SignRep: Enhancing Self-Supervised Sign Representations

Ryan Wong[1], Necati Cihan Camgoz[2], Richard Bowden[1]
[1]University of Surrey, [2]Meta Reality Labs
{r.wong, r.bowden}@surrey.ac.uk, neccam@meta.com

Abstract

Sign language representation learning presents unique challenges due to the complex spatio-temporal nature of signs and the scarcity of labeled datasets. Existing methods often rely either on models pre-trained on general visual tasks, that lack sign-specific features, or use complex multimodal and multi-branch architectures. To bridge this gap, we introduce a scalable, self-supervised framework for sign representation learning. We leverage important inductive (sign) priors during the training of our RGB model. To do this, we leverage simple but important cues based on skeletons while pretraining a masked autoencoder. These sign specific priors alongside feature regularization and an adversarial style agnostic loss provide a powerful backbone. Notably, our model does not require skeletal keypoints during inference, avoiding the limitations of keypoint-based models during downstream tasks. When finetuned, we achieve state-of-the-art performance for sign recognition on the WLASL, ASL-Citizen and NMFs-CSL datasets, using a simpler architecture and with only a single-modality. Beyond recognition, our frozen model excels in sign dictionary retrieval and sign translation, surpassing standard MAE pretraining and skeletal-based representations in retrieval. It also reduces computational costs for training existing sign translation models while maintaining strong performance on Phoenix2014T, CSL-Daily and How2Sign.

1. Introduction

Sign language is an important means of communication for millions of people worldwide. Sign languages have complex visual characteristics, which include intricate hand shapes, motions, body poses and facial expressions that models need to accurately interpret and process [5]. Furthermore, the computational demands of processing long video sequences add considerable challenges, making it difficult to scale these systems effectively. As a result, current approaches to sign recognition and translation often rely on general pretrained vision models [1, 7, 19, 22, 28, 49, 57]. Creating robust, label-free sign language representations that generalize across diverse datasets is challenging, yet essential for scalable sign language modeling.

Most existing methods for sign recognition rely on multimodal or an ensemble of specialized models to achieve state-of-the-art recognition results. These models often require multi-channel inputs (e.g. RGB, depth and skeleton data) or specialized architectures to capture the complex interactions of hand, body or facial expressions [17, 20, 22, 23, 52, 57]. With each country having its own sign language and linguistic study expensive, currently available labeled datasets are sparse with typically under 2000 unique signs [1, 19, 26, 28, 41]. Collecting annotated sign language data is costly and time-consuming, making it infeasible to rely solely on labeled datasets for sign representation learning. This highlights the need for methods that can learn sign language representations from large-scale, unlabeled data.

Newer methods for pretraining sign language translation models often require sentence-aligned annotations [25, 37, 49, 53, 55]. While these prior approaches jointly learn individual sign representations and inter-sign relationships, they are not scalable to unlabeled datasets as they require sentence-aligned annotations. We therefore focus on building a scalable and generalizable individual sign representation framework using self-supervised learning. A strong foundation in individual sign embeddings is an important step before building effective inter-sign models, as it ensures robust feature representations before incorporating complex long range temporal information.

Our main contributions are as follows: (1) We propose a scalable self-supervised Masked Autoencoding (MAE) framework which leverages sign priors, adversarial loss and feature regularizations for sign representation learning. (2) We introduce a method for analyzing sign class similarities in unseen datasets, introducing an auxiliary class probability distribution loss, which enhances recognition performance. (3) Our single pretrained model achieves state-of-the-art sign recognition, surpassing complex architecture and multimodal models. (4) We demonstrate the effectiveness of our sign representations for sign dictionary retrieval, achieving strong performance without any downstream training. (5) We demonstrate that using our pre-

trained model as a frozen feature extractor, training is more tractable by reducing memory requirements and improving the performance of existing sign translation models.

2. Related Work

Supervised Sign Recognition. Isolated Sign Recognition involves identifying a single sign within a given video. Sign recognition models often rely on pretrained spatio-temporal models from action recognition datasets such as Kinetics [8], fine-tuned for specific sign recognition tasks [1, 19, 22, 28, 57]. This transfer learning approach faces a domain shift problem, as sign videos have unique temporal dynamics and subtle gestures not well-represented in general action datasets, leading to performance degradation. To leverage invariance, some methods use skeletal keypoints instead of full-frame RGB models [10, 17, 20, 48, 52]. While keypoint-based models are more memory-efficient, they typically underperform relative to RGB models [2, 11, 36] and require complex architectural modifications to Graph Convolutional Networks (GCNs) [30] or transformers [46], to model the spatial-temporal relationships. Moreover, keypoints are prone to errors, such as missing or misdetected points [31, 32]. To solve these issues, recent state-of-the-art methods have combined keypoint and RGB modalities through branching or ensembling techniques [22, 23, 57]. While they achieve higher accuracy, they increase computational complexity due to the need for multiple models or support for multi-modal inputs.

Self-Supervised Sign Representation Learning. Self-supervised learning for vision [3, 33, 38, 44] and language [12, 35] has demonstrated substantial benefits by leveraging unlabeled datasets for various tasks. In sign language, approaches such as Skeletor [24], SignBERT [17], SignBERT+ [20] and BEST [52] focus on keypoint-based self-supervised learning, employing masked learning similar to BERT [12]. However, these methods depend on keypoints as inputs and therefore requiring ensembling with RGB models to achieve competitive performance, resulting in increasing complexity and limited pretraining benefits. This highlights the need for a simple and efficient model which capturing sign-specific information and retains the pretraining advantages.

Paired Sign-Text Pretraining for Sign Translation. Supervised pretraining approaches for sign translation, such as GFSLT-VLP [53], Sign2GPT [49], SignHiera [37], MSLU [55] and VAP [25], use sign video–spoken language pairs to enhance sign translation. Sign2GPT employs a pretrained DinoV2 [33], which is effective for general vision tasks but requires LoRA [16] fine-tuning for sign translation to achieve strong performance, significantly increasing computational costs. This highlights the domain gap between pretrained foundation vision models and sign language. SignHiera [37] demonstrates success with large-scale training on YT-ASL [45] using Hiera MAE [38] with language-supervised pretraining. However, they acknowledge that SignHiera requires significant computational resources (64 A100 GPUs for two weeks), emphasizing the need for more accessible and cost-effective video pretraining on large-scale datasets. These methods also face scalability challenges due to their reliance on sentence-aligned annotations. In this paper, we introduce a scalable self-supervised pretraining framework that learns sign representations without paired sign-text annotations, aiming to significantly reduce computational costs for pretraining and improve efficiency in downstream sign tasks.

3. Sign Priors

Sign language communication relies on hand shapes, body posture and interactions. To guide the model toward meaningful sign representations, we introduce "sign priors", a set of cues that we know capture essential sign features. These priors ($\mathcal{P}$) serve as the primary targets for our model, as detailed in Sec. 5.1. Using a human pose estimation model specialized for sign language videos [21], we extract joint angles and 3D keypoints. We categorize our priors into **keypoint**, **angle**, **distance** and **signer activity**.

