SignBERT: Pre-Training of Hand-Model-Aware Representation for Sign Language Recognition

Hezhen Hu [1], Weichao Zhao [1*], Wengang Zhou [1,2†], Yuechen Wang [1], Houqiang Li [1,2†]

[1] CAS Key Laboratory of GIPAS, EEIS Department, University of Science and Technology of China (USTC)

[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center

{alexhu, saruka, wyc9725}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn

Abstract

Hand gesture serves as a critical role in sign language. Current deep-learning-based sign language recognition (SLR) methods may suffer insufficient interpretability and overfitting due to limited sign data sources. In this paper, we introduce the first self-supervised pre-trainable SignBERT with incorporated hand prior for SLR. Sign-BERT views the hand pose as a visual token, which is derived from an off-the-shelf pose extractor. The visual tokens are then embedded with gesture state, temporal and hand chirality information. To take full advantage of available sign data sources, SignBERT first performs self-supervised pre-training by masking and reconstructing visual tokens. Jointly with several mask modeling strategies, we attempt to incorporate hand prior in a model-aware method to better model hierarchical context over the hand sequence. Then with the prediction head added, SignBERT is fine-tuned to perform the downstream SLR task. To validate the effectiveness of our method on SLR, we perform extensive experiments on four public benchmark datasets, i.e., NMFs-CSL, SLR500, MSASL and WLASL. Experiment results demonstrate the effectiveness of both self-supervised learning and imported hand prior. Furthermore, we achieve state-of-the-art performance on all benchmarks with a notable gain.

1. Introduction

Sign language, as a visual language, is the primary communication tool for the deaf community. To facilitate the communication between the deaf and hearing people, sign language recognition (SLR) has been widely studied with broad social influence. Isolated SLR serves as a fundamental task in visual sign language research. It aims to recognize sign language at the word-level and is a challenging fine-grained classification problem.

---

*Contribute equally with the first author.

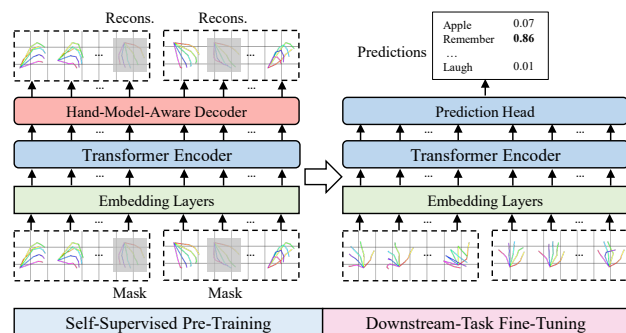†Corresponding author: Wengang Zhou and Houqiang Li.

Figure 1. The overview of our framework, which contains self-supervised pre-training and downstream-task fine-tuning.

Hand gesture serves as a dominant role during the expression of sign language. It occupies a relatively small area with dynamic backgrounds, exhibits similar appearance and encounters self-occlusion among joints. Such fact leads to the difficulty in hand representation learning. Current deep-learning-based methods [5, 28, 23] learn feature representations adaptively from the cropped RGB hand sequence. Given the highly articulated characteristic of hand, some methods represent them as sparse poses for recognition [1, 34, 24]. Pose is a compact and semantic representation, which is robust to appearance change and brings potential computation efficiency. However, hand poses are usually extracted from the off-the-shelf extractor, which suffers failure detection. Therefore, the performance of pose-based methods lags largely behind RGB-based counterparts. Besides, the aforementioned methods all follow a data-driven paradigm and may suffer insufficient interpretability and overfitting due to limited sign data sources.

Meanwhile, the effectiveness of pre-training has been validated for computer vision (CV) and natural language processing (NLP). Recent advance in NLP is largely derived from self-supervised pre-training strategies on large text corpus [43, 14, 56]. Among them, BERT [14] is one of the most popular methods due to its simplicity and superior performance. Its success is largely attributed to the power-

ful attention-based Transformer backbone [53], jointly with a well-designed pre-training strategy for modeling context inherent in text sequence.

