SignRep: Enhancing Self-Supervised Sign Representations

Supplementary Material

A1. Human Pose Extraction

To extract human pose features, we utilize angles derived from a human pose estimation model from [21]. We compute the bone lengths for all instances in the YouTube-SL-25 dataset (YT-SL) and select the median value as the standard bone length for each respective joint. This normalization ensures that all individuals are represented with the same body shape, thereby avoiding the leakage of person-specific features when converting angles into 3D keypoints.

We visualize the resulting keypoints in Fig. 2, separating the hands from the body for easier identification of indices. The left fingertips are defined using keypoint indices $\{44, 48, 52, 56, 60\}$. For the fingertip distance matrix, $\mathcal{P}^{\{b,d\}}$, these keypoints serve as the source, while indices $\{40, 41, 44, 45, 48, 49, 52, 53, 56, 57, 60\}$ are used as the destination for computing the distance matrix. Similarly, the same process is applied to the right hand using its respective keypoint indices.

For the hand-interaction distance prior, $\mathcal{P}^{\{b,d\}}$, we use the fingertip keypoint indices and the wrist keypoint ($\{40\}$ for left wrist and $\{19\}$ for right wrist) as the source. The destination includes the set of keypoint indices $\{0, 3, 6, 7, 10, 13, 15, 16, 17, 18, 19, 23, 27, 31, 35, 39, 40, 44, 48, 52, 56, 60\}$, which represent hands, face and body components. This matrix captures the distances between key positions involved in interactions between the hands and the rest of the body.

Human pose estimations often exhibit jitter across frames, which can affect temporal consistency. To mitigate this effect on the signer activity prior, $\mathcal{P}^{\{h,\text{act}\}}$, we determine whether a hand is inactive by checking two conditions: (1) its position is below the y-axis mean of keypoints $\{0, 3, 6, 7\}$, and (2) the sum of the standard deviations across time for all 21 visible hand keypoints is less than 0.26. These criteria help identify inactive hands in the presence of keypoint jitter.

A2. Pretraining Dataset Processing

For pretraining, we utilize the YT-SL dataset. We rely on pose estimations to ensure that a signer is present in each sequence, cropping the video to focus on the upper torso before resizing it to 256×256.

To prevent data leakage, since WLASL also contains YouTube videos, we ensure there is no overlap between the videos in the WLASL and YT-SL datasets. This is achieved by comparing the video IDs from WLASL with those in the YT-SL dataset, ensuring that no videos that are in the WLASL dataset are in our YT-SL pretraining data.

During pretraining, we randomly select 16 consecutive frames from each video. For each batch, we randomly select two sequences from the same video, ensuring that each batch contains a matching pair for the discriminator. These steps are then used to train the SignRep framework.

A3. Implementation Details

As described in Sec. 7.2, we initialize the pretraining of our SignRep framework using the video Hiera Base model, pretrained with MAE on Kinetics. The output dimension D is 768, with a drop path rate of 0.1. The sign decoder's upsampler has a hidden dimension of 512 and the output dimension D' is set to 384.

Pretraining. During training, data augmentations include Planckian Jitter [56], random resized cropping from 256×256 to 224×224, Gaussian blur and grayscale conversion. The model is trained for 500,000 iterations with a batch of 20 and a masking ratio of 80% on a single NVIDIA 3090 GPU. A warmup over the first 50,000 iterations gradually increases the learning rate to 1×10^{-4} using the Adam optimizer [27], followed by cosine annealing decay. A layer-wise learning rate decay [9] is applied with a factor of 0.85.

In Tab. 9, we list the hyperparameters used for the weighting of the loss functions during pretraining. Additionally, we apply a scaling factor ψ to the target to balance the target values.

Downstream Recognition. We use the same data augmentation as pretraining and apply cross-entropy loss with label smoothing of 0.1, with no patch masking applied, setting κ to 0.2 for the class distribution loss. The model is trained with a batch size of 8 for 100 epochs, with 1000 iterations of warmup, followed by cosine annealing of the learning rate, with a max learning rate of 1×10^{-4} using the Adam optimizer. The layer-wise learning rate decay factor is 0.85.

For the Adam optimizer, we utilize the AdamW version in Pytorch. We set the betas to $(0.9, 0.95)$ and use a weight decay of 0.5. To stabilize training, gradient clipping is applied with a maximum value of 1.0. During pretraining, the model is evaluated with retrieval on WLASL validation set every 25,000 iterations, the model achieving the best performance on the retrieval task using the WLASL validation set is selected for subsequent retrieval, recognition and translation tasks.