Keypoint Priors define the spatial structure of signing, we divide this into two priors:

Hand Keypoint Prior ($\mathcal{P}^{\{h,k\}}$): This prior is designed to capture hand shapes and orientations. We normalize the 3D coordinates of the 21 hand keypoints by setting the wrist as the origin, thereby eliminating positional variations in the body space. The resulting vector, $\mathcal{P}^{\{h,\mathrm{k}\}} \in \mathbb{R}^{21\times 3}$, represents the configuration of the left or right hand, where $h \in \{\mathrm{LH}, \mathrm{RH}\}$ denotes the left or right hands.

Full Body Keypoint Prior ($\mathcal{P}^{\{b,k\}}$). This prior captures the positioning of the body and hands within the body space. We use all 61 3D keypoints (21 for each hand and 19 for the body), resulting in $\mathcal{P}^{\{b,\mathrm{k}\}} \in \mathbb{R}^{61\times 3}$.

Joint Angle Priors capture finger flexion, extension and encode upper-body posture variations:

Hand Joint Angle Prior ($\mathcal{P}^{\{h,a\}}$). This prior captures hand joint orientations. We extract 41 hand joint angles, yielding a vector $\mathbb{R}^{41\times 1}$. To handle the continuous nature of angles, we apply sine and cosine transformations, resulting in the final angle prior $\mathcal{P}^{\{h,\mathrm{a}\}} \in \mathbb{R}^{41\times 2}$.

Body Angle Pior ($\mathcal{P}^{\{b,a\}}$). To capture the body's joint orientations, we extract the 22 body joint angles. These angles are transformed using sine and cosine functions to handle their continuous nature, resulting in the body angle prior $\mathcal{P}^{\{b,\mathrm{a}\}} \in \mathbb{R}^{22\times 2}$.

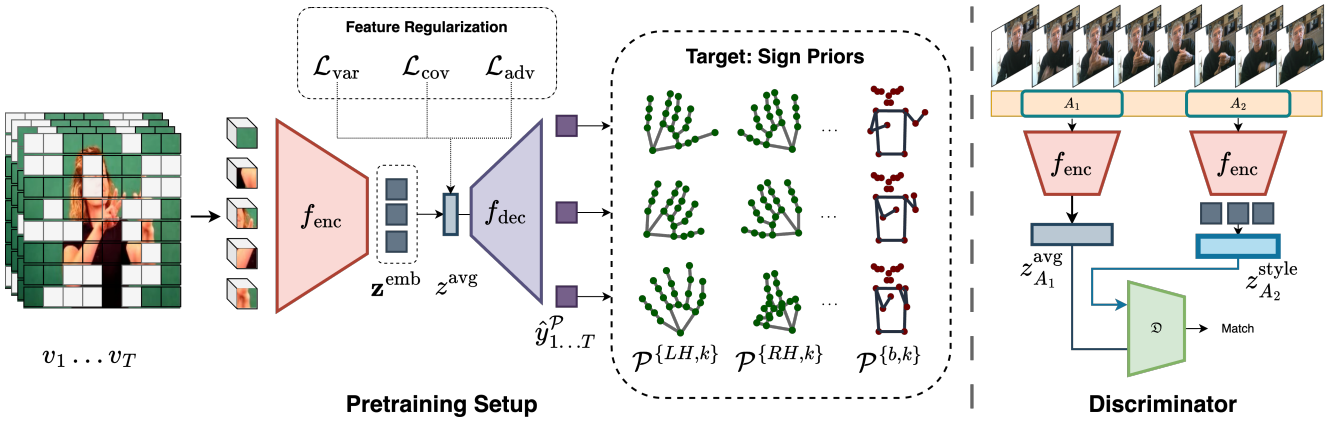

Figure 1. (Left): The pretraining process for SignRep, which leverages masked representation learning to predict sign priors such as hand keypoints and joint angles. This is achieved through a Hiera encoder and a lightweight sign decoder. The representation is further refined with regularization losses, including variance, covariance and adversarial style loss. (Right): An example setup for the discriminator to obtain a representation pair to predict a style-representation match.

Distance Priors capture fine-grained differences between signs and the interactions of hands in the signing space:

Fingertip Distance Prior ($\mathcal{P}^{\{h,d\}}$). Since fingertip interactions are a key component in sign language, we compute a distance matrix by measuring the distances from the fingertip keypoints to each of the knuckle, wrist and other fingertip keypoints. This produces the prior $\mathcal{P}^{\{h,\text{d}\}} \in \mathbb{R}^{5\times11\times3}$.

Hand-Interaction Distance Prior ($\mathcal{P}^{\{b,d\}}$). Similar to the fingertip distance prior, we calculate a distance matrix using the wrist and five fingertip keypoints from each hand, resulting in a 6×2 set of keypoints. The distance between the hands, body and facial keypoints is computed to capture the interactions between the hands and other body parts, resulting in the final hand-interaction prior $\mathcal{P}^{\{b,\text{d}\}} \in \mathbb{R}^{12\times22\times3}$.

Signer Activity. Some signs require only one hand or the signer may be in a resting pose. To enable the model to capture this information, we develop a simple heuristic. We define the active prior $\mathcal{P}^{\{h,\texttt{act}\}}$, where a hand is considered inactive if it remains below the middle of the stomach and has not moved over the course of a video clip. This results in a prior $\mathcal{P}^{\{h,\texttt{act}\}} \in [0, 1]$, indicating whether each hand is active (1) or inactive (0).

4. SignRep Architecture

To learn the sign priors, we introduce a novel adaptation of the Hierarchical Vision Transformer (Hiera) [38], designed with a pretraining task specifically tailored for sign representation learning without requiring labeled sign data. We choose the Hiera model as it aligns with our objective of being efficient, simple and effective. It has demonstrated strong performance in masked representation learning while eliminating the need for complex architectures and specialized modules. We aim to develop a single spatio-temporal sign model that enhances performance without relying on multi-branch or ensemble methods, instead improving the representation learning process.

The standard Hiera MAE framework processes a video clip represented as $V = \{v_1, v_2, \ldots, v_t, \ldots, v_T\}$, where v_t denotes the frame at time step t, and T is the total number of frames. The video is first divided into spatiotemporal patches. A masking strategy is then applied, with a subset of patches randomly masked according to a masking ratio M. The unmasked patches are fed into the encoder f_{enc}, which processes them to generate an output representation $\mathbf{z}^{\text{emb}} = \{z_0^{\text{emb}}, z_1^{\text{emb}}, \ldots, z_K^{\text{emb}}\}$, where K is the number of output tokens from the encoder. The representation $\mathbf{z}^{\text{emb}}$ is then passed to a decoder, along with learnable masked tokens as input. The pretraining objective is to reconstruct the masked patches through pixel-level prediction, allowing the model to learn contextual relationships and enhance the visual representation.

Rather than pixel reconstruction, we use sign priors to learn meaningful sign representations, replacing the pixel reconstruction decoder with a lightweight sign decoder.

Sign Decoder (f_{dec}). As shown in Fig. 1, instead of passing $\mathbf{z}^{\text{emb}}$ to the decoder, we take the average across $\mathbf{z}^{\text{emb}}$ tokens, which is then processed through layer normalization followed by a fully connected layer. This produces an output representation $z^{\text{avg}} \in \mathbb{R}^{1\times D}$, where D is the dimensionality of the representation. The representation z^{avg} is then upsampled temporally to match the sequence length T of the input video. The upsampling module is implemented using a lightweight network consisting of a 1D convolution with a kernel size of 1, followed by a GELU activation function, and then a transpose convolution with a kernel size of T to match the number of input frames and

an output dimension of D'. This results in an upsampled vector $\mathbf{z}^{\text{up}} \in \mathbb{R}^{T \times D'}$, where $\mathbf{z}^{up} = [z_1^{up}, z_2^{up}, \ldots, z_T^{up}]$.

