}_{n=1}^{N}$ and transform it to a sentence $\mathcal{Y} = \{y_n\}_{n=1}^{m}$.

3.3 Joint Loss Optimization

To further enhance the temporal cues in videos, we introduce the concept of pseudo-supervision. Note that pseudo labels denote the sign labeling path $\pi = \{\pi_n\}_{n=1}^{N}$ with the maximum product of probabilities P_{max}^{π} (Section 3.2). Here, $\pi = \{\pi_1, \cdots, \pi_n\}$ is taken as the available pseudo word labels of the sequential features. Based on these labels, the FCLS module evaluates the feature classification entropy. In addition, the FCOR module measures the inter- and intra- similarities of different feature groups under K sign classes. Both modules contribute to inhibit overfitting of the CTTR module.

Here, based on training set $\mathcal{S}$ (all video samples and annotations), we create set $\mathcal{M}$ of clip-level fused features and set $\mathcal{T}$ of all pseudo triplets in all training epochs. The objective function of the entire training framework is given as follows:

$$\mathcal{L} = \frac{1}{|\mathcal{S}|}\sum \mathcal{L}_{cttr} + \frac{1}{|\mathcal{M}|}\sum \mathcal{L}_{fcls} + \frac{1}{|\mathcal{T}|}\sum \mathcal{L}_{fcor} \tag{7}$$

where $\mathcal{L}_{cttr}$ is already given in formula (6), while $\mathcal{L}_{fcls}$ and $\mathcal{L}_{fcor}$ are expressed in following formulae 8 and 11.

Cross-entropy Loss

Based on fused features $\{\boldsymbol{f}_n\}$ obtained in subsection 3.1, we realize the FCLS module via Fully Connected layer (FC), Batch Normalization (BN), and softmax operations to obtain predicted probabilities at each clip-level time step. Here, we define $\mathbf{m}$ as a feature sample in $\mathcal{M}$ and $\pi_{\mathbf{m}}$ is its pseudo label. If the sequential features are reliable, the new predicted probability of $\mathbf{m}$ is close to its pseudo label $\pi_{\mathbf{m}}$. Note that we adopt cross entropy loss to measure the predicted probabilities and pseudo labels as follows:

$$\mathcal{L}_{fcls}(\mathcal{M}) = -\sum_{\mathbf{m}\in\mathcal{M}}\sum_{k\in K} \mathbf{y}_{\mathbf{m}}^{k} \log(p_{\mathbf{m}}^{k}) \tag{8}$$

where $\mathbf{y}_{\mathbf{m}}^{k}$ obtained by the CTTR module indicates whether sample $\mathbf{m}$ belongs to class $\pi_{\mathbf{m}}$ (value of 1 or 0), and $p_{\mathbf{m}}^{k}$ is the new predicted probability calculated by the FCLS module.

Triplet Loss

The triplet loss is designed to ensure that features with the same label have close embeddings together, while features with different labels are distant in the embedding space. We adopt the FC and BN operations in the FCOR module to model the new feature embeddings of $\{\boldsymbol{f}_n\}$.

Based on the sign labeling path π with the maximum product of probabilities P_{max}^{π}, we split these embedding features into different groups. In summary, features are divided into positive pairs with the same word label; otherwise, features are divided into the negative pairs. We denote it as set $\mathcal{T} = \{(e_+, e_+), (e_+, e_-), (e_-, e_+)\}$. For the video example shown in Figure 4, e_1 and e_6 are (e_+, e_+), while e_1 and e_2 are (e_+, e_-)/(e_-, e_+). In the correlation matrix, the colored squares indicate positive pairs, and the gray squares indicate negative pairs. Here, squares with snowflake points represent pairs containing a blank label '_' with no word meaning. Note that we do not consider self-pairs along the diagonal line and squares with snowflake points (blank label '_') in the matrix.