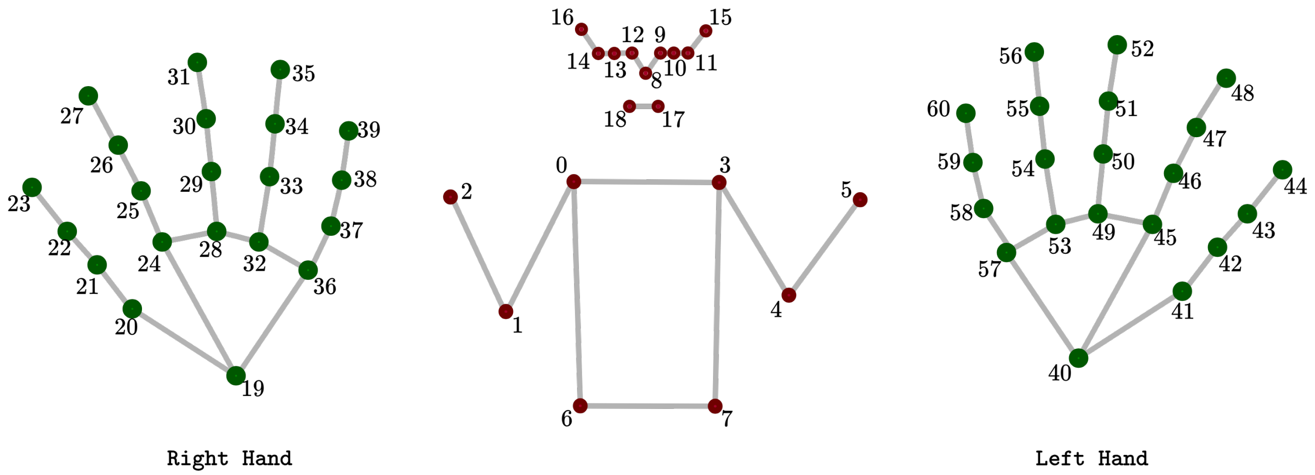

Figure 2. Visualization of 3D keypoint extracted. Numbers alongside the nodes represent the keypoint indices. For visualization purposes, we separate the left and right hand from the body.

Loss Components	**weighting** w	**scale** ψ
Priors		
body angles ($w_{\mathcal{P}\{b,a\}}$)	10.0	1.0
left hand angles ($w_{\mathcal{P}\{LH,a\}}$)	10.0	1.0
right hand angles ($w_{\mathcal{P}\{RH,a\}}$)	10.0	1.0
body kpt. ($w_{\mathcal{P}\{b,k\}}$)	10.0	1.0
left hand kpt. ($w_{\mathcal{P}\{LH,k\}}$)	10.0	2.0
right hand kpt. ($w_{\mathcal{P}\{RH,k\}}$)	10.0	2.0
body dist. ($w_{\mathcal{P}\{b,d\}}$)	20.0	1.0
left hand dist. ($w_{\mathcal{P}\{LH,d\}}$)	20.0	4.0
right hand dist. ($w_{\mathcal{P}\{RH,d\}}$)	20.0	4.0
signer activity ($w_{\mathcal{P}\{\text{act}\}}$)	0.2	-
Regularizations		
variance (w_{var})	1.0	-
covariance (w_{cov})	0.004	-
adversarial style (w_{adv})	2.0	-

Table 9. Hyperparameters for weighting factors for the different loss components used during pretraining of SignRep.

Downstream Translation. For the downstream translation task, we use Phoenix14T, CSL-Daily and How2Sign. Phoenix14T [6] is a German Sign Language (DGS) dataset consisting of weather forecast broadcasts with aligned sign and text translations. CSL-Daily [54] is a daily conversational Chinese Sign Language dataset recorded in a lab setting, covering various everyday interaction topics such as family life, shopping, travel and banking services. How2Sign [13] is an American Sign Language (ASL) dataset that provides parallel signed video and text translations of instructional videos across a broad range of categories.

For a fair comparison, we use the open-source code from [49] for Phoenix14T and CSL-Daily and follow [43] for How2Sign, applying the same hyperparameters specified in their respective papers. This ensures that improvements stem from our learned representations rather than differences in training configurations.

A4. Discriminator Setup

In Sec. 5.2, the discriminator determines whether the output features z^{avg} share the same style as a given style representation z^{style}. This process ensures that the representation encoder f_{enc} learns style-agnostic representations, for robust and generalizable features.

