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 $C$ for a dataset $D$ with an alphabet $S$. Given a training video $v = (v_1, \ldots, v_T) \in D$ containing $T$ frames and its associated ground-truth sign sequence $s = (s_1, \ldots, s_N), s_i \in S$, a well-trained CSLR model produces a frame-wise prediction sequence $y = (y_1, \ldots, y_T)$ in which $y_t \in \mathbb{R}^{|S|}$ is a probability distribution over the expanded alphabet $S' = S \cup \{\text{blank}\}$ for the $t$-th frame\(^1\). Therefore, the probability of a frame-wise sequence $\pi_{1:T} = (\pi_1, \ldots, \pi_T)$ where $\pi_t \in S'$, can be computed as

$$p(\pi_{1:T} | v) = \prod_{t=1}^{T} y_t(\pi_t), \quad (1)$$

where $y_t(\pi_t)$ indicates the probability of observing label $\pi_t$ at timestamp $t$.

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

$$\pi^*_{1:T} = \arg\max_{\pi_{1:T} \in \Pi(s)} p(\pi_{1:T} | v). \quad (2)$$

$\pi^*_{1:T}$ 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 $s$ of length $N$ to $s'$ of length $2N + 1$ by interleaving its items with blank:

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

\(^1\)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 $v$.

**Algorithm 1** Find the optimal alignment path

<table>
<thead>
<tr>
<th>Input:</th>
<th>frame-wise probabilities $y$; extended label $s'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>the most probable alignment path $\pi^*_{1:T}$</td>
</tr>
<tr>
<td>for $i \leftarrow 1$ to $2N + 1$ do &amp; $\triangleright$ Set the initial condition</td>
<td></td>
</tr>
<tr>
<td>\hspace{0.5cm} if $i \in {1, 2}$ then &amp; $\triangleright$ Iterative computation</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} $m(1, i) = y_1(s'_i)$ &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} else &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1.5cm} $m(1, i) = 0$ &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} end if &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} end for &amp;</td>
<td></td>
</tr>
<tr>
<td>for $i \leftarrow 1$ to $2N + 1$ do &amp; $\triangleright$ Backtracking</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} $i \leftarrow \arg\max_{j \in (2N, 2N+1)} m(T, j)$ &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} $\pi^*_T \leftarrow i$ &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} for $t \leftarrow T - 1$ to $1$ do &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1.5cm} $i \leftarrow \arg\max_{j \in G(i)} m(t, i)$ &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1.5cm} $\pi^*_i \leftarrow i$ &amp;</td>
<td></td>
</tr>
<tr>
<td>\hspace{1cm} end for &amp;</td>
<td></td>
</tr>
<tr>
<td>return $\pi^<em>_{1:T} = (\pi^</em>_1, \ldots, \pi^*_T)$ &amp;</td>
<td></td>
</tr>
</tbody>
</table>

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 $v$ with sign sequence label $s'_{1:t}$:

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

where $p(\pi_{1:t} | v)$ is formulated by Eq. 1. Then the probability of the optimal alignment path $\pi^*_{1:T}$ can be calculated
The most probable optical alignment path \( \pi \) is calculated by Eq. 4. After obtaining the result of Eq. 4, we can seek out 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 \( \pi_{1:T}^* \) is calculated by Eq. 4. After that, we could easily backtrack \( \pi_{1:T}^* \), as highlighted by the red nodes. Refer to Algorithm 1 for the whole process.

Algorithm 1, which includes the initial condition, the Bellman equation for the DP algorithm, and how to backtrack the optical alignment path \( \pi_{1:T}^* \). We find that among the estimated \( \pi_{1:T}^* \), many frames are predicted to be \( \text{blank} \). For an isolated sign \( s_i \in s \) in the label sequence, if we only take the frames whose predictions in \( \pi_{1:T}^* \) 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 \( s_i \in s \). First, we find the consecutive frames whose predictions are exactly \( s_i \) in the optimal alignment path \( \pi_{1:T}^* \). Then, we expand their left and right boundaries by including more \( \text{blank} \) frames whose predicted probability for \( s_i \) is the highest when the \( \text{blank} \) 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 $D_{A \rightarrow P}$ and $D_P$ 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 “NULL(Zero)”, or close meanings, “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. “ゼロ (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.
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