Improving Sign Language Understanding with a Multi-Stream Masked Autoencoder Trained on ASL Videos

Anonymous ACL submission

Abstract

Understanding sign language remains a significant challenge, particularly for low-resource sign languages with limited annotated data. Motivated by the success of large-scale pretraining in deep learning, we propose Multi-Stream Masked Autoencoder (MS-MAE) — a simple yet effective framework for learning sign language representations from skeleton-based video data. We pretrained a model with MS-MAE on the YouTube-ASL dataset, and then adapted it to multiple downstream tasks across different sign languages. Experimental results show that MS-MAE achieves competitive or superior performance on a range of isolated sign language recognition benchmarks and sign language translation tasks across several sign languages. These findings highlight the potential of leveraging large-scale, high-resource sign language data to boost performance in low-resource sign language scenarios. Additionally, analysis of the model's attention maps reveals its ability to cluster adjacent pose sequences within a sentence, some of which align with individual signs, offering insights into the mechanisms underlying successful transfer learning.

1 Introduction

Sign languages (SLs), which rely on hand movements, facial expressions, and body gestures to convey meaning, serve as a primary mean of communication within deaf communities. However, a significant communication gap persists between deaf and hearing populations. In response, research on sign language understanding, including Sign Language Recognition (SLR) (Li et al., 2020; Desai et al., 2023; Kapitanov et al., 2023) and Sign Language Translation (SLT) (Camgöz et al., 2018; Zhou et al., 2021a; Duarte et al., 2021), has garnered increasing attention, especially in the era of deep learning. Despite these advances, the development of sign language understanding systems is still hindered by the scarcity of large-scale, publicly available SL datasets.

To overcome this challenge, recent efforts have turned to the vast amount of sign language video content available online, particularly on YouTube. For instance, YouTube-ASL (YT-ASL) (Uthus et al., 2023) consists of 984 hours of annotated American Sign Language (ASL) videos. Meanwhile, YouTube-SL-25 (Tanzer and Zhang, 2024) expands the scope, collecting 3,207-hour videos spanning 25 different sign languages. These datasets have significantly accelerated progress in sign language understanding by enabling large-scale supervised pretraining strategies. The resulting pretrained models have proven effective in enhancing downstream tasks such as SLR and SLT.

Despite the significant contributions of YouTube-SL-25 toward the goal of "no language left behind" in sign language research, annotated resources for sign languages remain limited compared to those available for spoken language machine translation. Expanding annotated sign language datasets continues to be a major challenge. A more scalable way is to leverage unannotated data, as argued by (Rust et al., 2024). However, many sign languages still lack sufficient video resources for pretraining. This raises an important research question: Can knowledge learned from videos of known sign languages be transferred to unseen, low-resource sign languages? Addressing this question is crucial for making progress in adapting models to underrepresented sign languages. This study explores whether large-scale sign language video datasets from high-resource languages can be leveraged for effective representation learning and video encoder pretraining, with the goal of enhancing performance on downstream tasks in unseen sign languages.

Specifically, we introduce Multi-Stream Masked AutoEncoder (MS-MAE) designed to learn a strong sign language video encoder for sign language videos. MS-MAE begins by extracting three distinct pose streams, including body, left hand, and right hand, and encoding each as a separate token sequence. These sequences are then concatenated into a single unified stream and passed through a transformer (Vaswani et al., 2017) encoder. During self-supervised pretraining, MS-MAE randomly masks a subset of tokens in each stream and tasks the model with reconstructing the masked portions. In our experiments, we first pretrain the video encoder only with videos from the YT-ASL dataset, and then test on two downstream tasks, including Isolated SLR (ISLR) on ASL, Japanese Sign Language (JSL) and Russian Sign Language (RSL), and SLT on ASL, Chinese Sign Language (CSL) and German Sign Language (DGS).

Our contributions are as follows: (1) We propose a simple yet effective and efficient pretraining framework, MS-MAE. (2) Through experiments, we demonstrate that fine-tuning our model pretrained exclusively on ASL videos achieves competitive performance compared to SOTA methods across multiple sign languages. (3) We visualize the attention maps of the pretrained model and observe that it can group neighboring frames into clusters corresponding to individual signs or motion units in sentences of other SLs. This offers insights into what the model learns that can benefit the transfer learning.

2 Related Work

2.1 Transfer Learning of Supervised Pretraining

Data scarcity is a primary challenge in sign language processing, making transfer learning essential for enhancing both SLR and SLT performance. Several previous works employ either 2D Convolutional Neural Networks (CNNs) (Camgöz et al., 2020) pretrained on image classification tasks or 3D CNNs (Sarhan and Frintrop, 2020; Chen et al., 2022a) pretrained on action recognition tasks, such as S3D (Xie et al., 2018) and I3D (Carreira and Zisserman, 2017), as backbone feature extractors. While these approaches have demonstrated effectiveness, their performance is constrained by a domain shift between action recognition and sign language understanding. This gap arises from differences in task granularity, with sign language understanding requiring finer temporal and spatial understanding, thereby limiting further performance gains.

Another line of research involves in-domain transfer, or cross-lingual transfer learning, where models trained on high-resource sign languages are finetuned to adapt to low-resource sign languages, yielding significant performance improvements, including (Bird et al., 2020; Holmes et al., 2023). However, annotated sign language data are difficult to obtain and hard to scale up, highlighting the need for approaches that can leverage unannotated data.

2.2 Self-supervised Learning in Sign Language Understanding

Self-supervised learning, which leverages large-scale unlabeled data, has achieved remarkable success in various fields. In the domain of sign language, several works have adopted masked prediction strategies, such as BEST (Zhao et al., 2023) and SHuBERT (Gueuwou et al., 2024). Others follow a masked reconstruction paradigm. Among these, SignBERT (Zhou et al., 2021b) and SignBERT+ (Hu et al., 2023) employ BERT (Devlin et al., 2019)-like encoder-only architectures. Meanwhile, approaches like MASA (Zhao et al., 2024), SSVP-SLT (Rust et al., 2024), and SignRep (Wong et al., 2025) adopt MAE (He et al., 2022)-like asymmetric encoder-decoder architectures. Specifically, MASA performs masked reconstruction on skeleton-based input. SSVP-SLT targets RGB input, which is computationally intensive and demands substantial resources—its longest pretraining run reportedly takes two weeks on 64 A100 GPUs. To address these challenges, the recent work SignRep introduces an approach that takes RGB inputs but reconstructs pose sequences. This design significantly reduces computational costs during pretraining and removes the need for skeleton estimation tools at inference time.

However, RGB videos remain computationally intensive to process, especially in the context of sign language, which is inherently information-dense. Additionally, transformer-based architectures further amplify this challenge. As a result, finetuning the entire model for some downstream tasks, particularly in SLT, becomes impractical, limiting potential performance gains. Moreover, RGB-based MAEs typically tokenize videos into fixed-size patches. This patch-based tokenization can fragment critical visual cues across multiple tokens, potentially leading to low-efficiency learning.

In contrast, our pretraining framework operates on skeletal data in order to focus on the interaction among visual cues for efficient representation learning.

3 Method: Multi-Stream Masked AutoEncoder

An overview of MS-MAE is illustrated in Figure 1. The framework is an extension of SkeletonMAE (Wu et al., 2023) and MASA (Zhao et al., 2024). In this method, we begin by extracting skeleton sequences from sign language videos and dividing them into separate streams corresponding to the left hand, right hand, and upper body. Each stream is then encoded into a sequence of tokens. Our framework adopts an MAE-like asymmetric encoder-decoder architecture for pretraining. Specifically, we randomly drop several time steps within each stream, and the unmasked tokens are fed into a transformer encoder. The encoder outputs are then padded with learnable mask tokens and passed to the decoder. The pretraining objective is to reconstruct the original skeleton sequences. Unlike MASA, which aggregates all visual cues within each frame as a single input unit, our pretraining framework leverages skeletal data by explicitly decoupling visual cues and passing them through a transformer concurrently, which is close to the strategy of SignBERT (Zhou et al., 2021b).