The discriminator model is designed as a lightweight MLP-based architecture. To address the relatively small magnitude of the style representation values, z^{style}, we first scale these values by a factor of 100.0. The scaled style representation is then passed through a two-layer MLP with a hidden size of 768, which transforms it to match the dimensionality of z^{avg}. Layer normalization is applied after this transformation. Next, the transformed z^{style} is concatenated with z^{avg} and fed into a four-layer MLP with a hidden size of 768 and an output size of 1. This MLP is responsible for determining whether the representation of z^{avg} aligns with the style z^{style}. Spectral normalization is incorporated into this final MLP to stabilize discriminator training. All linear layers, except the final linear layer, are followed by the GELU activation function.

Matched and unmatched style samples for training the discriminator are constructed from items within the batch. For each item in the batch, its matching styles are derived from its paired sample described in Sec. A2, while unmatched pairs are randomly selected from the remaining batch items. This setup ensures that the discriminator learns to distinguish between matching and non-matching styles effectively.

The discriminator is trained using binary cross entropy loss to predict 0's for unmatched styles and 1 for matched styles. We use a learning rate of 1×10^{-4}, with a warm-up period of 50,000 iterations and cosine annealing decay. The Adam optimizer is used with betas $(0.5, 0.9)$ and a weight decay of 1×10^{-3}. An exponential moving average with an update momentum of 0.1 is used to compute the expected outputs of a matched style $\mathbb{E}_{q \sim \mathrm{M}} \mathfrak{D}(q)$ and unmatched style $\mathbb{E}_{q \sim \mathrm{U}} \mathfrak{D}(q)$. The discriminator is trained simultaneously with the SignRep representation model.

A5. Class Probability Distribution

To create the class distribution ϕ, we utilize the temperature-scaled distribution described in Sec. 6. Our goal is to avoid excessively weak low-confidence probabilities and overly strong high-confidence probabilities, thereby achieving a smoother loss function $\mathcal{L}_{\phi}$.

For each class, we select a temperature τ such that the scaled distribution $\texttt{softmax}(\hat{\phi}_c/\tau)$ yields a maximum class probability as close as possible to, but still below, 0.5. Here, $\hat{\phi}_c$ represents the inter-class cosine similarity for class c. We determine the appropriate τ by iterating over values in the interval $[0.001, 0.1]$ and selecting the temperature that produces ϕ_c satisfying $\texttt{max}(\phi_c) < 0.5$ while being nearest to 0.5. We repeat this process for every class to obtain the final class distribution ϕ.

A6. Inflated Patch Embeddings

To accommodate a 64-frame input without increasing the number of tokens processed during the downstream recognition task, we employ *inflated patch embeddings* as described in Sec. 7. This method preserves computational efficiency while capturing temporal relationships in the data. The pretraining is conducted on continuous sign data, whereas the downstream task involves isolated signs, which are temporally less dense. To address this discrepancy, we adapt the patch embeddings by inflating their temporal components, ensuring the preservation of temporal relations.

The original patch embeddings are defined with a kernel size of $(3, 7, 7)$, a stride of $(2, 4, 4)$, and padding of $(1, 3, 3)$. These parameters are updated to a kernel size of $(7, 7, 7)$, a stride of $(8, 4, 4)$, and padding of $(3, 3, 3)$. This adjustment allows for better modeling of the temporal relationships required for sign recognition without adding more patch tokens.

To ensure compatibility and preserve the pretrained weights, we employ a zero-initialization approach. The new kernel weights are first initialized to zero. Then, weights from the original patch embedding are mapped to the new kernels by transferring the weights from kernel indices $\{0, 1, 2\}$ to indices $\{1, 3, 5\}$ in the temporal dimension, respectively. This method ensures that the pretrained information is preserved during downstream initialization.

A7. Qualitative Retrieval

We show qualitative results of the pretrained SignRep model on the three downstream recognition datasets, ASL-Citizen in Fig. 3, NMFs-CSL in Fig. 4 and WLASL in Fig. 5. We note that the retrieved results are generated using the pretrained model, which has neither been fine-tuned on the downstream recognition task nor exposed to the downstream video dataset during pretraining. We display the top-3 closest retrieved video segments for randomly selected reference video segments with active signers. The results show that the model effectively retrieves segments with similar hand shapes, poses and motions, highlighting its ability to capture meaningful sign-related features during pretraining.

