Improving Continuous Sign Language Recognition with Cross-Lingual Signs Supplementary Material

A. Implementation Details

A.1. Dictionary Construction

In Section 3.1, we employ a pre-trained CSLR model to partition the continuous sign videos into isolated sign clips. Here we describe the partition algorithm as follows.

To construct a dictionary $\mathcal{C}$ for a dataset $\mathcal{D}$ with an alphabet $\mathcal{S}$. Given a training video $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T) \in \mathcal{D}$ containing T frames and its associated ground-truth sign sequence $\boldsymbol{s} = (s_1, \ldots, s_N), s_i \in \mathcal{S}$, a well-trained CSLR model produces a frame-wise prediction sequence $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$ in which $\boldsymbol{y}_t \in \mathbb{R}^{|\mathcal{S}'|}$ is a probability distribution over the expanded alphabet $\mathcal{S}' = \mathcal{S} \cup \{blank\}$ for the t-th frame[1]. Therefore, the probability of a frame-wise sequence $\boldsymbol{\pi}_{1:T} = (\pi_1, \ldots, \pi_T)$ where $\pi_t \in \mathcal{S}'$, can be computed as

$$p(\boldsymbol{\pi}_{1:T} \mid \boldsymbol{v}) = \prod_{t=1}^{T} \boldsymbol{y}_t(\pi_t), \tag{1}$$

where $\boldsymbol{y}_t(\pi_t)$ indicates the probability of observing label π_t at timestamp t.

A frame-wise sequence $\boldsymbol{\pi}_{1:T}$ can be mapped to a sign sequence by removing *blank* predictions and deduplicating the repeated non-blank predictions. For a label sequence $\boldsymbol{s}$, we use $\Pi(\boldsymbol{s})$ to denote the set of frame-wise sequences that are mapped to $\boldsymbol{s}$ and call $\boldsymbol{\pi}_{1:T} \in \Pi(\boldsymbol{s})$ as an alignment path of $\boldsymbol{s}$. We illustrate the relationship between the label sequence $\boldsymbol{s}$ and its possible alignment paths $\boldsymbol{\pi}_{1:T}$ in Figure 1. Now we need to find the optimal alignment path $\boldsymbol{\pi}^*_{1:T}$ as

$$\boldsymbol{\pi}^*_{1:T} = \operatorname*{arg\,max}_{\boldsymbol{\pi}_{1:T} \in \Pi(\boldsymbol{s})} p(\boldsymbol{\pi}_{1:T} \mid \boldsymbol{v}). \tag{2}$$

$\boldsymbol{\pi}^*_{1:T}$ can be efficiently searched by the dynamic time warping (DTW) algorithm [1]. Formally, to accommodate *blank* predictions in the alignment path, we first extend the label $\boldsymbol{s}$ of length N to $\boldsymbol{s'}$ of length $2N+1$ by interleaving its items with $blank$:

$$\boldsymbol{s'}_{1:2N+1} = (blank, s_1, blank, s_2, \ldots, blank, s_N, blank).$$

[1] Since there is a downsampling layer in our CSLR network, the length of the output sequence is $T/4$. We temporarily upsample it by a factor of four to match the length of input $\boldsymbol{v}$.

Algorithm 1 Find the optimal alignment path

Input: frame-wise probabilities $\boldsymbol{y}$; extended label $\boldsymbol{s}'$
Output: the most probable alignment path $\boldsymbol{\pi}^*_{1:T}$

```
for i ← 1 to 2N + 1 do                          ▷ Set the initial condition
    if i ∈ {1, 2} then
        m(1, i) = y_1(s'_i)
    else
        m(1, i) = 0
    end if
end for
for i ← 1 to 2N + 1 do                          ▷ Iterative computation
    if i = 1 then
        G(i) = {i}
    else if s'_i is blank or i = 2 or s'_i = s'_{i-2} then
        G(i) = {i − 1, i}
    else
        G(i) = {i − 2, i − 1, i}
    end if
    for t ← 2 to T do
        m(t, i) = y_t(s'_i) max_{j∈G(i)} m(t − 1, j)
    end for
end for
i ← arg max_{j∈{2N,2N+1}} m(T, j)               ▷ Backtracking
π*_T ← i
for t ← T − 1 to 1 do
    i ← arg max_{j∈G(i)} m(t, i)
    π*_t ← i
end for
return π*_{1:T} = (π*_1, ..., π*_T)
```