We can then incorporate prediction heads for each sign prior to the network as follows:

$$\hat{y}_t^{\mathcal{P}} = g_{\mathcal{P}}(z_t^{\text{up}}) \tag{1}$$

where each sign prior, $\mathcal{P} \in \{\mathcal{P}^{\{h,k\}}, \mathcal{P}^{\{h,a\}}, \mathcal{P}^{\{h,d\}}, \mathcal{P}^{\{b,k\}}, \mathcal{P}^{\{b,a\}}, \mathcal{P}^{\{b,d\}}\}$ has a corresponding fully connected layer ($g_{\mathcal{P}}$) with an associated output $\hat{y}_t^{\mathcal{P}}$ which matches the flattened dimension of the corresponding sign prior. For the activity prior we simply use an MLP layer using z^{avg} as input which produces an output of 2 to match the dimension of $[\mathcal{P}^{\{\text{LH,act}\}}, \mathcal{P}^{\{\text{RH,act}\}}]$.

The decoder is completely removed during downstream tasks and only the encoder is used as the sign representation model. As a result, the target sign priors are not required during downstream tasks, allowing the use of a single model that inherently captures sign knowledge. This approach ensures computational efficiency by leveraging masked representation learning while directly learning the essential sign priors.

5. Sign Representation Objectives

5.1. Sign Priors Reconstruction

In the previous section, we identified the sign priors that our model needs to learn. Since the sign decoder applies upsampling to match the number of input frames, each frame's prior has an associated output prediction. For each predicted sign prior output, $\hat{y}_t^{\mathcal{P}}$, we train the model to regress to the corresponding target value $y_t^{\mathcal{P}}$ from prior $\mathcal{P}$ at frame t, using smooth L1 loss ($L1$) across all sign priors for each frame in the input video sequence. The same loss is applied to the sign activity prior for consistency. We also mitigate the impact of low-quality keypoints in our objective function by only using keypoints with over 50% confidence from the pose estimation model and masking missing keypoints from the loss. The final reconstruction loss is as follows:

$$\mathcal{L}_{\text{recon}} = \sum_{\mathcal{P} \in \mathbf{P}} w_{\mathcal{P}} L1(\hat{y}^{\mathcal{P}}, y^{\mathcal{P}}) \tag{2}$$

where $w_{\mathcal{P}}$ is the loss weighting for the prior $\mathcal{P}$.

5.2. Representation Regularization

Feature Regularization. Self-supervised methods have shown that regularizing features improves representation quality [4, 51]. To enhance our representations, we incorporate a variance and covariance loss into z^{avg}.

The **variance loss**, $\mathcal{L}_{\text{var}}$, encourages diversity in the learned representations by spreading them across the representation space. To compute the variance loss, we first calculate the standard deviation for each feature dimension (j) across a batch (N).

$$\sigma_j = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (z_{i,j}^{\text{avg}} - \bar{z}_j^{\text{avg}})^2} \tag{3}$$

The diversity of the features are then encouraged through a hinge function to ensure that the variance does not fall below a threshold of one.

$$\mathcal{L}_{\text{var}} = \sum_{j=1}^{D} \max(0, 1 - \sigma_j) \tag{4}$$

The **covariance loss**, $\mathcal{L}_{\text{cov}}$, reduces correlations among features, helping to avoid redundancy. To compute the covariance loss, we first need to calculate the covariance matrix such that:

$$\mathcal{C}_{j,k} = \frac{1}{N} \sum_{i=1}^{N} (z_{i,j}^{\text{avg}} - \bar{z}_j^{\text{avg}})(z_{i,k}^{\text{avg}} - \bar{z}_k^{\text{avg}}) \tag{5}$$

where $\mathcal{C}_{j,k}$ represents the covariance between feature dimensions j and k and $\bar{z}_j^{\text{avg}}$ is the mean of the j-th feature across the batch. The loss then penalises off-diagonal values in the covariance matrix, encouraging different feature dimensions to be uncorrelated:

$$\mathcal{L}_{\text{cov}} = \sum_{j \neq k} \mathcal{C}_{j,k}^2 \tag{6}$$

Style Agnostic Representations. We also aim to encourage the encoder to capture robust and generalizable sign features, while filtering out irrelevant details such as background and person-specific appearance. We can explicitly learn this by introducing a discriminator that evaluates whether two representations share the same "style", defined here by background and appearance features.

To extract style information, we calculate the gram matrix, commonly used in image style transfer [15], from the $\mathbf{z}^{\text{emb}}$ tokens, averaging across the column dimension to produce the style representation $z^{\text{style}} \in \mathbb{R}^D$. We then pair z^{avg} with z^{style} and pass them to the discriminator, which learns to output 1 for matching pairs and 0 for non-matching pairs.

To generate these representations pairs, as shown in Fig. 1(right), we randomly crop a video sequence of length L to create two segments, A_1 and A_2, each of length T. For each, we extract $z_{A_1}^{\text{avg}}$ and $z_{A_1}^{\text{style}}$, as well as $z_{A_2}^{\text{avg}}$ and $z_{A_2}^{\text{style}}$, assuming that they share background and appearance features due to originating from the same video. From a different video, we obtain segment B with its style z_B^{style}, which provides a contrasting background and appearance.

The discriminator, $\mathfrak{D}$, is then trained to produce 0 output for mismatched styles and 1 for matched styles, such that $\mathfrak{D}(z_{A_1}^{\text{avg}}, z_B^{\text{style}}) = 0$ and $\mathfrak{D}(z_{A_1}^{\text{avg}}, z_{A_2}^{\text{style}}) = 1$. This can be used

to further enhance the generalization of the encoder, we add an adversarial loss during pretraining, designed to fool the discriminator which encourages the model to focus on sign-specific content over style-related features. The adversarial loss is formalized as follows:

$$\mathcal{L}_{pos}^{(A_1,A_2)} = \max(0, \mathfrak{D}(z_{A_1}^{\text{avg}}, z_{A_2}^{\text{style}}) - \mathbb{E}_{q\sim \text{U}}\mathfrak{D}(q)) \quad (7)$$

$$\mathcal{L}_{neg}^{(A_1,B)} = \max(0, \mathbb{E}_{q\sim \text{M}}\mathfrak{D}(q) - \mathfrak{D}(z_{A_1}^{\text{avg}}, z_{B}^{\text{style}})) \quad (8)$$

$$\mathcal{L}_{\text{adv}}^{A_1} = (\mathcal{L}_{pos}^{(A_1,A_2)})^2 + (\mathcal{L}_{neg}^{(A_1,B)})^2 \quad (9)$$

where $\mathbb{E}_{q\sim \text{M}}\mathfrak{D}(q)$ is the expected discriminator output of style for representation pairs that come from the same video sequence and $\mathbb{E}_{q\sim \text{U}}\mathfrak{D}(q)$ for when they come from different video sequence. We apply this loss only if $\mathbb{E}_{q\sim \text{M}}\mathfrak{D}(q) > \mathbb{E}_{q\sim \text{U}}\mathfrak{D}(q)$, ensuring stability by restricting the adversarial loss to cases where the discriminator is likely to correctly identify the matches.