A8. Limitations

Our model is pretrained on Youtube-SL-25, which carries inherent limitations in terms of signer diversity, language distribution and skin tone representation. These factors may affect the quality and generalizability of the learned representations. Additionally, our method focuses solely on manual sign features, leaving room for future improvements by incorporating non-manual components such as facial expressions and mouthing patterns. While our approach eliminates the need for keypoints during downstream tasks, the pretraining process still relies on keypoint-based supervision, which may be affected by low-quality detections. To mitigate this, we leverage a human pose estimation model specialized for sign language [21]. Furthermore, we filter out keypoints with confidence scores below 50% and mask missing keypoints in the loss function. These adjustments are advantageous over methods relying on keypoints as inputs.

Our model learns individual sign representations using a 16-frame window. Future work could explore extending this to longer temporal windows. However, doing so would require careful modifications to prevent excessive computational overhead, as increasing the number of frames also increases token complexity. Alternatively, our model can serve as a lightweight feature extractor for learning inter-sign relationships and long-range temporal dependencies in a more efficient manner.

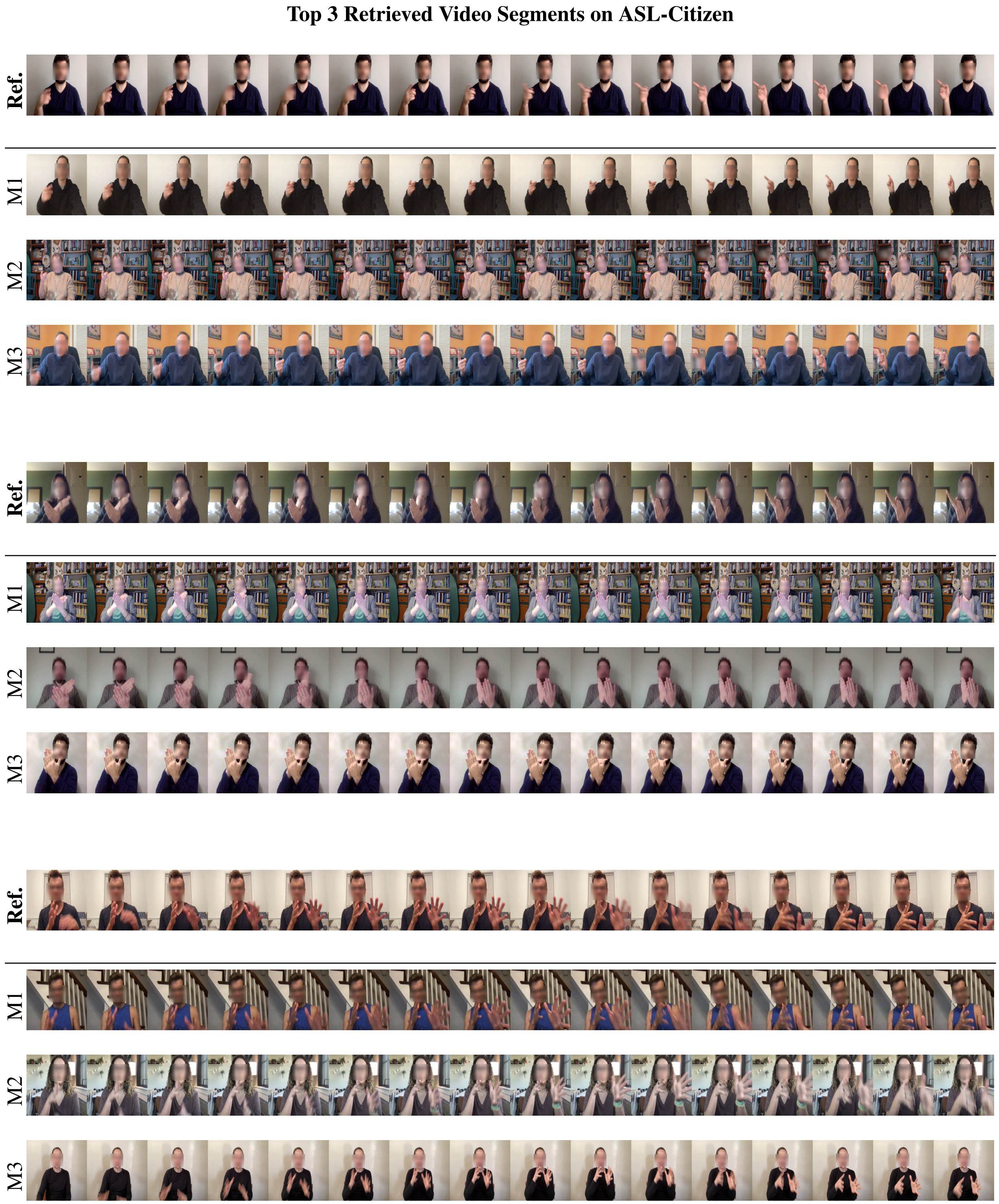

Figure 3. Qualitative results for ASL-Citizen for retrieval based on features extracted from the pretrained SignRep. Given the reference sequence (Ref.), the Top-3 most similar videos are retrieved based on the cosine similarity of the output representations. M1 denotes the closest match, M2 is the second closest match and M3 is the third closest match.