3.1 Multi-Stream Transformer

Our encoder architecture consists of an embedding layer followed by a standard transformer encoder. We utilize MediaPipe Holistic (Lugaresi et al., 2019) to extract skeletal data from sign language videos. Each pose sequence consists of three distinct streams—left hand, right hand, and upper body—denoted as $\mathcal{P} = \{(P_t^{\mathrm{LH}}, P_t^{RH}, P_t^{B})\}_{t=1}^{n}$, where n is the total number of frames and each $P_t^p \in \mathbb{R}^{|K_p| \times D}$ contains the D-dimensional keypoints of part $p \in \{\mathrm{LH}, \mathrm{RH}, \mathrm{B}\}$. In this work, we only use x- and y-coordinates of the keypoints, so $D = 2$. The term $|K_p|$ is the number of keypoints for each body part.

We flatten and project each stream frame-wise:

$$\begin{aligned} \boldsymbol{x}_k^{\mathrm{B}} &= \mathrm{Linear}_B(\mathrm{flatten}(P_t^B)) \\ \boldsymbol{x}_k^{\mathrm{LH}} &= \mathrm{Linear}_H(\mathrm{flatten}(P_t^{LH})), \\ \boldsymbol{x}_k^{\mathrm{RH}} &= \mathrm{Linear}_H(\mathrm{flatten}(P_t^{RH})) \end{aligned} \tag{1}$$

where $t = 1, \cdots, n$.

Inspired by video transformers (Arnab et al., 2021; Tong et al., 2022) that leverage cubelet embeddings to encode spatio-temporal cubes, which can reduce computational cost through mitigating the redundancy of neighboring frames, we adopt the same strategy to reduce sequence length. Specifically, we use 1D convolutions with kernel size = stride = S to encode streams separately to ensure non-overlapping encoding:

$$\begin{aligned} \hat{\boldsymbol{x}}^B &= \mathrm{Conv1D_B}(\boldsymbol{x}^{\mathrm{B}}) \in \mathbb{R}^{(n/S) \times C} \\ \hat{\boldsymbol{x}}^{LH} &= \mathrm{Conv1D_H}(\boldsymbol{x}^{\mathrm{LH}}) \in \mathbb{R}^{(n/S) \times C} \\ \hat{\boldsymbol{x}}^{RH} &= \mathrm{Conv1D_H}(\boldsymbol{x}^{\mathrm{RH}}) \in \mathbb{R}^{(n/S) \times C} \end{aligned} \tag{2}$$

. Each stream is added to the same positional encoding, denoted as PE, so that the part token at the same time step can be correctly identified, and concatenated along the time channel into a single sequence as inputs to the transformer:

$$\begin{aligned} \mathrm{Emb}^p &= \hat{\boldsymbol{x}}^p + \mathrm{PE}[:n/S] \in \mathbb{R}^{(n/S) \times C} \\ \mathrm{Emb} &= [\mathrm{Emb}^{\mathrm{B}}; \mathrm{Emb}^{\mathrm{LH}}; \mathrm{Emb}^{\mathrm{RH}}] \in \mathbb{R}^{(3n/S) \times C} \end{aligned} \tag{3}$$

, and feed Emb into a standard transformer encoder $\mathbf{Z} = \mathrm{Transformer}(\mathrm{Emb})$.

By keeping streams separate up through the patch embedding, self-attention can explicitly model both intra-stream dynamics (e.g. left-hand over time) and cross-stream dependencies (e.g. right-hand vs. body), and during pretraining, we may apply masking to individual streams rather than entire frames for more granular learning.

3.2 Pretrain

In the pretraining stage, we employ an asymmetric encoder–decoder MAE architecture tailored to our multi-stream setting. Let $\mathcal{P} = \{P_t^p\}_{p \in \{\mathrm{B,LH,RH}\}, t=1}^{n}$ denote the set of input pose sequences. We apply a $\mathrm{PatchEmbed}(\cdot)$ function to each stream, producing cubelet tokens, which are augmented with positional encodings $\mathrm{Emb}^p \in \mathbb{R}^{(n/S) \times C}$. A random fraction r of tokens in each stream is masked; we denote the sets of visible and masked indices as V_p and M_p, respectively.

1. **Encoding.** All unmasked token embeddings $\{\mathrm{Emb}_i^p : i \in V_p,\ p \in \{B, \mathrm{LH}, \mathrm{RH}\}\}$ are passed through a transformer encoder to yield contextual representations $\mathbf{Z}_V \in \mathbb{R}^{\sum_p |V_p| \times C}$.
2. **Decoding.** For each masked index, we prepend a learnable mask token, concatenate the resulting embeddings with $\mathbf{Z}_V$ to form $\mathbf{Z} \in \mathbb{R}^{n \times C}$, and pass $\mathbf{Z}$ through a lightweight decoder. The decoder reconstructs outputs $\hat{\boldsymbol{t}}_i^p$ for all $i \in M_p$.

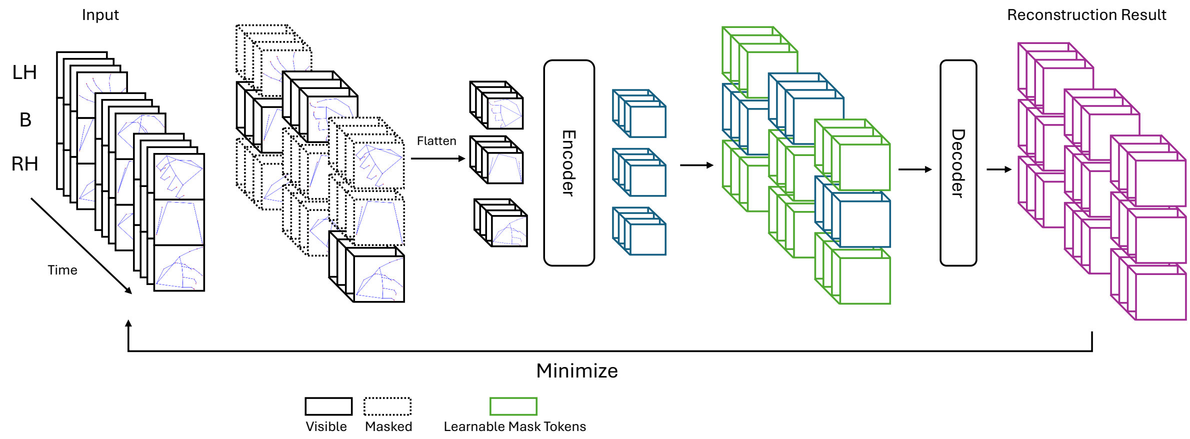


Figure 1: An overview of MS-MAE. Sign language videos are first converted into skeletal data using MediaPipe Holistic, and separated into left-hand, body, and right-hand streams. Each stream is divided into a sequence of spatiotemporal cubes. During pretraining, a portion of the tokens is masked, while the unmasked tokens are flattened and passed through an encoder to produce latent representations. These encoder outputs are concatenated with learnable mask tokens and fed into a decoder, which is trained to reconstruct the original input sequences.

3. **Reconstruction target & loss.** For each masked token index k, the target is the original sequence of keypoints within the corresponding cubelet $\mathbf{t}_k^p = [P_{kS}^p, \ P_{kS+1}^p, \ldots, P_{kS+S-1}^p] \in \mathbb{R}^{S\times(D\,|K_p|)}$.

   We minimize the mean squared error $\mathcal{L} = \frac{1}{\sum_p |M_p|} \sum_p \sum_{k\in M_p} \left\| \hat{\mathbf{t}}_k^p - \text{flatten}(\mathbf{t}_k^p) \right\|^2$.