The distance measurement for positive and negative feature pairs must satisfy the following constraint:

$$\begin{cases} s(e_+, e_+) > s(e_+, e_-) + \alpha \\ s(e_+, e_+) > s(e_-, e_+) + \alpha \\ s.t. \quad s(a,b) = \frac{a^T b}{\|a\|\|b\|} = L_2(a)^T \cdot L_2(b) \end{cases} \tag{9}$$

where $s(a, b)$ is similarity score between features a and b, L_2 represents L_2-normalization, and parameter α controls

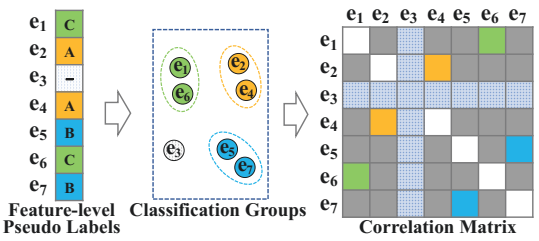

Figure 4: Triplet loss calculation based on different classification groups for feature correlation. e_3 indicates a blank symbol '_'. In the matrix, we do not consider diagonals and squares with snowflakes, where self-correlation and the blank label '_' have no word meaning.

similarity intensity during training. Here, we seek similarity scores for positive pairs that are greater than both negative pairs and the margin α.

Thus, we set triplet loss $\mathcal{L}_{fcor}$ to calculate all triplets $\{(e_+, e_+), (e_+, e_-), (e_-, e_+)\}$ in dataset $\mathcal{T}$. The similarity measurement $s(a, b)$ by L_2-normalization is symmetrical; thus, normal triplet loss is formalized as follows:

$$\mathcal{L}_{tri}(\mathcal{T}) = \sum_{t \in \mathcal{T}} max(s(t_{neg}) - s(t_{pos}) + \alpha, 0) \tag{10}$$

where $(e_+, e_-) = (e_-, e_+)$ is denoted t_{neg}, and (e_+, e_+) is denoted t_{pos} in $\mathcal{T}$.

As shown in Figure 4, positive pairs in a video are much less than negative pairs, no more than in the whole dataset. Thus, we adjust the calculation of triplet loss. For each training batch, we randomly sample the same number of negative pairs as positive pairs in formula (10). Then, we evaluate positive pairs individually; the same evaluation is performed for negative pairs. Finally, we adopt the triplet loss $\mathcal{L}_{fcor}$ as follows:

$$\begin{aligned} \mathcal{L}_{fcor}(\mathcal{T}) &= \sum_{t \in \mathcal{T}} max(s(t_{neg}) - s(t_{pos}) + \alpha, 0) \\ &= \sum_{t \in \mathcal{T}} max(s(t_{neg}) - \beta, 0) + \sum_{t \in \mathcal{T}} max(\beta - s(t_{pos}), 0) \end{aligned} \tag{11}$$

where β controls similarity intensity during training.

4 Experiment

4.1 Dataset and Evaluation

We experiment on two benchmarks, *i.e.*, RWTH-PHOENIX-Weather [Koller *et al.*, 2015] and USTC-CSL [Huang *et al.*, 2018]. The PHOENIX dataset includes 6841 Germany sign videos played by nine signers. The dataset is split into three independent parts, *i.e.*, "TRAIN", "VAL" and "TEST". The CSL dataset includes 5000 Chinese sign videos annotated by 50 signers. Under the "Split II" constraint [Guo *et al.*, 2018], it selects video samples of 94 sentences for training. The remaining unseen six sentences are used for testing.

To evaluate performance, Word Error Rate (**WER**) [Koller *et al.*, 2016a] measures the minimum number of insertion, replacement and deletion operations converting generated sequence L' to ground truth L. Here, **ins** and **del** denotes the proportions of insertion and deletion operations divided by length $|L|$, respectively.

input size	Layer	kernel, channel, stride	output size
$t \times d_0$	TCOV_1	$2 \times d_0,\ d_1,\ 2$	$(t/2) \times d_1$
	TCOV_2	$2 \times d_1,\ d_2,\ 2$	$(t/4) \times d_2$
	TCOV_3	$2 \times d_2,\ d_3,\ 2$	$(t/8) \times d_3$

Table 1: Parameter setting of the TCP module on 2D CNN features.