To tackle the aforementioned issues, we develop a self-supervised pre-trainable framework with model-aware hand prior incorporated, namely SignBERT, as shown in Figure 1. Considering the compactness and expressiveness of hand pose representation, we view hand pose as a visual token. Each hand token is embedded with gesture state, temporal and hand chirality information, and both hands are involved as input. SignBERT first performs self-supervised pre-training on a large volume of hand pose data, which is derived from sign language data sources using the off-the-shelf extractor. Specifically, inspired by BERT [14], we pre-train our framework on the encoder-decoder backbone by masking and reconstructing visual tokens. We design several mask modeling strategies to enforce the network capturing hierarchical contextual information. To better capture context and ease optimization, the decoder introduces hand prior in a model-aware method. For the downstream isolated SLR, the pre-trained encoder is fine-tuned with the added prediction head to perform recognition.

Our contributions are summarized as follows,

- To our best knowledge, we propose the *first* model-aware pre-trainable framework for sign language recognition, namely SignBERT. It performs self-supervised learning on a large volume of hand pose data for better performance on the downstream task.
- To better exploit hierarchical contextual information contained in the sign data sources, we design mask modeling strategies and incorporate model-aware hand prior during self-supervised pre-training.
- We perform extensive experiments to validate the feasibility of our framework and its effectiveness on the downstream SLR task. Our method achieves state-of-the-art performance on four popular benchmarks, *i.e.,* NMFs-CSL, SLR500, MSASL and WLASL.

2. Relate Work

In this section, we will briefly review the related topics, including sign language recognition, pre-training strategy and hand-modeling technique.

2.1. Sign Language Recognition

Previous works [27] on sign language recognition are generally grouped into two categories based on the input modality, *i.e.*, RGB-based (using the RGB video) and pose-based (using the pose sequence) methods.

RGB-based methods. With the strong representation capability of CNNs, many works in SLR adopt it as the backbone [10, 29, 24, 59]. Necati *et al.* [6] introduce a network consisting of 2D-CNNs for spatial representation and

Transformer for modeling temporal dependencies by supervised learning. Some other works [22, 24, 33, 34, 1] utilize 3D-CNNs for modeling spatio-temporal information.

Pose-based methods. As compact and semantic-aware data, pose sequences are processed by CNNs [32, 7, 1] or RNNs [15, 37, 45]. Considering its well-structured nature, more and more works represent it as a graph and adopt graph convolutional networks (GCNs) to model its representation [15, 45, 51]. Yan *et al.* [55] first propose a spatial-temporal GCN for action recognition. These GCN-based methods show both efficiency and promising performance. There also exists work combining Transformer without pre-training for SLR [51].

2.2. Pre-Training Strategy

Pre-training, a common strategy in NLP and CV, produces more generic feature representation and may alleviate overfitting for target tasks. In NLP tasks, early works focused on improving word embedding [40, 26]. With the advance of Transformer [53], many works propose to pre-train generic feature representations [14, 43, 56]. Of them, BERT is one of the most popular methods due to its simplicity and superior performance. Specifically, two tasks are adopted in BERT pre-training, *i.e.,* masked language modeling (MLM) and next sentence prediction (NSP). In MLM, BERT attempts to predict the masked words based on the cues from unmasked context words. In NSP, it defines a binary classification problem, which tries to predict whether two input sentences are consecutive.

In CV counterparts, it is common to pre-train the backbone on ImageNet [13], Kinetics [8] or large web sources [16] for the downstream tasks. There also exist works attempting to leverage the idea of BERT to CV tasks [48, 47, 35, 60, 9]. In sign language, Albanie *et al.* [1] propose to pre-train on a large annotated dataset and directly fine-tune on a small-scale one. Li *et al.* [33] fertilize recognition models by transferring knowledge of subtitled news sign videos to them. To our best knowledge, there exists no work focusing on the self-supervised pre-training for SLR.