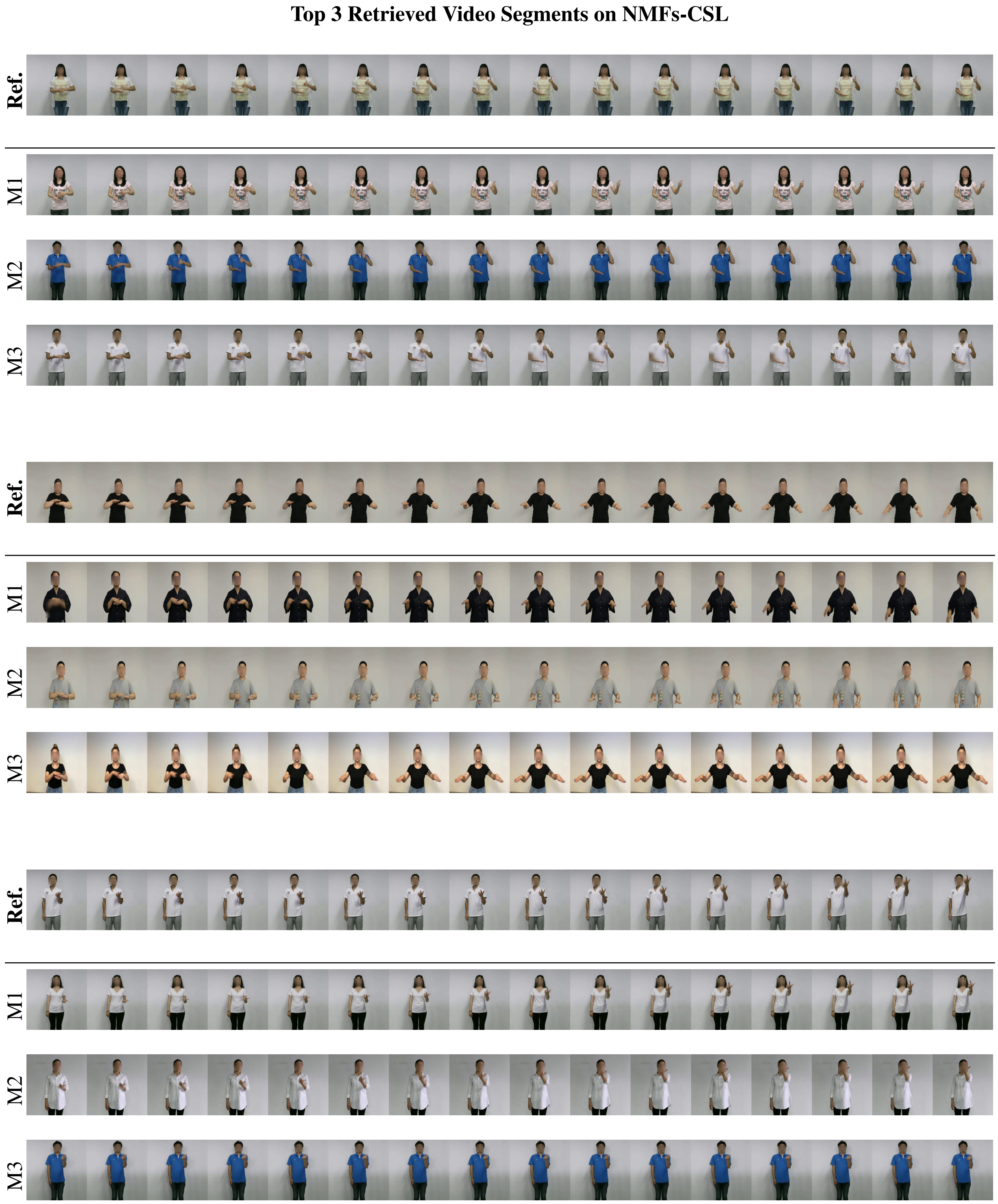

Figure 4. Qualitative results for NMFs-CSL for retrieval based on features extracted from the pretrained SignRep. Given the reference sequence (Ref.), the Top-3 most similar videos are retrieved based on the cosine similarity of the output representations. M1 denotes the closest match, M2 is the second closest match and M3 is the third closest match.