Features	VAL(%) del / ins / WER	TEST(%) del / ins / WER
$\boldsymbol{f_{2d}}$	55.1 / 1.5 / 69.4	53.6 / 1.8 / 68.3
$\boldsymbol{f'_{3d}}$	27.5 / 5.8 / 63.6	26.8 / 6.1 / 62.2
$\boldsymbol{f_{3d}}$	21.0 / 5.1 / 45.1	20.0 / 5.5 / 45.4
$\boldsymbol{f'_{3d}} + \boldsymbol{f_{3d}}$	10.5 / 7.3 / 42.2	10.8 / 7.8 / 42.2
$\boldsymbol{Fusion}_{\{f'_{3d}, f_{3d}\}}$	10.6 / 6.9 / **41.0**	10.1 / 7.9 / **41.3**

Table 2: Performance comparison on PHOENIX dataset using different features with the $\mathcal{L}_{cttr}$ loss.

4.2 Implementation

Videos in the RWTH-PHOENIX-Weather dataset are under pixels 210×260, which covers the human body. In the USTC-CSL dataset, each frame has 1280×720 pixels, which contains significant amounts of redundant blank background. We segment a local region with 210×260 pixels to cover the entire human body. Then, all images in both two datasets are resized to 224×224 pixels, and we combine the adjacent eight frames with a clip of four frames of overlap for ResNet-3D. We feed the same data as the ResNet-18 input.

We adopt the ADAM optimizer and set the batch size to 40 and initial learning rate of the overall network to 1×10^{-4}. Here, the learning rate is gradually reduced by $1/10$ every 20 epochs. As shown in Table 1. We set parameters d_0=d_1=d_2=d_3=512-dim in the TCP. Note that each d_i can differ. We also use ReLU and Dropout operations after each TCOV layer in the TCP to avoid over-fitting. We set the Dropout parameter to 0.2. Moreover, the initial learning rate of the TCP is 5×10^{-4}. $\beta = 0.5$ was found to provide the best experimental performance.

4.3 Model Validation

We test different features to reflect the impacts of the TCP and feature fusion modules. As shown in Table 2, the **del** rate of $\boldsymbol{f_{2d}}$ reaches 55.1%, which is far worse than other features. It indicates that as $\boldsymbol{f_{2d}}$ becomes overly long, it generates many redundant incoherent word labels along the temporal dimension. By introducing the TCP on $\boldsymbol{f_{2d}}$, the **del** rate significantly improves the effectiveness by dropping approximately 50% of the words. Directly adding $\boldsymbol{f_{3d}}'$ and $\boldsymbol{f_{3d}}$ improves performance; however, the improvement is insufficient. By implementing MLP fusion on $\boldsymbol{f_{3d}}'$ and $\boldsymbol{f_{3d}}$, the WER of the fused features is reduced from 42.2% to 41.0% and 42.2% to 41.3% on the VAL and TEST sets, respectively.

We verify the joint loss optimization, *i.e.*, the effectiveness of pseudo-supervision optimization. The CTM framework primarily conducts end-to-end weakly supervised learning with $\mathcal{L}_{cttr}$. By introducing $\mathcal{L}_{fcls}$, the performance on both the VAL and TEST sets are improved by 1.1% (Table 4). With the auxiliary of $\mathcal{L}_{fcor}$, the model learns the similarities and differences among clip-level features, which further re-

Methods	Off-line Iterations	Modality			VAL(%)		TEST(%)	
		hand	traj	face	des / ins	WER	des / ins	WER
HOG-3D [Koller *et al.*, 2015]	-	√			25.8 / 4.2	60.9	23.2 / 4.1	58.1
CMLLR [Koller *et al.*, 2015]	-	√	√	√	21.8 / 3.9	55.0	20.3 / 4.5	53.0
1-Mio-H [Koller *et al.*, 2016a]	3	√			19.1 / 4.1	51.6	17.5 / 4.5	50.2
1-Mio-H+CMLLR [Koller *et al.*, 2016a]	3	√	√	√	16.3 / 4.6	47.1	15.2 / 4.6	45.1
CNN-Hybrid [Koller *et al.*, 2016b]	3	√			12.6 / 5.1	38.3	11.1 / 5.7	38.8
Staged-Opt-init [Cui *et al.*, 2017]	-	√			16.3 / 6.7	46.2	15.1 / 7.4	46.9
Staged-Opt [Cui *et al.*, 2017]	3	√			13.7 / 7.3	39.4	12.2 / 7.5	38.7
SubUNets [Camgoz *et al.*, 2017]	-		√		14.6 / 4.0	40.8	14.3 / 4.0	40.7
Dilated-CNN-init [Pu *et al.*, 2018]	-				18.5 / 2.6	60.3	18.1 / 2.8	59.7
Dilated-CNN [Pu *et al.*, 2018]	**5**				**8.3 / 4.8**	**38.0**	**7.6 / 4.8**	**37.3**
Our Method	**-**				**11.6 / 6.3**	**38.9**	**10.9 / 6.4**	**38.7**