2.3. Hand-Modeling Technique

There have been many works to model the hand using various techniques, including sum-of-Gaussians [46], shape primitives [38, 41] and sphere-meshes [50]. In order to model the hand shape more precisely, some works [2, 52] propose to utilize a triangulated mesh with Linear Blend Skinning (LBS) [31]. Recently, MANO [44] has become the most popular model with successful applications [18, 3, 19, 20]. As a statistical model, MANO is learned from a large volume of high-quality hand scans. Considering its capability of representing hand geometric changes in the low-dimensional shape and pose space, we adopt it as a constraint in the pose decoder to import hand prior.
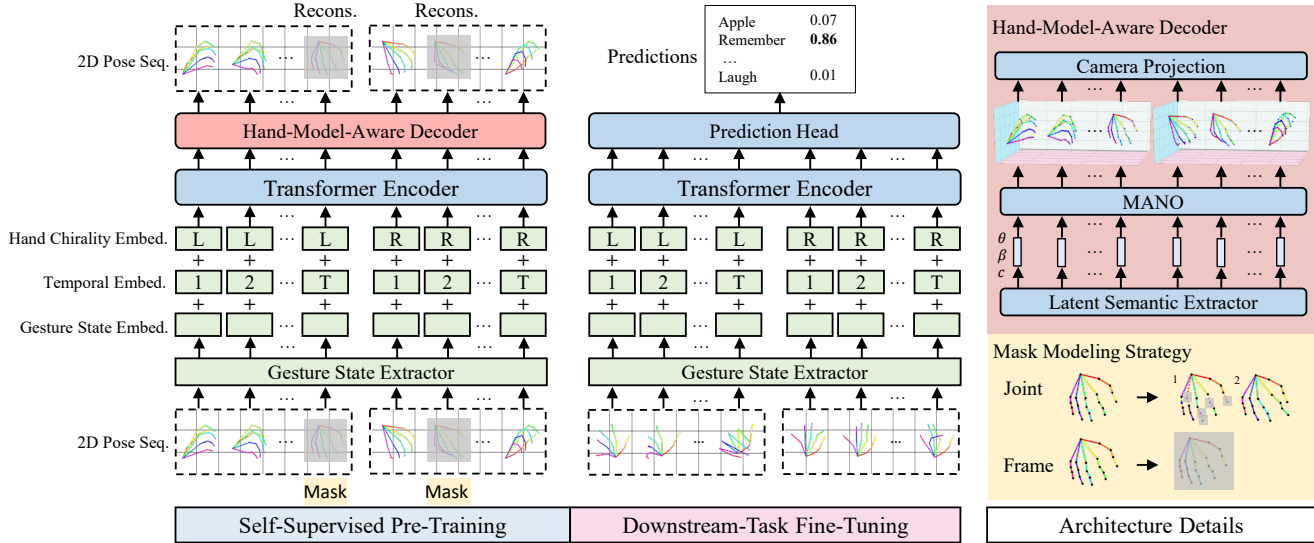
Figure 2. Illustration of our SignBERT framework, which contains self-supervised pre-training and fine-tuning for the downstream sign language recognition. The pre-extracted 2D hand pose sequence of both hands is fed into the framework. Each hand pose is viewed as a visual token, embedded with gesture state, temporal and hand chirality information. In self-supervised pre-training, we design several mask modeling strategies and incorporate model-aware hand prior to better exploit hierarchical contextual representation. For the downstream SLR task, the pre-trained Transformer encoder is fine-tuned with the prediction head to perform recognition.