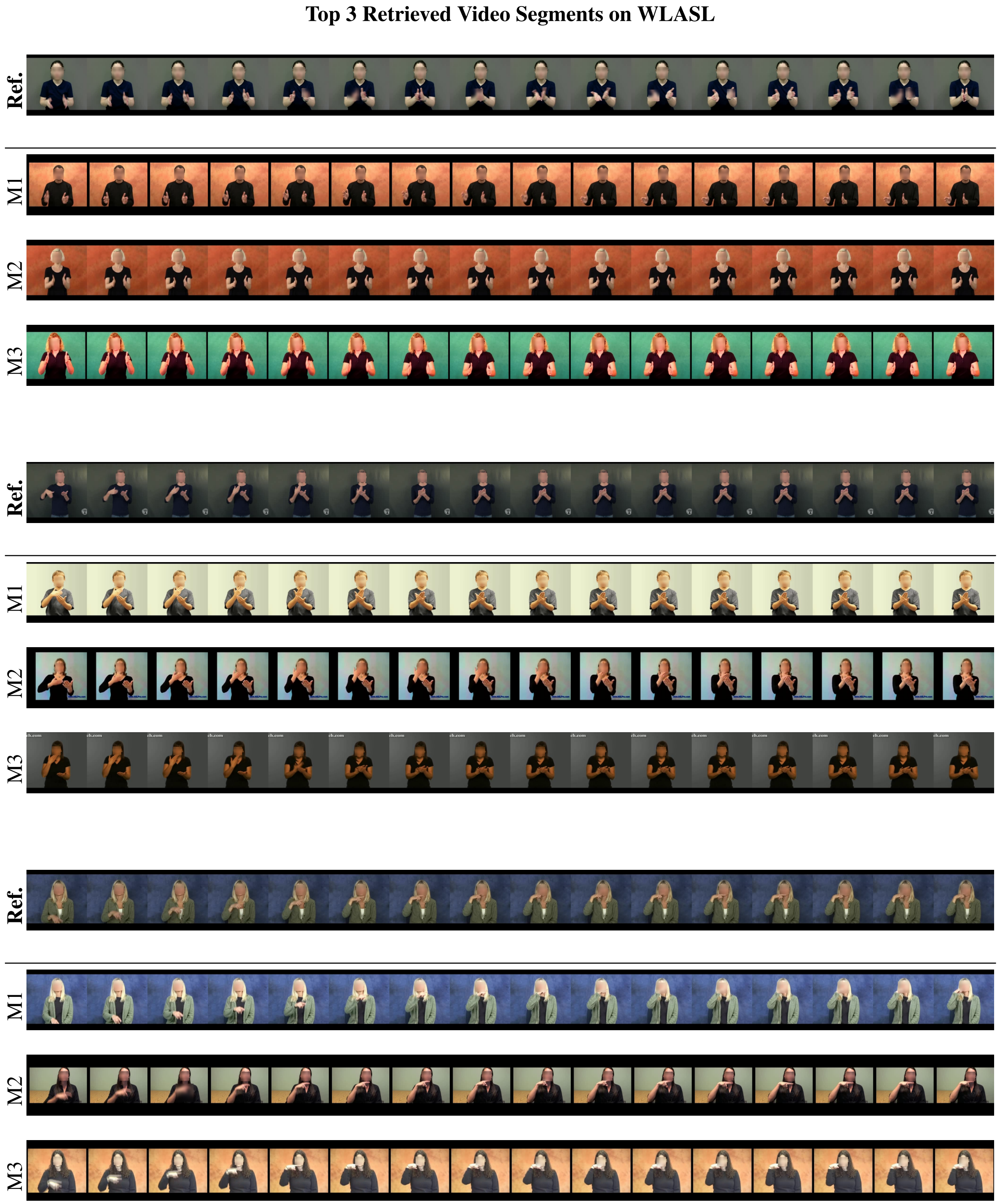

Figure 5. Qualitative results for WLASL for retrieval based on features extracted from the pretrained SignRep. Given the reference sequence (Ref.), the Top-3 most similar videos are retrieved based on the cosine similarity of the output representations. M1 denotes the closest match, M2 is the second closest match and M3 is the third closest match.