Table 3: Performance comparison with PHOENIX dataset. "Hand," "traj," and "face" indicate extra data-augmentation. "Off-line Iterations" refers to the number of offline optimizations, and "-" represents an end-to-end learning framework with no offline iteration.

Loss	VAL(%) del / ins / WER	TEST(%) del / ins / WER
$\mathcal{L}_{cttr}$	10.6 / 6.9 / 41.0	10.1 / 7.9 / 41.3
$\mathcal{L}_{cttr}$+$\mathcal{L}_{fcls}$	10.2 / 6.7 / 39.9	10.3 / 7.7 / 40.2
$\mathcal{L}_{cttr}$+$\mathcal{L}_{fcor}$	11.3 / 6.7 / 39.8	10.9 / 6.9 / 40.0
$\mathcal{L}_{cttr}$+$\mathcal{L}_{fcls}$+$\mathcal{L}_{fcor}$	11.8 / 5.9 / **38.9**	10.6 / 6.1 / **38.7**

Table 4: Results on PHOENIX dataset using different loss functions.

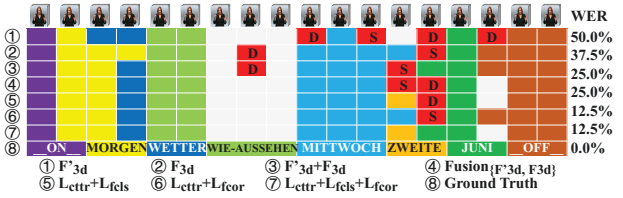

Figure 5: Example of decoding words using different module settings. "S" and "D" denote substitution and deletion operations.

duces the WER value on the VAL and TEST sets by 1.0% and 1.5%, respectively. When $\mathcal{L}_{cttr}$, $\mathcal{L}_{fcls}$, and $\mathcal{L}_{fcor}$ are used simultaneously, the proposed approach achieves the best results. A comparison of the obtained results relative to different module settings is shown in Figure 5.

4.4 Main Comparison

Here, we compare the proposed approach to the state of the art methods. As shown in Table 3, there are two obvious conclusions. (1) Extra visual hints are widely used to improve performance in previous studies, such as introducing visual representations of human body parts (hand and face) and pose trajectory. Moreover, 1M-Hands [Koller *et al.*, 2016a] imported a pre-trained sign vocabulary, and CNN-Hybrid [Koller *et al.*, 2016b] utilized an initialized word alignment. In contrast, the proposed model has no additional initialization and extra input data. (2) Most approaches adopt additional offline optimizations, such as 1-Mio-H, CNN-Hybrid, Staged-Opt [Cui *et al.*, 2017], and Dilated-CNN [Pu *et al.*, 2018]. Without offline iterations, the proposed CTM demonstrates much better performance than HOG-3D [Koller *et al.*, 2015], CMLLR [Koller *et al.*, 2015], SubUNets [Camgoz *et al.*, 2017], Staged-Opt-init and Dilated-CNN-init. Note that Staged-Opt and Dilated-CNN show excellent performance with offline iterations. Their initial WER value reduce rapidly to 46.9% and 59.7% on the TEST, and these values are much lower than those obtained by the proposed CTM model. Note that besides time consuming, offline iteration has another weakness, *i.e.*, it is always trained repeatedly using fixed datasets, which is not applicable to dataset extension. The proposed CTM model does not suffer this limitation, and still achieves comparable performance without extra supervision and offline iterations.

Methods	TEST WER(%)
S2VT [Venugopalan *et al.*, 2015]	67.0
S2VT(3-layer) [Yao *et al.*, 2015]	65.2
HLSTM [Guo *et al.*, 2018]	66.2
HLSTM-attn [Guo *et al.*, 2018]	64.1
Our Method	**61.9**

Table 5: Performance comparison on USTC-CSL dataset.