Our final loss for pretraining is defined as:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{recon}} + w_{\text{var}}\,\mathcal{L}_{\text{var}} + w_{\text{cov}}\,\mathcal{L}_{\text{cov}} + w_{\text{adv}}\,\mathcal{L}_{\text{adv}} \quad (10)$$

where w_{adv}, w_{var} and w_{cov} are the respective weighting factors for the adversarial, variance and covariance losses.

6. Representations for Dictionary Retrieval

Sign dictionary retrieval is the task of identifying signs from a predefined database. This allows for quick lookup of related signs given a query sign which supports efficient dataset creation and promotes cross-linguistic understanding of signs [39]. The SignRep model can be used as a feature extractor for dictionary retrieval. Given a query sign video of length L, we apply a sliding window to extract segment representations $\mathbf{Z}^{\text{avg}} = \{z_1^{\text{avg}}, z_2^{\text{avg}}, \ldots, z_n^{\text{avg}}, \ldots, z_N^{\text{avg}}\}$, where $N = \lfloor \frac{L-T}{\text{stride}} \rfloor + 1$. Since isolated signs typically begin and end in a resting pose, we use the output from

$$\gamma_n^{\text{act}} = \max(\mathcal{P}^{\{LH,\text{act}\}}, \mathcal{P}^{\{RH,\text{act}\}}) \quad (11)$$

to identify whether the signer is active. By checking hand activity, we compute a weighted average of the segment representations resulting in

$$z^{\text{out}} = \frac{1}{\sum \gamma_n^{\text{act}}} \sum_{n=1}^{N} \gamma_n^{\text{act}} z_n^{avg} \quad (12)$$

where z^{out} is the representation for the query video. Finally, we extract features for all dictionary videos and classify a query video by finding the closest match using cosine similarity of the normalized representations.

Class Probability Distribution. The retrieval approach enables the recognition of visually similar sign classes without additional training, which is useful for improving the accuracy of downstream tasks such as recognition. We construct a class distribution by computing the cosine similarity across all samples in the training dataset, capturing the visual proximity between them. From the resulting cosine similarity matrix, we compute inter-class similarities using the sign-sample labels to generate a matrix of shape $\mathbb{R}^{C\times C}$, where C represents the number of classes. Applying a temperature-scaled softmax to each row of this matrix yields a probability distribution for each class relative to the others, forming a class probability distribution matrix $\phi \in \mathbb{R}^{C\times C}$, where each element of ϕ lies in the range $[0, 1]$.

We can incorporate the class probability distribution ϕ as a regularization term in the downstream recognition task using KL divergence. Given a sign video with target label c, we can compute the KL divergence between the predicted class distribution $\bar{y}$ (from the output classifier layer) and ϕ_c which is the pre-computed class probability distribution of c. This results in:

$$\mathcal{L}_{\phi} = \kappa\,\phi_c[c]\,\text{KL}(\phi_c \parallel \bar{y}) \quad (13)$$

where κ is the weighting factor for the loss and $\text{KL}(\phi_c \parallel \bar{y})$ is the KL divergence. To emphasize the contribution of the target class distribution, we scale the KL divergence by $\phi_c[c]$, which is the value of the class probability for ϕ_c at c.

7. Experiments

We evaluate our model on sign recognition and retrieval using three datasets: ASL-Citizen [11], WLASL2000 [28], and NMFs-CSL [19]. The ASL-Citizen dataset includes 2,731 isolated ASL signs recorded from a webcam. The WLASL2000 dataset, sourced from the web, contains 2,000 common ASL signs and presents challenges due to its noisy nature and limited samples per sign. The NMFs-CSL dataset consists of 1,067 Chinese Sign Language (CSL) signs, which require recognition of non-manual cues, such as facial expressions, to accurately identify signs.

7.1. Evaluation Protocol

We follow standard evaluation protocols for sign recognition, measuring top-1 and top-5 per-instance and per-class accuracy on WLASL and NMFs-CSL. For ASL-Citizen, we use the benchmark metrics specified for this dataset, including discounted cumulative gain (DCG), mean reciprocal rank (MRR), and top-1 and top-5 instance accuracy.

To evaluate our model as a frozen feature extractor, we apply a retrieval protocol based on the dictionary-based retrieval approach from [11]. We choose a retrieval approach over a linear evaluation protocol, as dictionary retrieval is a common and practical task in sign language processing. This method also offers a better evaluation of the generalization of the features to unseen datasets, which is particularly important given the evolving nature of sign languages.

7.2. Experimental Setting

Pretraining. We pretrain our model on the YouTube-SL-25 dataset [42], which consists of large-scale continuous sign videos from YouTube. We initialize the model with the base video Hiera-B architecture, pretrained on Kinetics using MAE. We randomly select 16 consecutive frames as inputs based on a co-articulated sign typically lasting for around 13 frames [1, 34, 47]. No changes are made to the original encoder architecture, while the decoder is replaced with our lightweight sign decoder.

Recognition. For downstream tasks, our approach leverages only the pretrained encoder, discarding the decoder and entirely eliminating the reliance on keypoint extraction. For recognition, we extend the pretrained model's input sequence to 64 frames. We preserve the pretrained weights by inflating the patch embeddings, which preserves the computational efficiency by avoiding increasing the number of tokens in the transformer, detailed in the supplementary material. A linear classifier is applied to the pooled features from the encoder model to predict the sign classes.

Retrieval. For dictionary retrieval evaluation, we follow the approach outlined in Sec. 6 using a stride of 2. For the label assignment, we compute the cosine similarity between the test video representation and those in the training dictionary. Each test query video is assigned the label of the training video that yields the highest similarity score.

Additional implementation details are provided in the supplementary material.

7.3. Evaluation on Sign Recognition

In Tab. 1, our single-modality model demonstrates significant improvements over all existing single-modality approaches, achieving performance on par with complex multi-modal methods for WLASL2000. While ensembling with multi-modal approaches could further improve performance, we emphasize the importance of pretraining. Our Hiera encoder model with our pretraining, achieves these results without the extensive architectural modifications required by other methods. Instead of relying on additional modalities as input, our pretrained model has learned these features to create a robust sign representation model. Importantly, our pretraining strategy does not rely on any annotated sign data.

The effectiveness of our SignRep model on the NMFs-CSL dataset is further illustrated in Tab. 2, where we achieve a top-1 accuracy of 84.1% and a top-5 accuracy of 98.8%, outperforming all other methods, including multi-modal approaches. Notably, the closest single-modality comparison is the StepNet RGB model, which achieves 77.2% in top-1 accuracy, nearly 7% lower than our model. While StepNet focuses on developing specialized architecture for sign, we demonstrate our framework can effectively learn sign features from large-scale datasets.