3. Our Approach

Overview. As shown in Figure 2, SignBERT contains two stages, *i.e.,* pre-training for modeling context in sign videos and fine-tuning for the downstream SLR task. The hand poses, as visual tokens, are embedded with their gesture state, temporal and hand chirality information. Since sign language is performed by two hands, we jointly feed them into our framework. During pre-training, the whole framework works in a self-supervised paradigm by masking and reconstructing visual tokens. Jointly with the mask modeling strategies, the decoder incorporates hand prior for better capturing hierarchical context of both hands and temporal dependencies during the sign. When applying SignBERT to downstream recognition task, the hand-model-aware decoder is replaced by the prediction head, which is learned in a supervised paradigm by the corresponding video label.

In the following, we will first elaborate each component of our framework. Then we will describe the proposed pre-training and fine-tuning procedures, respectively.

3.1. Framework Architecture

The hand pose in each frame is viewed as a visual token. For each visual token, its input representation is constructed by summing the corresponding gesture state, temporal and hand chirality embeddings.

Gesture state embedding f_p**.** Since the hand pose is well-structured with the physical connection among joints, we organize it as a spatial graph. In this work, we adopt the spectral-based GCN from [4, 55] with a few modifications.

Given a 2D hand pose $\widetilde{J}_t$ representing the 2D location (x and y coordinates) at frame t, an undirected spatial graph is defined by the node V and edge E set, respectively. The node set includes all the corresponding hand joints, while the edge set contains the physical and symmetrical connections. The hand pose sequence is first fed into several graph convolutional layers frame-by-frame. Then graph pooling is performed based on neighbors to generate the frame-level semantics representation $f_{p,t}$.

Temporal embedding f_o**.** Temporal information matters in video-level SLR. Since self-attention does not consider the order information, we add the temporal order information by utilizing the position encoding strategy in [53]. Specifically, for the same hand, we add different temporal embeddings for different moments. Meanwhile, since two hands simultaneously convey the meaning during sign, we add the same temporal embedding for the same moment, regardless of hand chirality.

Hand chirality embedding f_h**.** Considering the meaning of sign language is conveyed by both hands, we introduce two special tokens to represent hand chirality of each frame, *i.e.,* 'L' and 'R' for the left and right hand, respectively. Specially, it is implemented by the WordPiece embeddings [54] with the same dimension as the gesture state and temporal embedding. Notably, all the frames belonging to the same hand contain the identical hand chirality embedding.

Transformer encoder. Given the aforementioned embedding representing the gesture status, temporal index and hand chirality, we sum them and feed it into the Transformer encoder following the original architecture [53], which con-

tains a multi-head attention module and a feed forward network. The encoder output $\mathbf{F}_N$, which retains the same size with the input, is computed as follows,

$$\begin{aligned} \mathbf{F}_0 &= \{f_p + f_o + f_h\}, \\ \widetilde{\mathbf{F}}_i &= L(M(\mathbf{F}_{i-1}) + \mathbf{F}_{i-1}), \\ \mathbf{F}_i &= L(C(\widetilde{\mathbf{F}}_i) + \widetilde{\mathbf{F}}_i), \end{aligned} \tag{1}$$

where i denotes the i-th layer of the Transformer encoder, and we utilize totally N layers. $L(\cdot)$, $M(\cdot)$ and $C(\cdot)$ denote the layer normalization, multi-head self-attention and feed forward network, respectively. $\mathbf{F}_i$ denotes the feature representation in i-th layer.

Hand-model-aware decoder. In our self-supervised pre-training paradigm, the framework needs to reconstruct the masked input sequence, in which the hand-model-aware decoder converts the feature to the pose sequence. Specifically, a fully-connected layer $D(\cdot)$ first extracts a latent semantic embedding describing the hand status and camera parameters from the representation generated by the Transformer encoder, which is formulated as follows,

$$\mathbf{F}_{la} = \{\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{c}_r, \mathbf{c}_o, c_s\}_{t=1}^{T} = D(\mathbf{F}_N), \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{25}$ and $\boldsymbol{\beta} \in \mathbb{R}^{10}$ are the pose and shape embedding for the following MANO, while $\mathbf{c}_r \in \mathbb{R}^{3\times3}$, $\mathbf{c}_o \in \mathbb{R}^2$, and $c_s \in \mathbb{R}$ are the weak-perspective camera parameters, indicating the rotation, translation and scale, respectively.