In order to find the optimal path by Eq. 2, we define an intermediate variable $m(t,i)$ as the probability of the optimal path associated to the first t frames of sign video $\boldsymbol{v}$ with sign sequence label $\boldsymbol{s'}_{1:i}$:

$$m(t,i) = \max_{\boldsymbol{\pi}_{1:t} \in \Pi(\boldsymbol{s'}_{1:i})} p(\boldsymbol{\pi}_{1:t} | \boldsymbol{v}), \tag{3}$$

where $p(\boldsymbol{\pi}_{1:t}|\boldsymbol{v})$ is formulated by Eq. 1. Then the probability of the optical alignment path $\boldsymbol{\pi}^*_{1:T}$ can be calculated

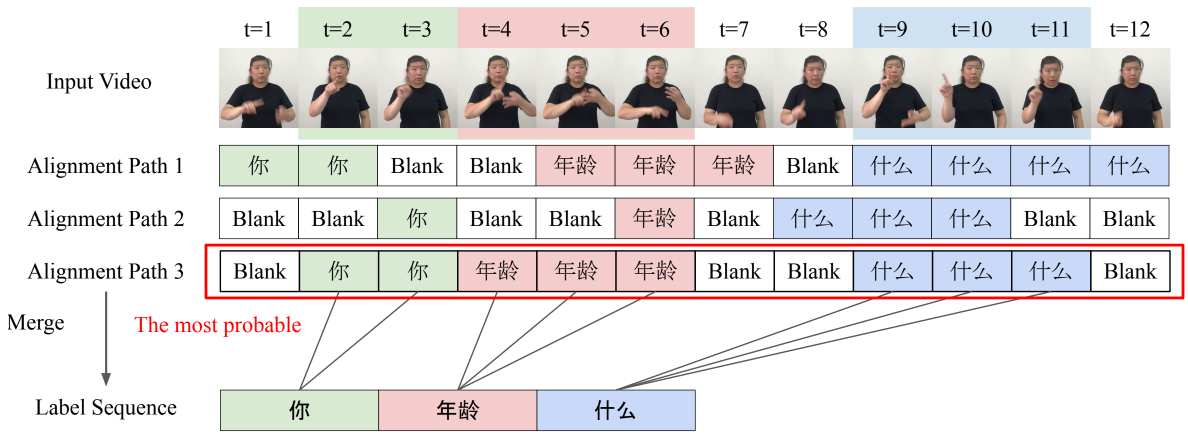


Figure 1: Illustration of partitioning a continuous sign video into the isolated sign clips. Given an input video $\boldsymbol{v}$ and its associated ground-truth sign sequence $\boldsymbol{s}$, we show three possible alignment paths (i.e. Alignment Path-1/2/3) with respect to $\boldsymbol{s}$. The probability of each alignment path can be computed by Eq. 1. The optimal alignment path is the one with the maximal probability. After removing blank predictions and deduplicating the repeated non-blank predictions from the optimal alignment path, we could partition the input video into a collection of isolated sign clips.

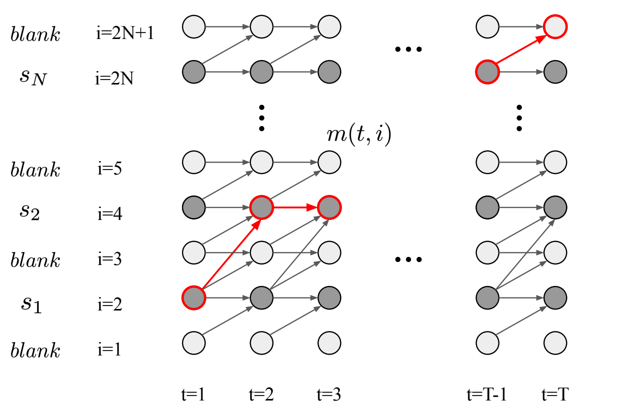


Figure 2: Illustration of the dynamic programming algorithm. Each node represents an intermediate variable $m(t,i)$ defined by Eq. 3. We iteratively compute the value of each node, as shown by the arrows. The probability of the optimal alignment path $\boldsymbol{\pi}^*_{1:T}$ is calculated by Eq. 4. After that, we could easily backtrack $\boldsymbol{\pi}^*_{1:T}$, as highlighted by the red nodes. Refer to Algorithm 1 for the whole process.