This architecture encourages the encoder to learn the dependencies among different visual cues at different time steps. When computing the loss, we ignore any missing keypoints in $\mathbf{t}_k^p$ due to MediaPipe failures, to avoid the model being misled by noisy and absent detections.

4 Experiment

4.1 Pretraining

We pretrain our model using the YT-ASL dataset, which contains ASL videos collected from YouTube. Subtitle information is not utilized, and sentence boundary information is assumed to be unavailable. We randomly sample 300 frames from a sequence of 600 consecutive frames (sampled at a rate of 2 frames per unit) during each pretraining step. We explore two masking strategies: full masking and random masking. In full masking, the same time steps are masked across all input streams, denoted as SameMask. In contrast, random masking applies different masked time steps to each stream while maintaining an equal number of masked tokens across streams, denoted as DiffMask.

Hyperparameters The encoder follows a Transformer architecture with $L = 8$, $H = 8$, and a hidden dimension of 512. The decoder uses a smaller Transformer encoder with $L = 4$, $H = 8$, and a hidden dimension of 512. We employ the AdamW optimizer (Loshchilov and Hutter, 2019) with a maximum learning rate of 8×10^{-4} and betas $(0.9, 0.95)$. A learning rate scheduler with warmup and cosine decay is used, with 2K warmup steps. The maximum number of optimization steps is set to 120K. We mask 50% of tokens for each stream in our experiments. Our pretraining takes approximately 14 hours with 8 GH200 GPUs.

4.2 Isolated Sign Language Recognition

Dataset We evaluate effectiveness through ISLR, a classification task that predicts a single gloss from a video clip. Our experiment includes four ISLR datasets: WLASL (Li et al., 2020), ASL Citizen (Desai et al., 2023), Slovo (Kapitanov et al., 2023), and the JSL Corpus (Bono et al., 2014). WLASL, a widely used and challenging ISLR dataset for ASL, serves as the in-domain benchmark. ASL Citizen provides an additional large-scale ASL dataset for evaluation. To assess cross-lingual generalization, we include Slovo and the JSL Corpus, which represent RSL and JSL, respectively. Since the JSL Corpus is not originally designed for ISLR, we extract word-level annotations and exclude non-lexicalized signs, such as classifier constructions, non-manual markers, and mislabeled instances that do not correspond to valid lexical signs.

Finetuning During finetuning, we prepend a learnable [CLS] token to the input pose sequences. The video features are obtained from the contextual embedding of the [CLS] token. We attach a classifier head to the [CLS] token's contextual embedding and optimize it using cross-entropy loss.

Comparison We compare our method with ST-GCN (Yan et al., 2018). We reproduce the result via the implementation from ST-GCN++ (Duan et al., 2022), which is an enhancement of ST-GCN, to provide stronger baseline results. We report top-k recall, where a prediction is considered correct if the target label appears among the top-k results. We evaluate performance with $k = 1, 5$. For WLASL, ASL Citizen and JSL Corpus, we choose the checkpoint with the best validation performance to evaluate on the test sets. For Slovo, which has no test set, we report the performance of the checkpoint with the best top-5 validation recall on the validation set.

4.2.1 Experiment Result

The experimental results are summarized in Table 1. Our model, pretrained on the large-scale YT-ASL dataset, consistently outperforms the pose-based ST-GCN baseline across all four benchmarks. Notably, on WLASL, our approach surpasses other masked reconstruction methods, including SignBERT and MASA. We attribute these improvements to two primary factors. First, pretraining on YT-ASL allows us to leverage a significantly larger and more diverse collection of sign language videos than those available in the public ISLR datasets used by SignBERT and MASA. Second, the separation of each modality stream enables more flexible and effective masking strategies. As shown in Table 1, the DiffMask outperforms SameMask, suggesting that applying different temporal masks to each stream during pretraining contributes to a more robust sign language video encoder.

Effect of Masking Ratio The correlation between the performance and the masking ratio is shown in Figure 2. We can observe that the trends are similar across all datasets. The masking ratio of 0.5 yields the best overall performance, while ratios of 0.3 or 0.7 achieve the second-best results, depending on the dataset. An extremely high ratio, 0.9, leads to performance degradation.

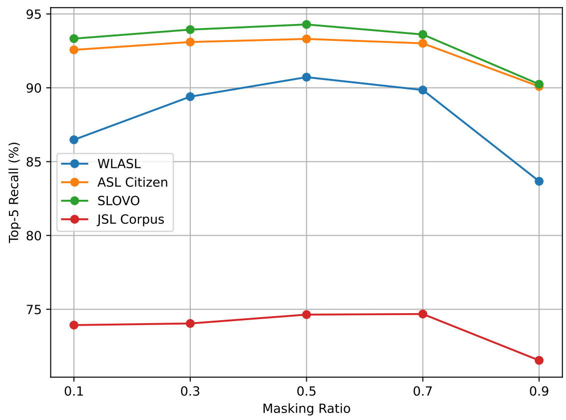


Figure 2: The correlations between top-5 recall and the masking ratio are similar across all ISLR datasets. The best performance is achieved by the masking ratio of 50%. The second best masking ratio is 30% or 70%, depending on the dataset.

4.2.2 Frozen Video Encoder

To further evaluate the pretrained encoder, we conducted experiments by freezing the pretrained video encoder. Specifically, we freeze the pretrained model, apply average pooling to its contextual embeddings, and project the resulting features using a simple trainable linear layer. We utilized the checkpoint with a masking ratio of 0.5 for this experiment. Table 2 summarizes the results.

On the WLASL dataset, our learned representations outperform the baseline model. However, on other datasets, the performance declines. In SLOVO, the performance is slightly below the baseline, while in the ASL Citizen and JSL Corpus datasets, there is a drop of 10 points or more compared to the baseline in top-1 recall. These findings indicate that the learned video encoder is effective without further finetuning.

4.3 Sign Language Translation

We evaluate our approach on three SLT benchmarks: Phoenix14T (P14T) (Camgöz et al., 2018), CSL-Daily (Zhou et al., 2021a), and How2Sign (H2S) (Duarte et al., 2021), representing DGS, CSL and ASL, respectively. In our experiment, we don't use gloss information. We integrate our pretrained sign language video encoder with the mBART translation model (Liu et al., 2020)[1] translation model. We fully finetune the mBART encoder while adapting the decoder using Low-Rank Adaptation (LoRA) (Hu et al., 2022) to avoid overfitting. During training, 20% of video frames are

[1] https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

Table 1: ISLR results across four benchmarks. * denotes the ST-GCN implementation reproduced from previous work. ST-GCN++ is an enhancement of ST-GCN, proposed by (Duan et al., 2022), to provide a stronger baseline. MR denotes Masking Ratio. Our method outperforms previous pose-based self-supervised learning approaches.

Method	WLASL		ASL Citizen		Slovo		JSL Corpus	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ST-GCN* (Yan et al., 2018)	34.40	66.57	63.10	86.09				
ST-GCN++ (Duan et al., 2022)	41.70	74.36	70.67	90.72	64.94	87.71	46.17	70.87
SignBERT (Zhou et al., 2021b)	47.46	83.32						
MASA (Zhao et al., 2024)	49.06	82.90						
Ours (DiffMask, MR=0.5)	56.95	90.72	75.72	93.31	74.98	94.29	52.40	74.64
Ours (SameMask, MR=0.5)	52.05	87.21	71.87	91.23	72.24	94.14	51.26	72.43

Table 2: Result of freezing the pretrained model with a masking ratio of 50%. The results show that the pretrained model is effective even without further finetuning, although in most cases, the performance lags behind the baseline model.

Dataset	Method	Split	Rec@1	Rec@5
WLASL	ST-GCN++	test	41.70	74.36
WLASL	probe	test	42.88	74.77
ASL Citizen	ST-GCN++	test	70.67	90.72
ASL Citizen	probe	test	54.54	79.45
SLOVO	ST-GCN++	valid	64.94	87.71
SLOVO	probe	valid	60.12	85.61
JSL Corpus	ST-GCN++	test	46.17	70.87
JSL Corpus	probe	test	37.68	62.19

randomly deleted or copied as temporal augmentation. We experiment with both freezing and finetuning the pretrained video encoder. Gradient clipping is employed to stabilize the training. Details of the hyperparameter setup can be found shown in Appendix C.