Then MANO [44] imports hand prior in a model-aware method and decodes the latent semantic embedding to hand representation. MANO is a fully-differentiable model providing a mapping from low-dimensional pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$ space to the triangulated hand mesh $\mathbf{M} \in \mathbb{R}^{N_v \times 3}$ with $N_v = 778$ vertices and $N_f = 1538$ faces. To produce a physically plausible mesh, the pose and shape are constrained in a PCA space learned from a large volume of hand scan data. The decoding process is formulated as follows,

$$\mathbf{M}(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(\mathbf{T}(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}), \tag{3}$$

$$\mathbf{T}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \bar{\mathbf{T}} + B_S(\boldsymbol{\beta}) + B_P(\boldsymbol{\theta}), \tag{4}$$

where $\mathbf{W}$ is a set of blend weights. $B_S(\cdot)$ and $B_P(\cdot)$ denote shape and pose blend functions, respectively. The hand template $\bar{\mathbf{T}}$ is first posed and skinned based on the pose and shape corrective blend shapes, *i.e.,* $B_P(\boldsymbol{\theta})$ and $B_S(\boldsymbol{\beta})$, Then the mesh is generated by rotating each part around joints $J(\boldsymbol{\beta})$ using the linear skinning function $W(\cdot)$ [25]. Besides, we are able to extract sparse 3D joints $\widetilde{J}_{3D}$ from the mesh. To keep consistent with the widely-used hand annotation format, we further add 5 extra vertices with the index of 333, 443, 555, 678 and 734 as the fingertips, leading to total 21 3D joints. Based on the predicted camera parameter, the predicted 3D joints are projected to the 2D

plane. The projected 2D hand pose is derived as follows,

$$\widetilde{J}_{2D} = c_s \prod (\mathbf{c}_r \widetilde{J}_{3D}) + \mathbf{c}_o, \tag{5}$$

where $\prod(\cdot)$ denotes the orthographic projection.
Prediction head. Since discriminative cues may only contain in certain frames, we utilize a simple attention mechanism to weight features temporally. Then the weighted features are summed to perform final classification.

3.2. Pre-Training SignBERT

In this section, we elaborate SignBERT pre-training paradigm on a large volume of sign data sources to exploit semantic context hierarchically. Different from the original BERT pre-training on discrete word space, we aim to pre-train on continuous hand pose space. Substantially, the classification problem is transformed into regression, which poses new challenges on the reconstruction of the hand pose sequence. To tackle this issue, we view hand poses as visual 'words' (continuous tokens) and jointly utilize the aforementioned model-aware decoder as a constraint with hand prior incorporated. Given a hand sequence containing both hands, we first randomly choose 50% tokens. Similar to BERT, if the token is chosen, we randomly perform one of three operations with equal probability, *i.e.,* masked joint modeling, masked frame modeling and identity modeling.
Masked joint modeling. Since current pose detectors may contain failure detection on some joints, we incorporate masked joint modeling to mimic the usual failure cases. In a chosen token, we randomly choose m joints ranging from 1 to M. For these chosen joints, we perform two operations with equal probability, *i.e.,* zero masking (masking the coordinates of joints with zeros) or random spatial disturbance. This modeling attempts to embed our framework the capability to infer the gesture state from remaining hand joints, thus capturing context at the joint level.
Masked frame modeling. Masked frame modeling is performed on a more holistic view. For a chosen token, all the joints are zero masked. The framework is enforced to reconstruct this token by observations from remaining pose tokens of the other hand or different temporal points. In this way, temporal context in each hand and mutual context between hands are captured.
Identity modeling. Identity modeling makes the unchanged token fed into the framework. This operation is indispensable for the framework to learn identity mapping on those unmasked tokens.