by:

$$\max_{\boldsymbol{\pi}_{1:T} \in \Pi(\boldsymbol{s})} p(\boldsymbol{\pi}_{1:T} \mid \boldsymbol{v}) = \max_{j \in \{2N, 2N+1\}} m(T, j). \quad (4)$$

Eq. 3 can be computed recursively using dynamic programming (DP) as each $m(t,i)$ is a function of several earlier values. After obtaining the result of Eq. 4, we can seek out the optical alignment path $\boldsymbol{\pi}^*_{1:T}$ that gives rise to the maximum probability by backtracking. We illustrate the computation procedure in Figure 2. The details are also formulated in Algorithm 1, which includes the initial condition, the Bellman equation for the DP algorithm, and how to backtrack the optical alignment path $\boldsymbol{\pi}^*_{1:T}$.

Dataset	Frames	Train	Dev	Test	Vocab.
Phoenix-2014 [8]	8.3	65,227	5,607	6,608	1231
Phoenix-2014T [2]	8.8	55,247	3,748	4,264	1085
CSL-Daily [11]	9.0	133,714	8,173	9,002	2000

Table 1: Statistics of the constructed isolated sign dictionaries produced by partitioning the continuous datasets (See Section 3.1). We show the average length of the segments (isolated signs), the number of segments in the Train/Dev/Test splits, and the vocabulary size for each dataset.

We find that among the estimated $\boldsymbol{\pi}^*_{1:T}$, many frames are predicted to be $blank$. For an isolated sign $s_i \in \boldsymbol{s}$ in the label sequence, if we only take the frames whose predictions in $\boldsymbol{\pi}^*_{1:T}$ are s_i as the video clip for s_i, the resulting isolated video clips may be fairly short and not encompass the entire duration of that sign. To address this issue, we adopt the following strategy to find the video segment for $s_i \in \boldsymbol{s}$. First, we find the consecutive frames whose predictions are exactly s_i in the optimal alignment path $\boldsymbol{\pi}^*_{1:T}$. Then, we expand their left and right boundaries by including more $blank$ frames whose predicted probability for s_i is the highest when the $blank$ class is excluded. This approach yields an average length of 9 frames for an isolated segment. Table 1 shows the statistics of the constructed isolated sign dictionaries.

A.2. CSLR

For continuous sign language recognition (CSLR), we re-use the architecture and training procedure of TwoStream-SLR [3] except that we add an auxiliary dataset into the training dataset. We summarize our implementations as follows.

Architecture. TwoStream-SLR [3] contains two independent sub-networks to model RGB videos and estimated keypoint sequences. The keypoints are estimated by an HRNet [9] trained on COCO-WholeBody [5]. Each of the two sub-networks is an S3D [10] backbone (only the first four blocks are used) pretrained on Kinetics-400 [6]. TwoStream-SLR also adopts bidirectional lateral connection, sign pyramid network and separate classification heads. Please refer to the original paper [3] for more details.

Training. The training of our CSLR model consists of two stages. In the first stage, we separately pre-train the SingleStreamSLR-RGB/-keypoint using a single CTC loss [4] without sign pyramid network and bidirectional lateral connection. In the second stage, we load the pre-trained SingleStreamSLR networks and train the TwoStreamSLR using the CTC loss [4] and a set of auxiliary losses proposed in [3]. In each stage, we use the Adam optimizer [7] with $\beta_1 = 0.9, \beta_2 = 0.998$, weight decay $= 1e - 3$ and a cosine learning scheduler to train the network for 40 epochs with a batch size of 8 and a learning rate of $1e - 3$. For our cross-lingual method, we mix $\mathcal{D}_{A \rightarrow P}$ and $\mathcal{D}_P$ with $\alpha = 0.2$ defined in Equation 5.

Inference. During inference, the final prediction is decoded into a sign sequence by CTC beam decoding [4]. We use a beam width of 5.