We report BLEU scores (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics to evaluate translation quality. Specifically, we compute BLEU-1 and BLEU-4 using SacreBLEU (Post, 2018)[2], and report the ROUGE-L F1 score[3].

We compare our model against recent gloss-free approaches. For the P14T and CSL-Daily datasets, we evaluate performance relative to Sign2GPT (Wong et al., 2024), VAP (Jiao et al., 2024), C^{2}RL (Chen et al., 2024), and Sign-LLMs (Gong et al., 2024), which are language-supervised pretraining methods. For the How2Sign dataset, we compare our results with SSVP-SLT, an MAE-based method on RGB modality, and T5 models pretrained on YT-ASL with subtitle supervision (Uthus et al., 2023).

We explore two input strategies: (1) Flat concatenation: Tokens from all three input streams are concatenated into a single sequence and passed to mBART. (2) Per-time-step averaging: At each time step, embeddings from the three streams are averaged to produce a single fused embedding per time step. The resulting sequence is input to mBART.

4.3.1 Experimental Results

Results for P14T and CSL-Daily are shown in Table 3, and results for How2Sign are shown in Table 5. On CSL-Daily, our method outperforms Sign2GPT using only skeleton data. On How2Sign, it matches the performance of SSVP-SLT without vision-language alignment, while being more lightweight and computationally efficient, with only 14 hours for training and skeletal data as the input. Compared to T5 with supervised pretraining on YT-ASL, our model achieves comparable performance without relying on the subtitle data.

While the performance on P14T is weaker, we attribute this to the dataset's low video resolution and motion blur, which leads to inaccurate keypoint estimation. The pose quality gap between finetuning and pretraining stages may hurt the performance. This highlights a key limitation of skeleton-based pretraining: its reliance on high-quality pose data. The skeleton quality between pretraining and finetuning should be aligned.

While our encoder does not surpass all prior methods, it demonstrates the effectiveness of our method. It shows that the video encoder pretrained on only ASL videos can be generalized to other SLs. The flat concatenation of stream features shows similar performance to per-time-step averaging. Our following experiments will use per-time-step averaging as the default setup, because of its lower computational cost.

[2]For Chinese, we use the 'zh' tokenizer; for English and German, we use the '13a' tokenizer

[3]We adopted the ROUGE implementation from the official codebase of TwoStreamSLT (Chen et al., 2022b)

Table 3: Experimental results on P14T and CSL-Daily. Following the observation in (Jiao et al., 2024), the mBART tokenizer exhibits an inconsistent punctuation bug, particularly affecting evaluations in Chinese due to the use of full-width punctuation marks. To ensure a fair comparison, we report the results after correcting the bug, with the uncorrected results shown in parentheses.

		P14T			CSL-Daily		
Method	Modality	B1	B4	R	B1	B4	R
Sign2GPT (Wong et al., 2024)	RGB	45.43	19.42	45.23	34.80	12.96	41.12
Sign2GPT(Pseudo-Gloss Pretraining) (Wong et al., 2024)	RGB	49.54	22.52	48.90	41.75	15.40	42.36
VAP (Jiao et al., 2024)	Skeleton	53.07	26.16	51.28	52.98(49.99)	23.65(20.85)	51.09(48.56)
SignLLMs (Gong et al., 2024)	RGB	45.21	23.40	44.49	39.55	15.75	39.91
C^2RL (Chen et al., 2024)	RGB	52.81	26.75	50.96	49.32	21.61	48.21
Ours (Flat Concatenation)	Skeleton	44.46	20.69	42.64	52.73(49.75)	22.83(20.04)	50.72(48.21)
Ours (Per-time-step averaging)	Skeleton	45.76	21.58	43.45	53.14(50.15)	22.83(20.10)	50.56(48.05)

4.4 Analysis

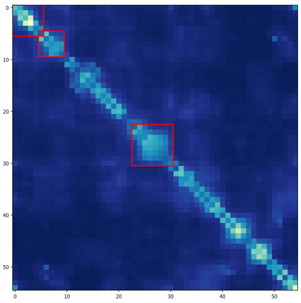
4.4.1 Facial Information

Facial information plays a critical role in sign language understanding (Mukushev et al., 2020; Chaudhary et al., 2024). Facial expressions often serve grammatical purposes, while mouthing can help disambiguate signs that share similar manual gestures. However, it remains unknown whether facial information in ASL can also benefit understanding in other sign languages.

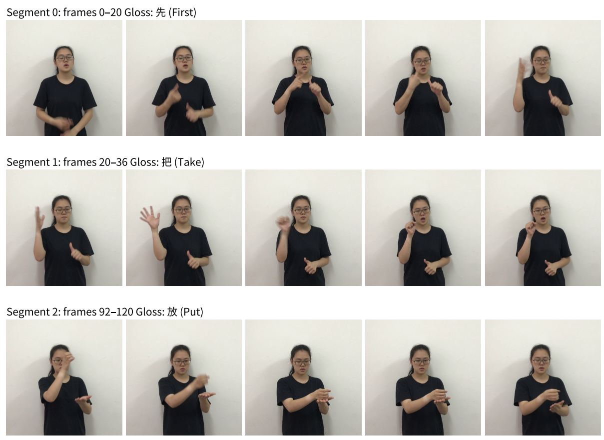
Table 4: Results of varying stream setups during the pretraining and finetuning stages, denoted as PT and FT in the header. B, H, and F represent body, hands, and face, respectively. The best performance for each dataset is highlighted in bold.

Dataset	PT	FT	B1	B4	R
P14T	B,H	B,H	45.76	21.58	43.45
	B,H	B,H,F	44.92	21.28	42.86
	B,H,F	B,H,F	**47.29**	**22.71**	**44.86**
CSL-Daily	B,H	B,H	53.14	22.83	50.56
	B,H	B,H,F	52.75	22.46	50.00
	B,H,F	B,H,F	**54.30**	**24.09**	**52.03**
H2S	B,H	B,H	33.66	11.14	**28.70**
	B,H	B,H,F	38.47	11.94	26.78
	B,H,F	B,H,F	**39.71**	**12.64**	27.85

To investigate the impact of facial information, we experiment with different configurations for incorporating facial keypoints during pretraining and finetuning. The results are presented in Table 4. Through this result, we find that incorporating facial information into both pretraining and finetuning stages can bring improvement of around 1 to 2 points on BLEU-4 across all SLs, although the gains are not particularly significant.



(a) Symmetric attention matrix obtained by summing the original attention weights and their transpose. Red rectangles denote the clusters identified in the attention map.



(b) Video clip segmented based on the clusters found in attention weight.

Figure 3: An example from the CSL-Daily validation set illustrating successful segmentation based on the attention weights of our pretrained model.

4.4.2 Attention Map Visualization

We have demonstrated that pretraining on large-scale ASL video datasets benefits sign language processing tasks across different sign languages. In this section, we conduct a preliminary analysis to better understand the effectiveness of this transfer learning. While performance gains on ISLR

Table 5: Experiment results on How2Sign. VLA denotes Vision-Language Alignment pretraining. SSL denotes Self-Supervised Learning.