	Instance Acc.		Class Acc.	
Method	Top-1	Top-5	Top-1	Top-5
Skeleton-based				
ST-GCN [50]	34.40	66.57	32.53	65.45
SignBERT [17]	39.40	73.35	36.74	72.38
BEST [52]	46.25	79.33	43.52	77.65
SignBERT+[20]	**48.85**	**82.48**	**46.37**	**81.33**
Multi-modal				
BEST (+R) [52]	54.59	88.08	52.12	87.28
SignBERT(+R) [17]	54.69	87.49	52.08	86.93
SignBERT+(+R) [20]	55.59	89.37	53.33	88.82
SAM (5ξ)[23]	58.73	91.46	55.93	90.94
SAM-v2 (5ξ) [22]	59.39	91.48	56.63	90.89
NLA-SLR [57]	61.05	91.45	58.05	90.70
NLA-SLR(3ξ) [57]	**61.26**	91.77	58.31	90.91
StepNet (R+F) [40]	61.17	**91.94**	**58.43**	**91.43**
RGB-based				
I3D [8]	32.48	57.31	-	-
I3D(BSL1K) [1]	46.82	79.36	44.72	78.47
StepNet [40]	56.89	88.64	54.54	87.97
SignRep (Ours)	**61.05**	**90.27**	**58.89**	**89.44**

Table 1. Comparison of downstream sign recognition results on WLASL2000. ξ, (R) and (F) denotes a multi-crop inference, RGB and Optical Flow modality respectively.

Method	**Top-1**	**Top-5**
I3D [8]	64.4	88.0
TSM [29]	64.5	88.7
Slowfast [14]	66.3	86.6
GLE-Net [19]	69.0	88.1
HMA ($\diamond$) [18]	75.6	95.3
StepNet [40]	77.2	92.5
SignBERT (H+R) ($\diamond$) [17]	78.4	97.3
BEST ($\diamond$) [52]	79.2	97.1
NLA-SLR ($\diamond$) [57]	83.4	98.3
StepNet (R+F) ($\diamond$) [40]	83.6	97.0
NLA-SLR ($\diamond$,3ξ) [57]	83.7	98.5
SignRep (Ours)	**84.1**	**98.8**

Table 2. Comparison of downstream sign recognition results on NMFs-CSL. $\diamond$ denotes methods using multi-modality/ensemble models and ξ indicates a multi-crop inference.

Our results on the ASL-Citizen dataset, shown in Tab. 3, further demonstrate the effectiveness of our model where it surpasses the previous baseline I3D model by 18% in top-1 accuracy.

Model	DCG	MRR	Rec@1	Rec@5
ST-GCN* [50]	76.37	69.97	59.52	82.68
I3D* [8]	79.13	73.32	63.10	86.09
SignRep (Ours)	**90.84**	**88.05**	**81.37**	**96.11**

Table 3. Comparison of downstream sign recognition results on ASL-Citizen. * denotes results produced by [11]

.

7.4. Evaluation on Sign Dictionary Retrieval

To assess the effectiveness of our model as a feature extractor, we utilize a dictionary-based sign retrieval method that directly evaluates the quality of the learned representations without any fine-tuning. In Tab. 4, we compare our approach to the Hiera model pretrained with MAE on both the Kinetics and YT-SL datasets. We apply the same training settings outlined in Sec. 7.2 for pretraining on YT-SL. Our results demonstrate that our proposed pretraining strategy significantly outperforms standard MAE, even when trained on YT-SL, by optimizing specifically for sign language representation, whereas MAE's pixel-reconstruction objective does not directly support learning the sign specific features needed for sign retrieval.

We also compare our model's retrieval performance with the use of raw 3D keypoints and joint angles averaged over time. As shown in Tab. 4, our method achieves a substantial performance increase, with top-1 retrieval scores more than tripling those based on hand joint angles. These results underscore the strength of our pretraining approach in capturing spatio-temporal features specific to sign language, establishing our method as a highly effective framework for sign representation learning. We also validate the importance of incorporating hand activity awareness, observing that weighted averaging based on hand activity improves retrieval performance compared to averaging.

7.5. Ablation Studies

Impact of Pretraining on Sign Retrieval. In Tab. 5, we examine the effects of various elements in our pretraining strategy on retrieval performance in WLASL2000. Our results show that combining all three types of sign priors that have angle, keypoint and distance yields the highest retrieval accuracy, underscoring the benefit of capturing multiple aspects of the sign features. We also analyze the role of masking during pretraining, removing it leads to a noticeable decline in retrieval performance, indicating that masking is essential for robust representation learning. While we observe lower reconstruction loss without masking, this does not translate to better retrieval scores, suggesting that precise reconstruction alone does not necessarily produce effective sign representations. Finally, the addition of both adversarial loss and variance-covariance regularization yields the strongest retrieval results, confirming that these feature regularization techniques improve the quality and robustness of learned sign representations.

Impact of Pretraining for Recognition. In Tab. 6, we demonstrate that our pretrained model substantially outperforms both the MAE model pretrained on Kinetics and the model pretrained on YT-SL with pixel MAE. Additionally, the baseline Hiera model pretrained on Kinetics, achieving a top-1 accuracy of 51.5%, surpasses prior keypoint-based masked learning methods such as SignBERT, BEST and SignBERT+, as shown in Tab. 1. This result underscores the value of RGB input, revealing the limitations of using keypoints as a primary modality. Previous keypoint methods have required ensembling with RGB to remain competitive, illustrating the challenges of relying solely on skeletal information as an input modality for sign recognition. Finally, we observe that incorporating the sign class distribution loss further enhances recognition accuracy, improving top-1 accuracy from 59.9% to 61.0% with a κ of 0.2.

8. Feature Extractor for Sign Translation

In this section, we evaluate SignRep as a feature extractor for sign translation by integrating it into two open-source translation models: Sign2GPT [49] on RWTH-PHOENIX-Weather 2014T (Phoenix14T) [6] and CSL-Daily [54], as well as the approach from [43] on How2Sign [13]. We chose these models because they allow direct evaluation of SignRep as a frozen feature extractor without requiring sign-text pretraining or significant modifications to the architecture. We follow the same translation training hyperparameters as the original papers to ensure a fair comparison and isolate the impact of replacing the visual backbone.

For Phoenix14T and CSL-Daily, we omit the pseudo-gloss pretraining used in [49] to ensure a fair evaluation of SignRep's learned features without additional linguistic supervision. In [49], they show that the DinoV2 backbone requires fine-tuning with LoRA to perform well on sign translation. Instead, we replace the original learnable DinoV2 backbone with pre-extracted SignRep features, obtained using our sliding window approach with a stride of 2. This setup places ours at a disadvantage, as we remove the trainable backbone weights and data augmentation, relying solely on pre-extracted features. Despite this, SignRep achieves competitive translation performance, as shown in Tab. 7, performing comparably on Phoenix14T and achieving substantial gains on CSL-Daily. By replacing the learnable DinoV2 model with pre-extracted SignRep features during training, we significantly reduce computational costs, making large-scale translation training more efficient and accessible without sacrificing performance. Our approach removes an additional 5.5GFLOP per frame required by the learnable DinoV2 model, which is important as translation models process tens to hundreds of frames.

Features	ASL-Citizen			WLASL2000			NMFs-CSL		
	DCG	Rec@1	Rec@5	DCG	Rec@1	Rec@5	DCG	Rec@1	Rec@5
HieraMAE-Kinetic	11.64	0.25	0.84	13.21	2.08	3.40	23.29	3.96	12.18
HieraMAE-YTSL	12.12	0.39	1.34	14.06	2.57	4.41	28.03	7.57	18.38
All Joint Angles	13.82	0.57	2.17	17.92	2.54	6.50	32.51	7.93	25.50
All Keypoints	14.29	0.98	2.74	19.37	3.16	8.23	36.64	12.26	30.99
Hand Keypoints	25.91	7.96	19.58	28.11	7.57	20.92	41.72	15.91	42.62
Hand Joint Angle	26.93	8.81	21.41	30.61	9.42	24.36	44.17	18.13	46.34
SignRep (avg)	61.40	37.47	68.77	53.57	26.16	60.98	79.51	58.48	91.85
SignRep (weighted)	**71.21**	**49.95**	**80.09**	**57.93**	**29.92**	**67.41**	**83.05**	**63.04**	**95.63**

Table 4. Comparison of generalization results on sign retrieval with no downstream training applied to SignRep.