In addition, as shown in Table 5, this proposed model also achieves the best performance compared to other methods on the USTC-CSL dataset. The proposed model demonstrates the performance gains of 2.2∼5.1%. More importantly, both S2VT and HLSTM involves a encoding-decoding architecture for sentence generation, and this architecture decodes sentences after encoding the entire video. In contrast, the proposed CTM model exploits online connectionist temporal encoding; thus, the proposed approach is more flexibility relative to online SLT.

5 Conclusion

The paper proposes a temporal convolution pyramid module to learn the short-term temporal correlation, and a connectionist decoding scheme for long-term sequential learning. In addition, we design a joint objective function to optimize online learning under pseudo supervision. The proposed CTM model represents an end-to-end deep network for online translation and sign labeling. The experimental results demonstrate that the proposed approach achieves results that comparable to state of the art methods.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under grants 61725203, 61732008, and 61876058.

References

[Camgoz *et al.*, 2017] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Subunets: end-to-end hand shape and continuous sign language recognition. In *ICCV*, pages 3075–3084, 2017.

[Cihan Camgoz *et al.*, 2018] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *CVPR*, pages 7784–7793, 2018.

[Cui *et al.*, 2017] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *CVPR*, pages 1610–1618, 2017.

[Graves *et al.*, 2006] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[Guo *et al.*, 2017] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Online early-late fusion based on adaptive hmm for sign language recognition. *TOMM*, 14(1):8, 2017.

[Guo *et al.*, 2018] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. Hierarchical lstm for sign language translation. In *AAAI*, 2018.

[Hara *et al.*, 2017] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV*, pages 3154–3160, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Huang *et al.*, 2018] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI*, 2018.

[Jelodar *et al.*, 2018] Ahmad Babaeian Jelodar, David Paulius, and Yu Sun. Long activity video understanding using functional object-oriented network. *TMM*, 2018.

[Joshi *et al.*, 2017] Ajjen Joshi, Soumya Ghosh, Margrit Betke, Stan Sclaroff, and Hanspeter Pfister. Personalizing gesture recognition using hierarchical bayesian neural networks. In *CVPR*, pages 6513–6522, 2017.

[Kacem *et al.*, 2018] Anis Kacem, Mohamed Daoudi, Boulbaba Ben Amor, Stefano Berretti, and Juan Carlos Alvarez-Paiva. A novel geometric framework on gram matrix trajectories for human behavior understanding. 2018.

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

[Koller *et al.*, 2016a] 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 *CVPR*, pages 3793–3802, 2016.

[Koller *et al.*, 2016b] Oscar Koller, O Zargaran, Hermann Ney, and Richard Bowden. Deep sign: hybrid cnn-hmm for continuous sign language recognition. In *BMVC*, 2016.

[Koller *et al.*, 2017] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms. In *CVPR*, pages 4297–4305, 2017.

[Lefebvre *et al.*, 2015] Grégoire Lefebvre, Samuel Berlemont, Franck Mamalet, and Christophe Garcia. Inertial gesture recognition with blstm-rnn. In *ANN*, pages 393–410. Springer, 2015.

[Lin *et al.*, 2014] Yushun Lin, Xiujuan Chai, Yu Zhou, and Xilin Chen. Curve matching from the view of manifold for sign language recognition. In *ACCV*, pages 233–246, 2014.

[Liu *et al.*, 2016] Tao Liu, Wengang Zhou, and Houqiang Li. Sign language recognition with long short-term memory. In *ICIP*, pages 2871–2875, 2016.

[Liu *et al.*, 2018] Jun Liu, Amir Shahroudy, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ssnet: scale selection network for online 3d action prediction. In *CVPR*, pages 8349–8358, 2018.

[Neverova *et al.*, 2016] Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. Moddrop: adaptive multi-modal gesture recognition. *TPAMI*, 38(8):1692–1706, 2016.

[Nguyen *et al.*, 2018] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, pages 6752–6761, 2018.

[Pigou *et al.*, 2018] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video. *IJCV*, 126(2-4):430–439, 2018.

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

[Venugopalan *et al.*, 2015] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015.

[Wu *et al.*, 2016] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *TPAMI*, 38(8):1583–1597, 2016.

[Yao *et al.*, 2015] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.