Method	Modality	VLA	SSL	Translation Data	B1	B4	R
T5 (Uthus et al., 2023)	Skeleton			H2S	14.96	1.22	
T5 (Uthus et al., 2023)	Skeleton			YT-ASL→ H2S	37.82	12.39	-
SSVP-SLT (Rust et al., 2024)	RGB	✓	YT-ASL + H2S	YT-ASL + H2S	43.2	15.5	38.4
SSVP-SLT (Rust et al., 2024)	RGB		YT-ASL	H2S	38.1	11.7	33.8
VAP (Jiao et al., 2024)	Skeleton	✓		H2S	39.22	12.87	27.77
C^2RL (Chen et al., 2024)	RGB	✓		H2S	29.07	9.37	27.02
SHuBERT (Gueuwou et al., 2024)	RGB + Skeleton		YT-ASL	YT-ASL→ H2S	-	16.2	-
Ours (Flat Concatenation)	Skeleton		YT-ASL	H2S	33.66	11.14	28.70
Ours (Per-time-step averaging)	Skeleton		YT-ASL	H2S	32.09	10.45	28.22

tasks suggest improved motion discrimination after pretraining, they do not fully account for the model's enhanced ability to process longer, multi-sign sentences lacking clear boundaries. To explore this further, we examine the attention patterns of our pretrained model on CSL sentence examples, aiming to uncover how pose tokens interact.

Specifically, we extract intra-stream attention weights from the final layer's self-attention matrices, average them across all heads and streams, and then symmetrize the result by adding its transpose. An example of the resulting symmetric attention map is shown in Figure 3a. We observe that several prominent blocks appear along the diagonal, indicating that the model groups temporally adjacent frames into clusters while also exhibiting boundaries between them. Further experimental details and examples are provided in Appendix F.

When we segment video clips based on these identified clusters from the attention weight, we find that many of the resulting segments roughly correspond to individual signs, as illustrated in Figure 3b. This suggests that the pretrained model tries to segment sign language sentences, which is important in sign language understanding.

Some segments correspond to individual signs, suggesting that the model may be learning structural patterns within signs or patterns of transitions between signs. SLs often exhibit shared motion characteristics, such as indexical signs or other lexicalized forms discussed in (Wei and Chen, 2023), which may facilitate such grouping. Additionally, despite variation across different sign languages, many signs share similar movement features, which can serve as cross-linguistic cues.

The model may also be leveraging implicit knowledge of sign boundaries acquired from ASL training data, potentially influenced by prosodic features (Fenlon et al., 2007), allowing it to detect natural break points within signing sequences.

However, segmentation is not always accurate. For instance, we observe that signs consisting of two sub-actions are sometimes split into separate segments. In cases where the signer signs quickly and naturally, multiple signs may be merged into a single group.

Despite these limitations, the findings offer valuable insight into the model's learned knowledge that contributes to sign language understanding and highlight the potential of using attention weights from pretrained models for sign segmentation, which is a low-resource yet important task in applications such as sign spotting.

5 Conclusion

In this paper, we explored leveraging ASL videos to improve performance in other sign languages. We introduced MS-MAE, a simple yet effective pretraining framework that concatenates multiple skeleton streams along the temporal dimension. Experimental results demonstrate that pretraining solely on ASL videos significantly boosts performance in both ISLR and SLT tasks across different sign languages. In ISLR, our approach outperforms other methods trained only on the target data. For SLT, it achieves performance comparable to state-of-the-art gloss-free, RGB-based methods after full fine-tuning, validating the effectiveness of our strategy. Furthermore, attention visualization reveals that the pretrained model naturally groups neighboring frames into clusters, some of which align with specific signs, in cross-lingual sentences. This provides qualitative evidence that the MAE may learn to segment sign language sentences. A more comprehensive quantitative analysis, beyond the case study, is left for future work.

Limitations

In our proposed pretraining framework, separating visual cues results in significantly longer input sequences, which increases the complexity of the transformer due to the quadratic nature of the self-attention mechanism. Although we have not yet conducted specific experiments to validate this, we hypothesize that without sufficient data, training such a framework effectively would be difficult. Besides, as mentioned in Section 4, the pose quality gap between pretraining and finetuning may lead to performance degradation, which is the inherent issue of skeleton-based methods. One future direction is to improve the robustness to noisy keypoints. Additionally, although skeletal modalities can substantially reduce computational demands during both pretraining and finetuning, they require extra preprocessing time to extract pose data.

Regarding our experiments, we acknowledge that the evaluation did not encompass a sufficiently diverse range of sign language categories, primarily due to the limited availability of datasets and computational resources. As a result, we were unable to thoroughly investigate the factors that contribute to improved cross-lingual transferability, and thus could not provide concrete guidelines for future work. Additionally, existing benchmarks are built under varying conditions, making it difficult to isolate the specific factors that influence model performance. For example, we did not control for confounding variables such as video quality, dataset scale, and dataset difficulty, which may have limited the strength and generalizability of our conclusions. In our future work, we will conduct more comprehensive experiments on other datasets.

Acknowledgments

References

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6816–6826. IEEE.

Jordan J Bird, Anikó Ekárt, and Diego R Faria. 2020. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors*, 20(18):5151.

Mayumi Bono, Kouhei Kikuchi, Paul Cibulka, and Yutaka Osugi. 2014. A colloquial corpus of Japanese Sign Language: Linguistic resources for observing sign language conversations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1898–1904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7784–7793. Computer Vision Foundation / IEEE Computer Society.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. *CoRR*, abs/2003.13830.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733. IEEE Computer Society.

Lipisha Chaudhary, Fei Xu, and Ifeoma Nwogu. 2024. Cross-attention based influence model for manual and nonmanual sign language analysis. In *Pattern Recognition - 27th International Conference, ICPR 2024, Kolkata, India, December 1-5, 2024, Proceedings, Part XXI*, volume 15321 of *Lecture Notes in Computer Science*, pages 372–386. Springer.

Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. 2022a. A simple multi-modality transfer learning baseline for sign language translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5110–5120. IEEE.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie Liu, and Brian Mak. 2022b. Two-stream network for sign language recognition and translation. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zhigang Chen, Benjia Zhou, Yiqing Huang, Jun Wan, Yibo Hu, Hailin Shi, Yanyan Liang, Zhen Lei, and Du Zhang. 2024. C^{2}rl: Content and context representation learning for gloss-free sign language translation and retrieval. *CoRR*, abs/2408.09949.

Aashaka Desai, Lauren Berger, Fyodor Minakov, Nessa Milano, Chinmay Singh, Kriston Pumphrey, Richard E. Ladner, Hal Daumé III, Alex X. Lu, Naomi Caselli, and Danielle Bragg. 2023. ASL citizen: A community-sourced dataset for advancing isolated sign language recognition. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. 2022. PYSKL: towards good practices for skeleton action recognition. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 7351–7354. ACM.

Amanda Cardoso Duarte, Shruti Palaskar, Lucas Ventura, Deepti Ghadiyaram, Kenneth DeHaan, Florian Metze, Jordi Torres, and Xavier Giró-i-Nieto. 2021. How2sign: A large-scale multimodal dataset for continuous american sign language. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2735–2744. Computer Vision Foundation / IEEE.

Jordan Fenlon, Tanya Denmark, Ruth Campbell, and Bencie Woll. 2007. Seeing sentence boundaries. *Sign Language & Linguistics*, 10(2):177–200.

Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. 2024. Llms are good sign language translators. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 18362–18372. IEEE.

Shester Gueuwou, Xiaodan Du, Greg Shakhnarovich, Karen Livescu, and Alexander H. Liu. 2024. Shubert: Self-supervised sign language representation learning via multi-stream cluster prediction. *CoRR*, abs/2411.16765.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. 2022. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE.

Ruth Holmes, Ellen Rushe, Mathieu De Coster, Maxim Bonnaerens, Shinichi Satoh, Akihiro Sugimoto, and Anthony Ventresque. 2023. From scarcity to understanding: Transfer learning for the extremely low resource irish sign language. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023 - Workshops, Paris, France, October 2-6, 2023*, pages 2000–2009. IEEE.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Hezhen Hu, Weichao Zhao, Wengang Zhou, and Houqiang Li. 2023. Signbert+: Hand-model-aware self-supervised pre-training for sign language understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(9):11221–11239.