Prior Comp.				Reg.		
angle	kpt	dist	mask	var+cov	adv	DCG
✓			✓			45.1
	✓		✓			32.9
		✓	✓			47.2
✓	✓	✓	✓			48.5
✓	✓	✓				46.3
✓	✓	✓	✓	✓		49.9
✓	✓	✓	✓	✓	✓	**50.7**

Table 5. Comparison of the impact of sign priors, masking and regularization on the impact of retrieval. Retrieval results are obtained on WLASL using a stride of 8.

Method	κ	Instance Acc.		Class Acc.	
		top-1	top-5	top-1	top-5
MAE (Kinetic)	-	51.5	83.4	48.4	82.0
MAE (YT-SL)	-	57.2	87.4	54.6	86.3
SignRep	0.0	59.9	90.2	57.4	89.2
SignRep	0.1	60.4	90.1	57.8	89.1
SignRep	0.2	**61.0**	**90.3**	**58.9**	**89.4**
SignRep	0.5	59.8	89.4	57.7	88.8

Table 6. Comparisons of the impact of pretraining and weighting for the class distribution loss on WLASL.

Sign2GPT Backbone	Phoenix14T		CSL-Daily	
	B-4	R	B-4	R
DinoV2(LoRA)	19.42	**45.23**	12.96	41.12
SignRep(extracted)	**20.38**	45.17	**16.33**	**42.67**

Table 7. Comparison of translation results using Sign2GPT [49] by replacing the learnable DinoV2 backbone with our extracted SignRep features. B-4 denotes BLEU4 and R denotes ROUGE.

This leads to an 80% reduction in GPU memory usage during training, lowering requirements from 60GB to 11GB with a batch size of 8. This reduction makes training on standard hardware more feasible, enabling broader accessibility for researchers working with sign translation models.

Tarrés's Backbone	How2Sign	
	rBLEU	BLEU
I3D(Supervised)	2.21	8.03
SignRep(Self-Supervised)	**2.74**	**8.66**

Table 8. Comparison of translation results using Tarrés [43], replacing the I3D features with our SignRep features.

For How2Sign, in Tab. 8, we replace the supervised I3D features with our self-supervised SignRep features on the translation framework from [43] and observe performance improvements. Unlike I3D features, which are pretrained with supervision on a labeled sign recognition data, SignRep learns features purely through self-supervision, highlighting our framework's effectiveness for learning sign-specific features without supervision.

9. Conclusions

In this work, we introduced a scalable, self-supervised framework for sign representation learning without labeled sign datasets. Using masked autoencoding with sign priors, adversarial loss and feature regularizations, our approach enhances generalization while eliminating the need for skeletal keypoints, multimodal or multi-branch architectures in downstream tasks. Our single model achieves state-of-the-art performance on multiple sign recognition benchmarks with a simpler, more efficient design. Beyond recognition, our representations demonstrate versatility in sign dictionary retrieval on unseen datasets and serve as an efficient feature extractor for existing sign translation systems, reducing computational costs while maintaining strong performance. Our findings highlight the potential of our self-supervised learning framework for scalable sign modeling and encourage further research into practical applications of sign representations.

Acknowledgements

This work was supported by SNSF project 'SMILE II' (CRSII5 193686), European Union's Horizon2020 programme ('EASIER' grant agreement 101016982) and the Innosuisse IICT Flagship (PFFS-21-47). This work reflects only the authors view and the Commission is not responsible for any use that may be made of the information it contains. Neither Necati Cihan Camgoz nor Meta were involved in the model training, evaluation, or use of the datasets. The authors also thank Maksym Ivashechkin for their assistance in providing the 3D pose estimation model.

References

[1] Samuel Albanie, Gül Varol, Liliane Momeni, Triantafyllos Afouras, Joon Son Chung, Neil Fox, and Andrew Zisserman. Bsl-1k: Scaling up co-articulated sign language recognition using mouthing cues. In *European conference on computer vision*, pages 35–53. Springer, 2020. 1, 2, 6

[2] Samuel Albanie, Gül Varol, Liliane Momeni, Hannah Bull, Triantafyllos Afouras, Himel Chowdhury, Neil Fox, Bencie Woll, Rob Cooper, Andrew McParland, et al. Bbc-oxford british sign language dataset. *arXiv preprint arXiv:2111.03635*, 2021. 2

[3] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023. 2

[4] Adrien Bardes, Jean Ponce, and Yann Lecun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR 2022-International Conference on Learning Representations*, 2022. 4

[5] P Boyes Braem and RL Sutton-Spence. *The Hands Are The Head of The Mouth. The Mouth as Articulator in Sign Languages*. Hamburg: Signum Press, 2001. 1

[6] 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*, pages 7784–7793, 2018. 7, 2

[7] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10033, 2020. 1

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 6, 7

[9] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. 1

[10] Konstantinos M Dafnis. Bidirectional skeleton-based isolated sign recognition using graph convolution networks. In *LREC proceedings*, 2022. 2

[11] Aashaka Desai, Lauren Berger, Fyodor Minakov, Nessa Milano, Chinmay Singh, Kriston Pumphrey, Richard Ladner, Hal Daumé III, Alex X Lu, Naomi Caselli, et al. Asl citizen: a community-sourced dataset for advancing isolated sign language recognition. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 5, 7

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. 2

[13] Amanda Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giro-i Nieto. How2sign: a large-scale multimodal dataset for continuous american sign language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2735–2744, 2021. 7, 2

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 6

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2

[17] Hezhen Hu, Weichao Zhao, Wengang Zhou, Yuechen Wang, and Houqiang Li. Signbert: Pre-training of hand-model-aware representation for sign language recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11087–11096, 2021. 1, 2, 6

[18] Hezhen Hu, Wengang Zhou, and Houqiang Li. Hand-model-aware sign language recognition. In *AAAI*, pages 1558–1566, 2021. 6

[19] Hezhen Hu, Wengang Zhou, Junfu Pu, and Houqiang Li. Global-local enhancement network for nmf-aware sign language recognition. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, 17(3): 1–19, 2021. 1, 2, 5, 6

[20] Hezhen Hu, Weichao Zhao, Wengang Zhou, and Houqiang Li. Signbert+: Hand-model-aware self-supervised pre-training for sign language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11221–11239, 2023. 1, 2, 6

[21] Maksym Ivashechkin, Oscar Mendez, and Richard Bowden. Improving 3d pose estimation for sign language. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5. IEEE, 2023. 2, 1, 3

[22] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Sign language recognition via skeleton-aware multi-model ensemble. *arXiv preprint arXiv:2110.06161*, 2021. 1, 2, 6

[23] Songyao Jiang, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li, and Yun Fu. Skeleton aware multi-modal sign language

recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3413–3423, 2021. 1, 2, 6

[24] Tao Jiang, Necati Cihan Camgoz, and Richard Bowden. Skeletor: Skeletal transformers for robust body-pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3394–3402, 2021. 2