3.3. Objective Functions in Pre-Training

The proposed three strategies allow the network to maximize the likelihood of the joint probability distribution to reconstruct the hand pose sequence. In this manner, the context contained in the sequence is captured. During pre-training, only the output corresponding to chosen tokens are

included in the following loss calculation as follows,

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda\mathcal{L}_{reg}, \tag{6}$$

where λ denotes the weighting factor.

Hand reconstruction loss $\mathcal{L}_{rec}$. Since hand pose detection results J_{2D} serve as the pseudo label, we ignore the joints with the prediction confidence lower than ϵ and utilize the remaining joints weighted by the confidence in the calculation of this loss term.

$$\mathcal{L}_{rec} = \sum_{t,j} \mathbb{1}(c(t,j) >= \epsilon)c(t,j)\left\|\tilde{J}_{2D}(t,j) - J_{2D}(t,j)\right\|_1, \tag{7}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function, and $c(t,j)$ denotes the confidence of the J_{2D} with joint j at time t.

Regularization loss $\mathcal{L}_{reg}$. To ensure the hand model working properly, a regularization loss is added. It is implemented by constraining magnitude and derivative of the MANO input, which is responsible for generating the plausible mesh and keeping the signer identity unchanged. The regularization loss is calculated as follows,

$$\mathcal{L}_{reg} = \sum_{t}(\|\theta_t\|_2^2 + w_\beta\|\beta_t\|_2^2 + w_\delta\|\beta_t - \beta_{t-1}\|_2^2), \tag{8}$$

where w_β and w_δ denote the weighting factor.

3.4. Fine-Tuning SignBERT

After pre-training SignBERT, it is relatively simple to fine-tune it for the downstream SLR task. The hand-model-aware decoder is replaced by the prediction head. The input hand pose sequence is all unmasked and we use the cross-entropy loss to supervise the output of the prediction head.

Considering only the hand pose sequence is insufficient to convey the full meaning of sign language, it is necessary to fuse recognition results based on hands with that of full frame. The full frame can be represented by full RGB data or full keypoints. In our work, we use the simple late fusion strategy, which directly sums their prediction results. Besides, the full RGB and keypoints baseline method utilized for fusion are marked in each dataset for clarity. In the following, we refer our method with only hands, fusion of hands and full RGB data, fusion of hands and full keypoints as **Ours (H)**, **Ours (H + R)** and **Ours (H + P)**, respectively.

4. Experiments

4.1. Datasets and Evaluation

Datasets. We evaluate our proposed method on four public sign language datasets, including NMFs-CSL [21], SLR500 [22], MSASL [24] and WLASL [34].

NMFs-CSL is the most challenging Chinese sign language (CSL) dataset due to a large variety of confusing words caused by fine-grained cues. It totally contains 1,067

words with 610 confusing words and 457 normal words. There are 25,608 and 6,402 samples for training and testing, respectively. **SLR500** is another CSL dataset, which contains 500 daily words with 125,000 recording samples performed by 50 signers. Specifically, 90,000 and 35,000 samples are utilized for training and testing, respectively.

MSASL is an American sign language dataset (ASL) containing a vocabulary size of 1,000, with 25,513 samples in total for training, validation and testing, respectively. Besides, the Top-100 and Top-200 most frequent words are chosen as its two subsets, referred to as MSASL100, MSASL200. **WLASL** is another ASL dataset with a vocabulary of 2,000 words and 21,083 samples. Similar to MSASL, it releases WLASL100 and WLASL300 as its subsets. MSASL and WLASL are both collected from Web videos and bring new challenges due to unconstrained real-life recording conditions and limited samples for each word.

Meanwhile, since STB [58] and HANDS17 [57] provide 2D hand joint annotations, we utilize them to validate the feasibility of our proposed framework.