Zifan Jiang, Gerard Sant, Amit Moryossef, Mathias Müller, Rico Sennrich, and Sarah Ebling. 2024. SignCLIP: Connecting text and sign language by contrastive learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9171–9193, Miami, Florida, USA. Association for Computational Linguistics.

Peiqi Jiao, Yuecong Min, and Xilin Chen. 2024. Visual alignment pre-training for sign language translation. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLII*, volume 15100 of *Lecture Notes in Computer Science*, pages 349–367. Springer.

Alexander Kapitanov, Karina Kvanchiani, Alexander Nagaev, and Elizaveta Petrova. 2023. Slovo: Russian sign language dataset. In *Computer Vision Systems: 14th International Conference, ICVS 2023, Vienna, Austria, September 27-29, 2023, Proceedings*, volume 14253 of *Lecture Notes in Computer Science*, pages 63–73. Springer.

Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1448–1458. IEEE.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172.

Medet Mukushev, Arman Sabyrov, Alfarabi Imashev, Kenessary Koishybay, Vadim Kimmelman, and

Anara Sandygulova. 2020. Evaluation of manual and non-manual components for sign language recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6073–6078, Marseille, France. European Language Resources Association.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Phillip Rust, Bowen Shi, Skyler Wang, Necati Cihan Camgoz, and Jean Maillard. 2024. Towards privacy-aware sign language translation at scale. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8624–8641, Bangkok, Thailand. Association for Computational Linguistics.

Noha A. Sarhan and Simone Frintrop. 2020. Transfer learning for videos: From action recognition to sign language recognition. In *IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020*, pages 1811–1815. IEEE.

Garrett Tanzer and Biao Zhang. 2024. Youtube-sl-25: A large-scale, open-domain multilingual sign language parallel corpus. *CoRR*, abs/2407.11144.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093.

David Uthus, Garrett Tanzer, and Manfred Georg. 2023. Youtube-asl: A large-scale, open-domain american sign language-english parallel corpus. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Fangyun Wei and Yutong Chen. 2023. Improving continuous sign language recognition with cross-lingual signs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23612–23621.

Ryan Wong, Necati Cihan Camgöz, and Richard Bowden. 2024. Sign2gpt: Leveraging large language models for gloss-free sign language translation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2025. Signrep: Enhancing self-supervised sign representations. *arXiv preprint arXiv:2503.08529*.

Wenhan Wu, Yilei Hua, Ce Zheng, Shiqian Wu, Chen Chen, and Aidong Lu. 2023. Skeletonmae: Spatial-temporal masked autoencoders for self-supervised skeleton action recognition. In *IEEE International Conference on Multimedia and Expo Workshops, ICMEW Workshops 2023, Brisbane, Australia, July 10-14, 2023*, pages 224–229. IEEE.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, volume 11219 of *Lecture Notes in Computer Science*, pages 318–335. Springer.

Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 7444–7452. AAAI Press.

Weichao Zhao, Hezhen Hu, Wengang Zhou, Yunyao Mao, Min Wang, and Houqiang Li. 2024. MASA: motion-aware masked autoencoder with semantic alignment for sign language recognition. *IEEE Trans. Circuits Syst. Video Technol.*, 34(11):10793–10804.

Weichao Zhao, Hezhen Hu, Wengang Zhou, Jiaxin Shi, and Houqiang Li. 2023. BEST: BERT pre-training for sign language recognition with coupling tokenization. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 3597–3605. AAAI Press.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021a. Improving sign language translation with monolingual data by sign back-translation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1316–1325. Computer Vision Foundation / IEEE.

Zhenxing Zhou, Vincent W. L. Tam, and Edmund Y. Lam. 2021b. Signbert: A bert-based deep learning

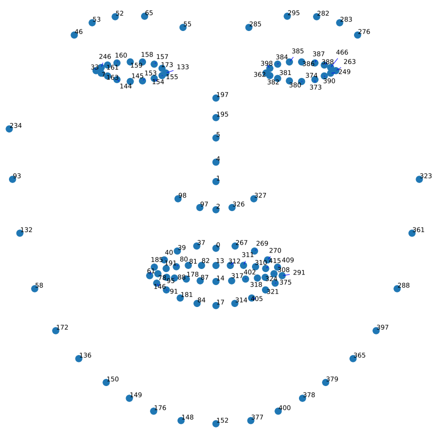


Figure 4: Face keypoints used in our experiments

framework for continuous sign language recognition. *IEEE Access*, 9:161669–161682.

A Keypoints

In our experiments, we use MediaPipe Holistic with the complexity of 2 for pose estimation and extract the following keypoints:

1. **Hands:** All 21 keypoints of each hand (indices 0–20).

2. **Body:** Upper-body keypoints with indices $\{11, 12, 13, 14, 15, 16\}$.

3. **Face:** Includes keypoints from the contour, mouth, nose, and eyes:

   Contour `234, 93, 132, 58, 172, 136, 150, 149, 176, 148, 152, 377, 400, 378, 379, 365, 397, 288, 361, 323`

   Mouth `0, 267, 269, 270, 409, 291, 375, 321, 405, 314, 17, 84, 181, 91, 146, 61, 185, 40, 39, 37, 13, 312, 311, 310, 415, 308, 324, 318, 402, 317, 14, 87, 178, 88, 95, 78, 191, 80, 81, 82`

   Nose `98, 97, 2, 326, 327, 1, 4, 5, 195, 197`

   Eyes `46, 53, 52, 65, 55, 285, 295, 282, 283, 276, 33, 246, 161, 160, 159, 158, 157, 173, 133, 155, 154, 153, 145, 144, 163, 7, 362, 398, 384, 385, 386, 387, 388, 466, 263, 249, 390, 373, 374, 380, 381, 382`

Table 7: Statistics of the SLT datasets used in our experiments. For the H2S dataset, we use the manually re-aligned version provided on their homepage and exclude a very small subset of samples due to invalid time ranges.

Dataset	P14T	CSL-Daily	H2S
# Train	7,096	18,401	31,086
# Valid	519	1,077	1,738
# Test	642	1,176	2,349

An example showing face keypoints is shown in Figure 4.

B Dataset Statistics

The statistical information of the ISLR datasets used in our experiments is shown in Table 6, including WLASL, ASL Citizen, Slovo and JSL Corpus. The statistical information of the SLT datasets is shown in Table 7.

Table 6: Statistics of the used ISLR datasets.

Dataset	WLASL	ASL Citizen	Slovo	JSL Corpus
Gloss	2,000	2,731	1,001	696
Train	14,289	40,154	15,300	32,282
Valid	3,916	10,304	5,100	4,306
Test	2,878	32,941		4,676

C Hyperparameter Setup for Downstream Task

ISLR We set the batch size to 128 and sample 32 frames per video as input. We use AdamW optimizer with a weight decay of 10^{-3}. A cosine learning rate scheduler is used with a 10-epoch linear warm-up and a peak learning rate of 5×10^{-5}. Training is conducted for 100 epochs. During training, we apply temporal augmentation by randomly sampling frames from each video. We also augment the pose data by randomly rotating, shearing, and scaling, as suggested by SignCLIP (Jiang et al., 2024), on all datasets except JSL Corpus.

SLT We use LoRA with hyperparameters $\alpha = 32$ and $r = 32$. The training objective is cross-entropy loss. We employ the AdamW optimizer with a weight decay of 10^{-3}, and apply a cosine learning rate schedule with a 10-epoch warmup. We train for up to 100 epochs with a batch size of 32, applying gradient clipping to stabilize optimization.

D Computational Resource Usage

We conducted pretraining on 8 nodes, each equipped with an NVIDIA GH200 GPU, for ap-

Table 8: Learning rates and gradient clipping norms for each dataset and encoder status.