[25] Peiqi Jiao, Yuecong Min, and Xilin Chen. Visual alignment pre-training for sign language translation. In *European Conference on Computer Vision*, pages 349–367. Springer, 2025. 1, 2

[26] Hamid Reza Vaezi Joze and Oscar Koller. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053*, 2018. 1

[27] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[28] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469, 2020. 1, 2, 5

[29] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 6

[30] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020. 2

[31] Amit Moryossef. Optimizing hand region detection in mediapipe holistic full-body pose estimation to improve accuracy and avoid downstream errors. *arXiv preprint arXiv:2405.03545*, 2024. 2

[32] Amit Moryossef, Ioannis Tsochantaridis, Joe Dinn, Necati Cihan Camgoz, Richard Bowden, Tao Jiang, Annette Rios, Mathias Muller, and Sarah Ebling. Evaluating the immediate applicability of pose estimation for sign language recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3434–3440, 2021. 2

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 2

[34] Tomas Pfister, James Charles, and Andrew Zisserman. Large-scale learning of sign language by watching tv. 2013. 6

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[36] Charles Raude, KR Prajwal, Liliane Momeni, Hannah Bull, Samuel Albanie, Andrew Zisserman, and Gül Varol. A tale of two languages: Large-vocabulary continuous sign language recognition from spoken language supervision. *arXiv preprint arXiv:2405.10266*, 2024. 2

[37] Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgöz, and Jean Maillard. Towards privacy-aware sign language translation at scale. *arXiv preprint arXiv:2402.09611*, 2024. 1, 2

[38] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 2, 3

[39] Marc Schulder, Sam Bigeard, Maria Kopf, Thomas Hanke, Anna Kuder, Joanna Wójcicka, Johanna Mesch, Thomas Björkstrand, Anna Vacalopoulou, Kyriaki Vasilaki, et al. Signs and synonymity: Continuing development of the multilingual sign language wordnet. In *Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources*, pages 343–353, 2024. 5

[40] Xiaolong Shen, Zhedong Zheng, and Yi Yang. Stepnet: Spatial-temporal part-aware network for isolated sign language recognition. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(7):1–19, 2024. 6

[41] Ozge Mercanoglu Sincan and Hacer Yalim Keles. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE access*, 8:181340–181355, 2020. 1

[42] Garrett Tanzer and Biao Zhang. Youtube-sl-25: A large-scale, open-domain multilingual sign language parallel corpus. *arXiv preprint arXiv:2407.11144*, 2024. 6

[43] Laia Tarrés, Gerard I Gállego, Amanda Duarte, Jordi Torres, and Xavier Giró-i Nieto. Sign language translation from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5625–5635, 2023. 7, 8, 2

[44] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35, 2022. 2

[45] Dave Uthus, Garrett Tanzer, and Manfred Georg. Youtube-asl: A large-scale, open-domain american sign language-english parallel corpus. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[46] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2

[47] Ville Viitaniemi, Tommi Jantunen, Leena Savolainen, Matti Karppa, and Jorma Laaksonen. S-pot-a benchmark in spotting signs within continuous signing. In *LREC*, pages 4–1, 2014. 6

[48] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. Learnt contrastive concept embeddings for sign recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1945–1954, 2023. 2

[49] Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. Sign2gpt: Leveraging large language models for gloss-free sign language translation. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 7, 8

[50] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 6, 7

[51] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021. 4

[52] Weichao Zhao, Hezhen Hu, Wengang Zhou, Jiaxin Shi, and Houqiang Li. Best: Bert pre-training for sign language recognition with coupling tokenization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3597–3605, 2023. 1, 2, 6

[53] Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang. Gloss-free sign language translation: Improving from visual-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20871–20881, 2023. 1, 2

[54] 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*, pages 1316–1325, 2021. 7, 2

[55] Wengang Zhou, Weichao Zhao, Hezhen Hu, Zecheng Li, and Houqiang Li. Scaling up multimodal pre-training for sign language understanding. *arXiv preprint arXiv:2408.08544*, 2024. 1, 2

[56] Simone Zini, Alex Gomez-Villa, Marco Buzzelli, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de weijer. Planckian jitter: countering the color-crippling effects of color jitter on self-supervised training. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[57] Ronglai Zuo, Fangyun Wei, and Brian Mak. Natural language-assisted sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14890–14900, 2023. 1, 2, 6

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.

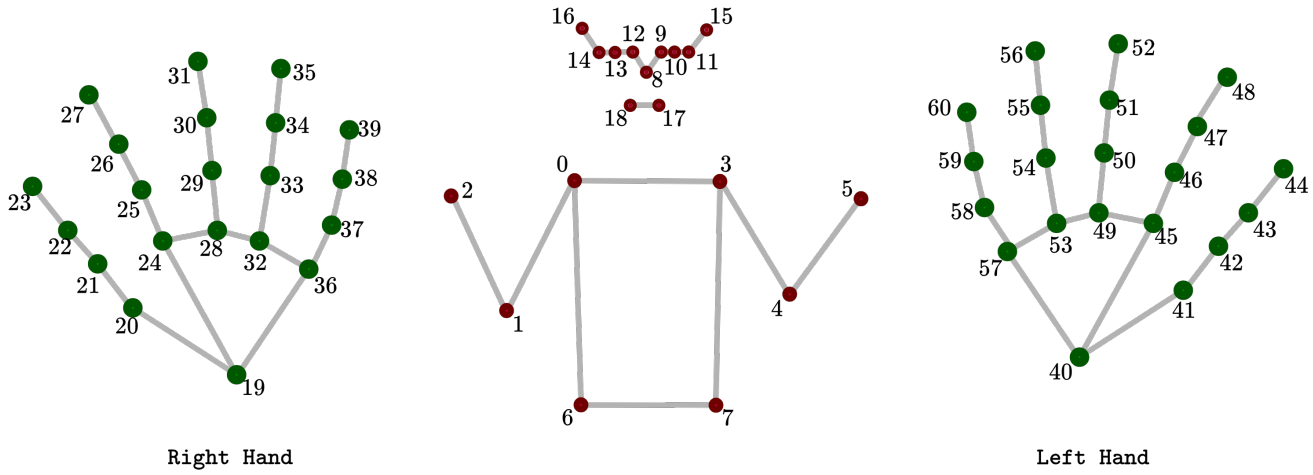

Figure 2. Visualization of 3D keypoint extracted. Numbers alongside the nodes represent the keypoint indices. For visualization purposes, we separate the left and right hand from the body.

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.

