Cross-Modal Consistency Learning for Sign Language Recognition

Supplementary Material

A. Framework Implementation

In this section, we describe the details of the pose data extraction, the encoder architecture in the pose branch and data preprocessing.

A.1. Pose Keypoint Extraction.

We employ RTMPose-x [22] from MMPose to extract 133 whole-body keypoints, which provide detailed spatial information for subsequent pose analysis. The visualization of whole-body keypoints are shown in Figure 7. We select 75 keypoints and divide the keypoints into several sub-pose (left hand, right hand, face, mouth, and body). We select the indices for the left hand ($\{92\text{-}112\}$), right hand ($\{113\text{-}133\}$), face ($\{24\text{-}41, 54\}$), mouth ($\{84\text{-}92\}$) and body ($\{0,5,7,9,6,8,10\}$) to represent each group (denoted as $\mathcal{G}_r, r \in \{LH, RH, F, M, B\}$).

A.2. Encoder of Pose Branch.

We utilize a GCN+Transformer architecture [50] as the pose branch encoder. Specifically, four ST-GCN modules [41] are employed to extract group-specific features $\mathbf{v}_r$ from pose group $\mathcal{G}_r$, where the left and right hands share the same module for feature extraction. The extracted 512 dimension features are then concatenated into manual features $\mathbf{v}_{man}$ and non-manual features $\mathbf{v}_{non}$:

$$\mathbf{v}_{man} = \text{Concat}([\mathbf{v}_{LH}, \mathbf{v}_{RH}, \mathbf{v}_B]), \tag{15}$$

$$\mathbf{v}_{non} = \text{Concat}([\mathbf{v}_F, \mathbf{v}_M]), \tag{16}$$

where $\mathbf{v}_{LH}$, $\mathbf{v}_{RH}$, $\mathbf{v}_B$, $\mathbf{v}_F$, and $\mathbf{v}_M$ represent features from left hand, right hand, body, face, and mouth, respectively. To model temporal relationships, two separate 3-block 8-head Transformers ($\mathcal{T}_{man}$ and $\mathcal{T}_{non}$) are leveraged to process these features and generate the final pose embedding:

$$\mathbf{v} = \text{Concat}([\mathcal{T}_{man}(\mathbf{v}_{man}), \mathcal{T}_{non}(\mathbf{v}_{non})]), \tag{17}$$

where final pose embedding $\mathbf{v}$ is 1024 dimensions.

A.3. Data Preprocessing.

For data preprocessing, we select aforementioned 75 keypoints for pose input per frame and resize all RGB frames to 224×224. For computational efficiency, we uniformly sample $T = 32$ frames from each sequence.

A.4. Data Augmentation

For the query and key samples, we ensure consistency in hyperparameters during data augmentation while performing the process independently. For random temporal cropping, we first apply continuous temporal cropping to the input sequence, randomly extracting a clip from the interval $[lT, T]$ frames. Subsequently, we uniformly sample a fixed number of K frames from the cropped interval. In our approach, we set $l = 0.1$ and $K = 64$.

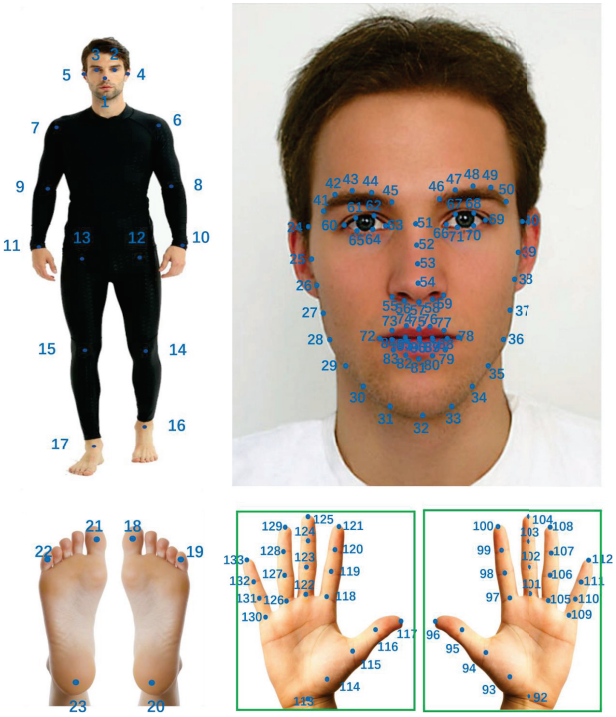


Figure 7. The visualization of the whole-body 133 keypoints from [23].

B. Visualization of Motion-Preserving Masking

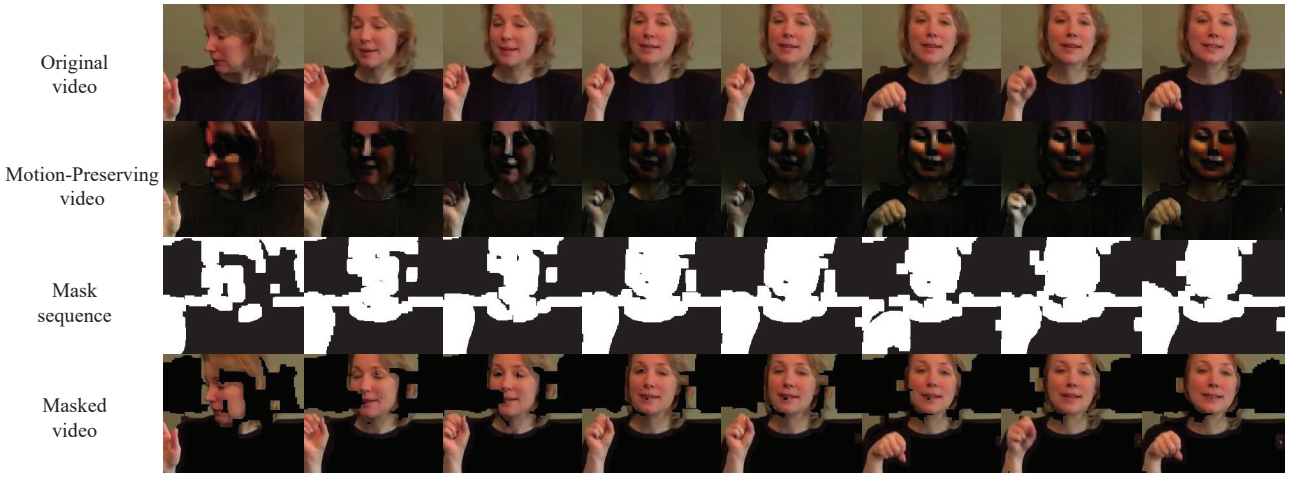
As shown in Fig. 8 and Fig. 9, the generated motion-preserving videos effectively suppress static regions while highlighting motion areas. Furthermore, since motion-preserving videos significantly alter the pixel distribution of the original videos, we further generate binary motion-preserving mask sequences from the motion-preserving videos. These mask sequence are applied to mask original videos, explicitly mitigating static information redundancy in RGB modality. The visualizations indicate that MPM preserves semantically informative regions such as hand and facial features, while effectively suppressing semantically irrelevant areas including clothing textures and background elements, enhancing the cross-modal feature alignment between pose and RGB representations.

(a) Label: good

(b) Label: understand

Figure 8. The visualizations of motion-preserving masking.

(a) Label: wow



(b) Label: yes

Figure 9. The visualizations of motion-preserving masking.