Video Encoder Status	P14T & CSL-Daily Frozen	P14T & CSL-Daily Unfrozen	H2S Frozen	H2S Unfrozen
Learning Rate	3×10^{-4}	1×10^{-4}	3×10^{-4}	5×10^{-5}
Gradient Clipping	1.0	0.1	1.0	

proximately 14 hours. To ensure convergence, we used a total of 120,000 training steps. Our in-house experiments show that the checkpoint at 60% of training steps achieved performance comparable to the final checkpoint.

E Frozen Video Encoder in SLT

In this section, we present experimental results examining the effect of freezing the video encoder during fine-tuning, in order to more comprehensively demonstrate the strengths of our pretrained model. We evaluate two pretrained variants with a masking ratio of 0.5—one incorporating facial information and one without. The hyperparameter configuration closely follows that described in Appendix C, with the exception of learning rates and gradient clipping settings. Because optimal performance varies depending on whether the video encoder is frozen, we perform a grid search over learning rates and report the best-performing configuration based on validation set performance. The setup is shown in Table 8. For simplicity, we fix the gradient clipping norm to 1.0 across all experiments. The result is shown in Table 9. Without further finetuning, the video encoder has fairly good ability, with around 2 points lower.

Table 9: Results of varying stream setups during the pretraining and finetuning stages, denoted as PT and FT in the header. B, H, and F represent body, hands, and face, respectively. The best performance for each dataset is highlighted in bold.

Dataset	PT	FT	Frozen	B1	B4	R
P14T	B,H	B,H		45.76	21.58	43.45
	B,H	B,H	✓	42.37	19.25	40.47
	B,H,F	B,H,F		**47.29**	**22.71**	**44.86**
	B,H,F	B,H,F	✓	44.25	20.89	42.52
CSL-Daily	B,H	B,H		53.14	22.83	50.56
	B,H	B,H	✓	49.48	20.50	47.57
	B,H,F	B,H,F		**54.30**	**24.09**	**52.03**
	B,H,F	B,H,F	✓	53.22	22.73	50.14
H2S	B,H	B,H		33.66	11.14	**28.70**
	B,H	B,H	✓	33.48	9.46	25.75
	B,H,F	B,H,F		**39.71**	**12.64**	27.85
	B,H,F	B,H,F	✓	33.51	9.95	26.92

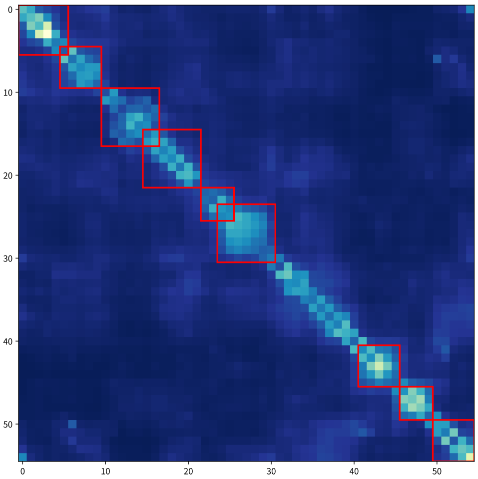


Figure 5: The same attention matrix with Figure 3a, drawed with manual segmentation. The resulting video segmentation is visualized in Figure 7.

F Attention Visualization

In this session, we provide more visualization examples of attention weights in our pretrained models. Given a pose sequence, we feed it into the pretrained model and extract the attention weight from the last layer $A_i \in \mathbb{R}^{3n \times 3n}$, where $i < h$ is the index of head, h is the number of heads, n is the length of the sequence, and 3 denotes the number of streams. We obtain the average attention weight across heads $A_{\text{avg}} = 1/h \sum_i^h A_i$. We denote the indices for body, left hand, and right hand as I_b, I_{lh}, and I_{rh}. The intra-stream attention weights are obtained by selecting the corresponding submatrices from A_{avg}:

$$A^B = A_{\text{avg}}[I_b, I_b], \tag{4}$$

$$A^{LH} = A_{\text{avg}}[I_{lh}, I_{lh}], \tag{5}$$

$$A^{RH} = A_{\text{avg}}[I_{rh}, I_{rh}]. \tag{6}$$

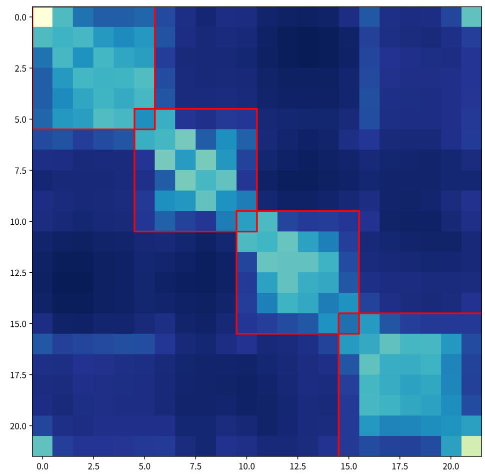
We obtained the average attention across all streams as $\hat{A} = \frac{1}{3}(A^B + A^{LH} + A^{RH})$. Finally, the symmetric attention, $\hat{A}_{\text{sym}}$, is obtained by averaging $\hat{A}$ with its transpose:

$$\hat{A}_{\text{sym}} = \frac{1}{2}(\hat{A} + \hat{A}^T).$$

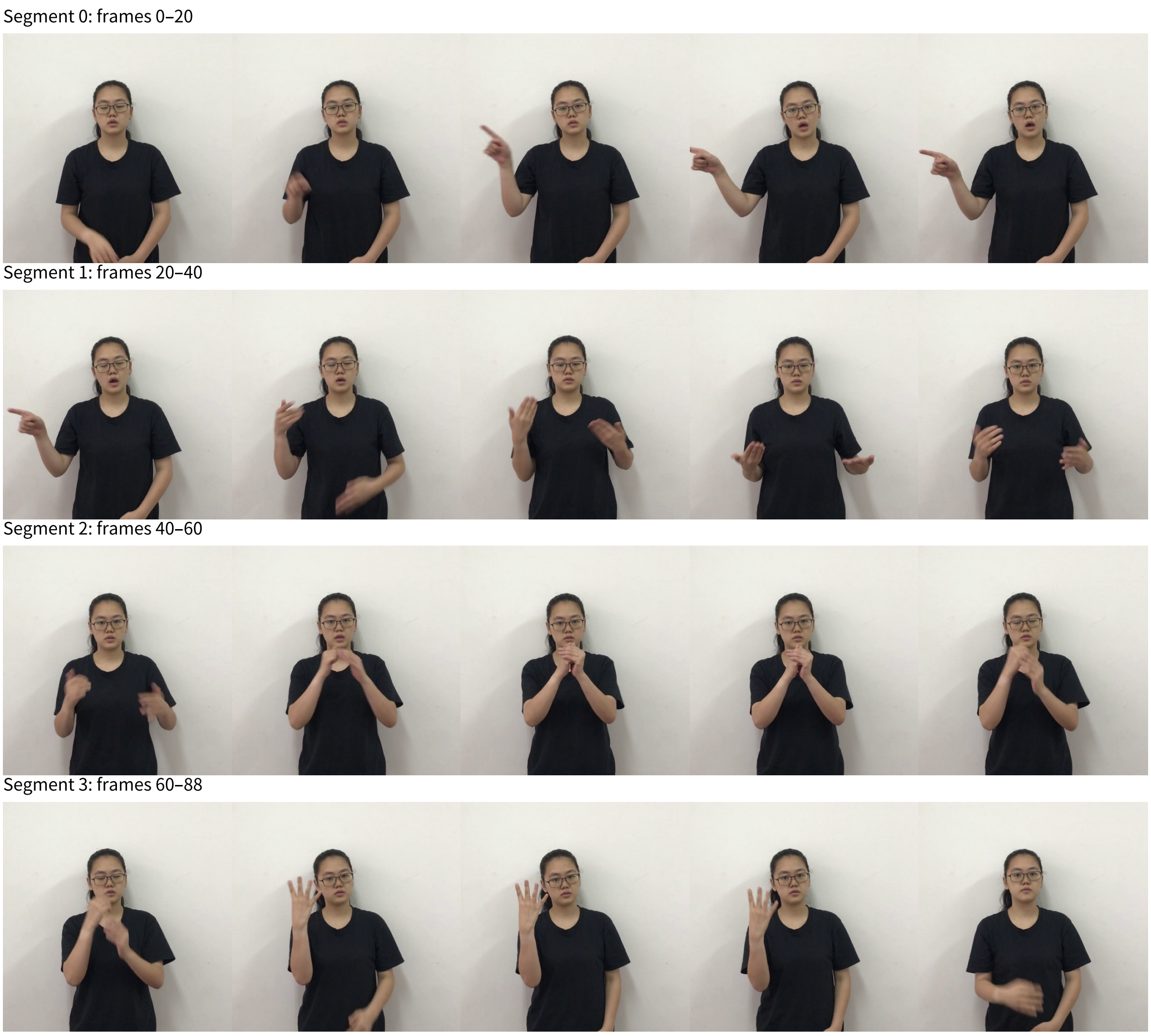
Segmentation is performed based on the symmetric attention. We manually choose reasonable segments from the attention weight. In our pretrained model, each token represents four frames; thus, the frame index is computed by multiplying the token index by four. Attention matrix of CSL are shown in Figure 6a and 5. Corresponding segmented videos are shown in Figure 6b and 7.