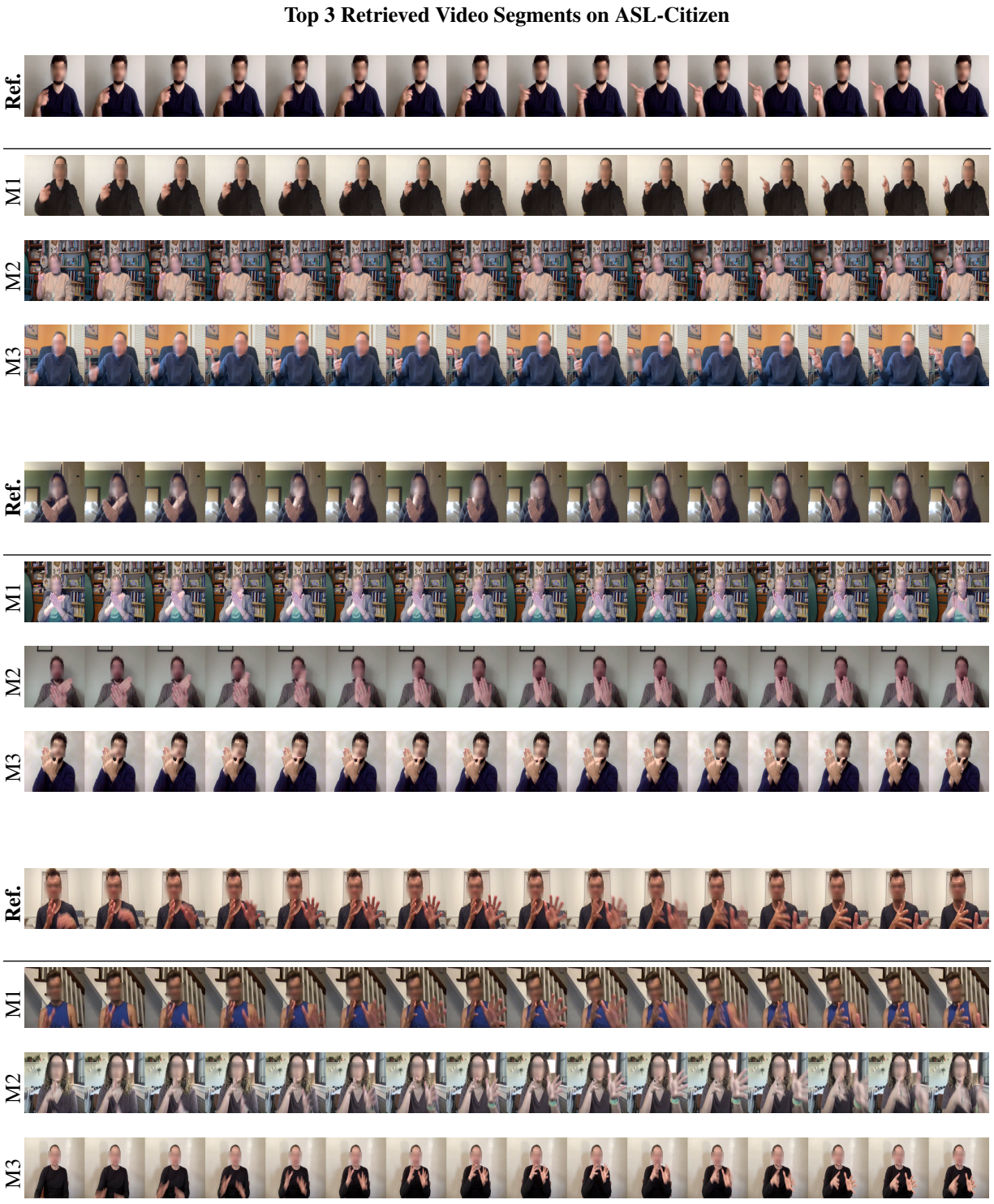

Figure 3. Qualitative results for ASL-Citizen for retrieval based on features extracted from the pretrained SignRep. Given the reference sequence (Ref.), the Top-3 most similar videos are retrieved based on the cosine similarity of the output representations. M1 denotes the closest match, M2 is the second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

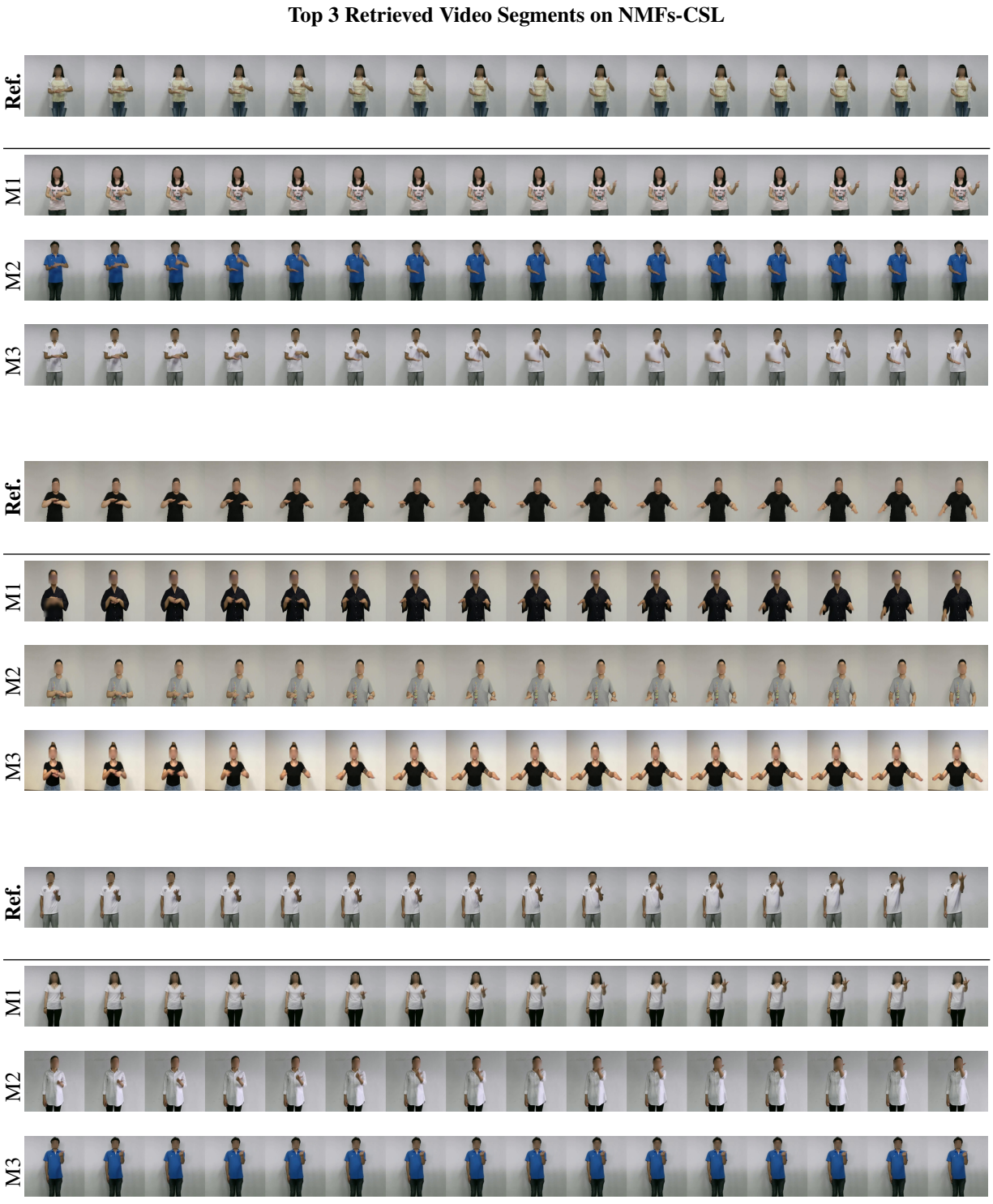

Figure 4. Qualitative results for NMFs-CSL for retrieval based on features extracted from the pretrained SignRep. Given the reference sequence (Ref.), the Top-3 most similar videos are retrieved based on the cosine similarity of the output representations. M1 denotes the closest match, M2 is the second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