A.3. ISLR

Here we describe the architecture and training details of the isolated sign language recognition (ISLR) model we use for cross-lingual mapping.

Architecture. We adopt a TwoStream-ISLR architecture similar to the TwoStream-CSLR. The differences include: (1) the TwoStream-ISLR uses five blocks of the S3D network; (2) the sign pyramid networks are discarded; (3) a pooling layer is appended.

Training. The two S3D backbones in our TwoStream-ISLR are pre-trained on Kinetics-400 [6]. We train the whole network for 100 epochs with a batch size of 32 and a learning rate of $1e - 4$. We use the Adam optimizer [7] with $\beta_1 = 0.9, \beta_2 = 0.998$, weight decay $= 1e - 3$ and a cosine learning schedule. We adopt the label smoothing with a smoothing weight of 0.2. We pad or truncate the input segments into the length of 16 and apply augmentation including random spatial crop and random temporal sampling. We remove sign classes of frequency lower than 8 for Phoenix-2014 and Phoenix-2014T and 20 for CSL-Daily during training. This reduces their vocabulary size from 1231/1085/2000 to 428/389/981 respectively.

Inference. During inference, we evenly pad or truncate input videos to the length of 16. We forward samples of all classes to compute their cross-lingual predictions.

B. Visualization of Cross-lingual Signs

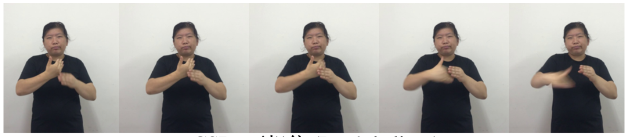
We illustrate more examples of the cross-lingual signs from CSL-Daily and Phoenix-2014T identified by our method in Figure 3, where we sort the examples by their cross-lingual prediction confidences.

First, we observe that all pairs of cross-lingual signs share similar visual cues, primarily the shape and movement of the hands. Furthermore, there appears to be a general trend where signs with higher confidence levels exhibit more detailed similarities. For example, in either Figure 3a or Figure 3b, the right hands of the two signers move similarly, while their left hands exhibit distinguishable patterns. In contrast, cross-lingual signs with confidence scores higher than 0.5, as depicted in Figure 3e-3h, not only share comparable hand orientations but also exhibit similar finger patterns and even facial expressions.

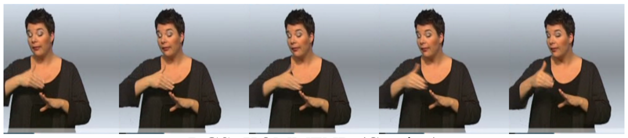
Next, cross-lingual signs usually carry distinct word meanings. For examples, “面包 (Bread)” is mapped to “KOMMEND(Coming)” and “停 (Stop)” is mapped to “MAXIMAL (Maximal)”. This demonstrates that DGS and CSL are mutually unintelligible. However, we also observe that some cross-lingual pairs convey identical meanings, *e.g.* “零 (Zero)” and “NULL(Zero)”, or close meanings, *e.g.* “下 (Down)” and “TIEF(Deep)”. This interestingly suggests that different deaf communities may share a common understanding of some semantic concepts regardless of their cultural and geographical difference and thus invent similar visual cues to convey some meanings.

C. Discussion

Limitations and Future Directions. Although our method is the first to demonstrate the effectiveness of cross-lingual transfer in CSLR, it requires both the primary dataset and the auxiliary dataset to have sequence-level annotations. Due to the limited number of labeled CSLR datasets, we are currently only able to apply our cross-lingual method to two sign languages, namely CSL and DGS. However, in the future, we aim to expand our approach to encompass a wider range of languages as more CSLR datasets become available. Additionally, we are excited to explore ways to utilize more cross-lingual data that lack labels so as to further enrich the training sources.