G Use of AI Assistance

In this research, we primarily used GitHub Copilot for coding and debugging, and ChatGPT for refining the writing of this paper.



(a) $\hat{A}_{\text{sym}}$ for the example.



(b) Segmentation result based on attention weight of video sample with gloss Ta(He) JinTian(Today) NianLing(Age) 4.

Figure 6: A sample from CSL-Daily demonstrating good alignment between attention-based segmentation and ground-truth gloss sequence.

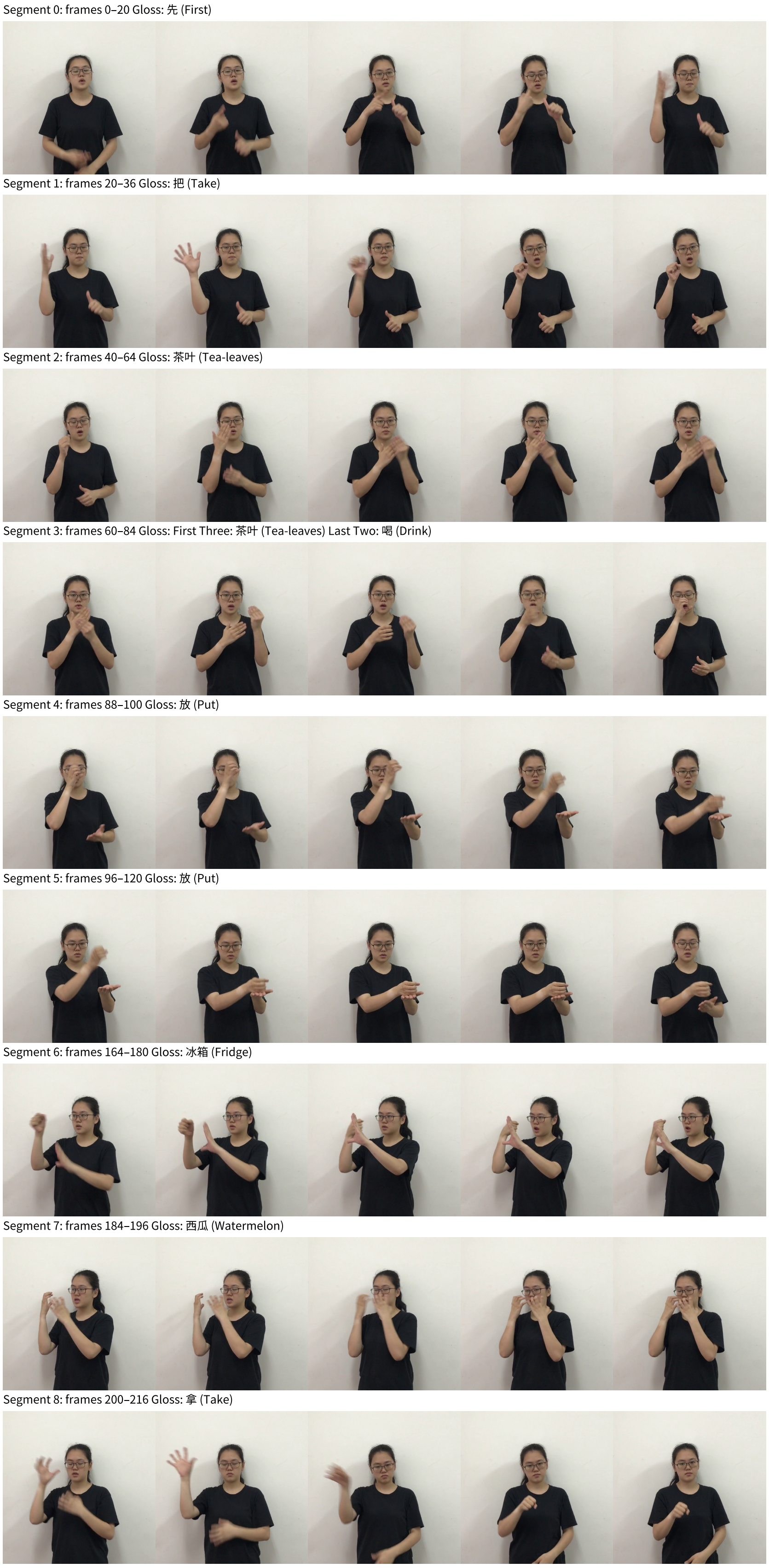
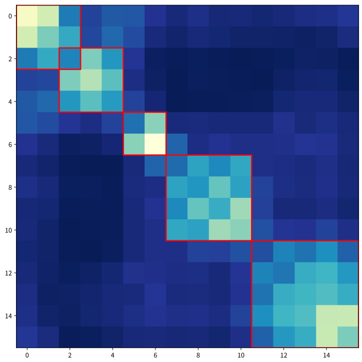
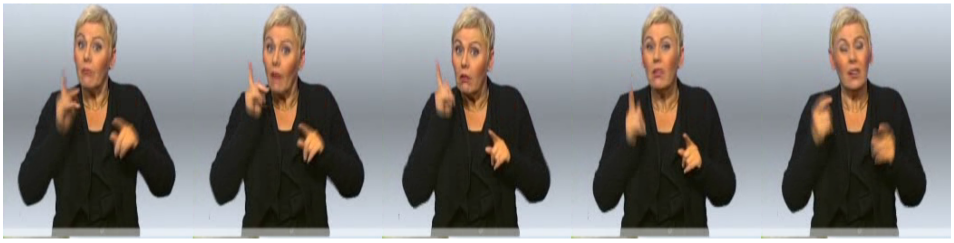


Figure 7: The same sample from CSL-Daily with Figure 3b. The GT gloss sequence is Xian(first) Ba(pick up/take) Chaye(tea-leaves) He(drink) Fang(put) Huan(return) BingXiang(fridge) XiGua(watermelon) Na(take). Sign 'Chaye Tea-leaves' is segmented into two parts based on the attention matrix, possibly because it is a repetitive motion. Fang(put) is also segmented, due to a long holding motion at the end.



(a) $\hat{A}_{\text{sym}}$. We manually choose reasonable boundaries.

Segment 0: frames 0–8



Segment 1: frames 8–16

Segment 2: frames 20–24

Segment 3: frames 28–40

Segment 4: frames 44–60

(b) The sample with gloss sequence of DANN(THEN) STARK(STRONG) SCHNEE(SNOW) SCHNEIEN(SNOWING) KOMMEN(COME). The segments align fairly well with the gloss sequence.

Figure 8: A sample of P14T.