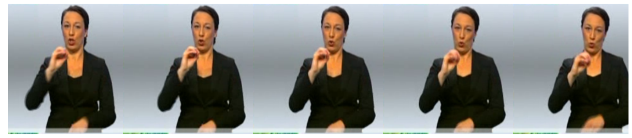
CSL: 不相信 (Don't believe)



DGS: KOMMEND (Coming)

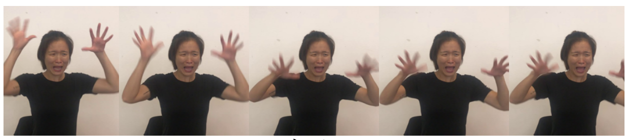
(a) Confidence: 0.1

CSL: 零 (Zero)



DGS: NULL (Zero)

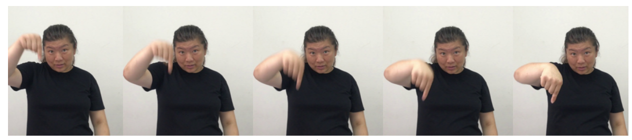
(b) Confidence: 0.2



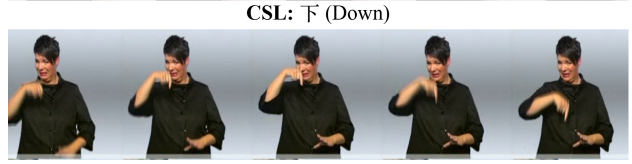
CSL: 大雨 (Storm)

DGS: REGEN (Rain)

(c) Confidence: 0.3

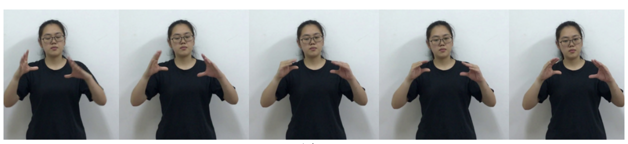


CSL: 下 (Down)



DGS: TIEF (Deep)

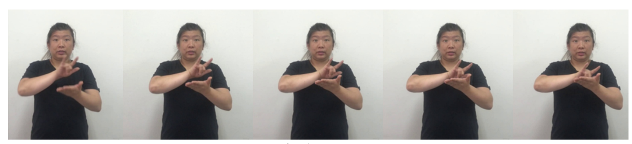
(d) Confidence: 0.4



CSL: 面包 (Bread)

DGS: NASS (Wet)

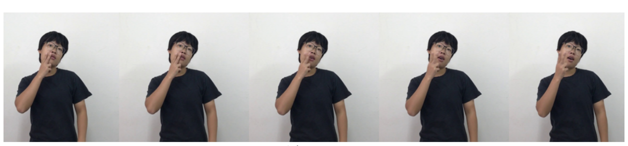
(e) Confidence: 0.5



CSL: 机场 (Airport)

DGS: MITTE (Center)

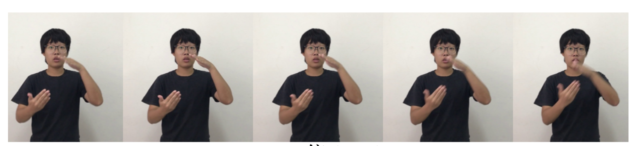
(f) Confidence: 0.6



CSL: 吸烟 (Smoking)

DGS: WIE-AUSSEHEN (How-look)

(g) Confidence: 0.7



CSL: 停 (Stop)

DGS: MAXIMAL (Maximal)

(h) Confidence: 0.8

Figure 3: We show some examples of cross-lingual signs between Chinese sign language (CSL) and German sign language (DGS) using videos from CSL-Daily and Phoenix-2014T. We sort them by the cross-lingual prediction confidence. In general, higher confidence indicates higher similarity between the signs. Cross-lingual signs usually convey distinct meanings but occasionally share the same meaning, *e.g.* both express 'zero' in Figure 3b.

Broader Impacts. With the variation in sign languages across different regions, it has been a challenge to develop recognition systems that can cater to the needs of various deaf communities. However, our findings show that despite these variations, visually similar signs can be leveraged to improve the performance of such systems. This is particularly beneficial for under-represented deaf communities that have low-resource training data. Furthermore, our work has the potential to contribute to the broader field of sign linguistics. By identifying the commonalities and differences between different sign languages, we can enhance cross-cultural communication among deaf communities.

References

[1] *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] 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*, 2018.

[3] Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. Two-stream network for sign language recognition and translation, 2022.

[4] 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.

[5] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *European Conference on Computer Vision*, pages 196–214. Springer, 2020.

[6] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, 2017.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] 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, Dec. 2015.

[9] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[10] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision*, pages 305–321, 2018.

[11] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.