STB is a real-world hand pose estimation datasets, which contains 18,000 samples. Following Zimmermann *et al.* [61], we split this dataset into 15,000 training and 3,000 testing samples for single-frame validation. **HANDS17** is a video-level hand pose estimation dataset, containing a total of 292,820 frames from 99 video sequences. In this dataset, we split the first 70% and last 30% frames in each sequence for training and testing, respectively.

Evaluation. For the downstream isolated SLR task, we utilize the accuracy metrics, *i.e.,* the per-class (**P-C**) and per-instance (**P-I**) metrics, which denote the average accuracy over each class and each instance, respectively. We report the Top-1 and Top-5 accuracy under both per-instance and per-class for MSASL and WLASL. Since NMFs-CSL and SLR500 contain the same number of samples for each class, we only report per-instance accuracy following [21, 22].

For STB and HANDS17, we report the Percentage of Correct Keypoints (PCK) score and the area under the curve (AUC) on the PCK ranging from 20 to 40 pixels, which are widely-used criteria to evaluate pose estimation accuracy. Specifically, PCK defines a candidate keypoint to be correct if it falls within a circle (2D) of a given radius around the ground truth, where the distances are expressed in pixels.

4.2. Implementation Details

In our experiment, all the models are implemented by PyTorch [39] and trained on NVIDIA RTX 3090. Since no pose annotation is available in sign language datasets, we use MMPose [11] for its efficiency to extract the 133 full 2D keypoints, *i.e.,* the 23 body joints, 68 face and 42 hand joints. The extracted hand and shoulder joints are further utilized to crop the left and right hand pose and rescale them to 256×256. Both hands are fed into the framework.

M	Input		Output	
	P@20	AUC	P@20	AUC
3	88.81	91.02	99.90	99.54
5	82.26	85.65	99.89	99.53
7	76.19	80.91	99.85	99.53
9	70.85	76.63	99.81	99.50
11	66.29	72.85	99.79	99.44

Table 1. Frame-level framework feasibility on the STB dataset. 'P@20' denotes the PCK metrics with the error threshold set as 20 pixel. We only utilize the masked joint modeling, and M denotes the number of masked joints ranges from 1 to M.

Mask		Input		Output	
Joint	Frame	P@20	AUC	P@20	AUC
✓		86.38	89.02	95.13	95.49
	✓	80.85	80.85	95.33	95.57
✓	✓	81.43	82.32	95.14	95.48

Table 2. Video-level framework feasibility on HANDS17. 'P@20' denotes the PCK metrics with the error threshold set as 20 pixel. 'Joint' and 'Frame' denote the masked joint modeling and masked frame modeling, respectively.

The framework is trained with the Adam optimizer. The weight decay and momentum are set to 0.0001 and 0.9, respectively. We start at the initial learning rate of 0.001 and reduce it by a factor of 0.1 every 20 epochs. In all experiments, the hyper parameters ϵ, λ, w_β and w_δ are set as 0.5, 0.01, 10.0 and 100.0, respectively. During the pre-training stage, we include the training data from all four aforementioned sign language datasets. For the downstream task, we temporally extract 32 frames using random and center sampling during training and testing, respectively.

4.3. Ablation Study

In this section, we first validate the feasibility of our framework. Then we perform ablation studies to demonstrate the effectiveness of the main components in our framework.

Framework feasibility. We validate the feasibility of our framework on the datasets with hand pose annotation available. As shown in Table 1, we first validate reconstruction ability under the single-frame setting on the STB dataset. Specifically, a single frame is fed into the framework. We only perform the masked joint modeling, where M indicates the number of masked joints ranges from 1 to M, resulting the average number as $M/2$. With the gradual increase of M, the PCK and AUC metrics of reconstructed joints are consistently higher than those of the input. It demonstrates that our framework is able to hallucinate the whole hand pose by observing partial joints.

From Table 2, the framework feasibility under the video-level setting is tested on the HANDS17 dataset. We utilize all masking strategies on the original pose sequence to
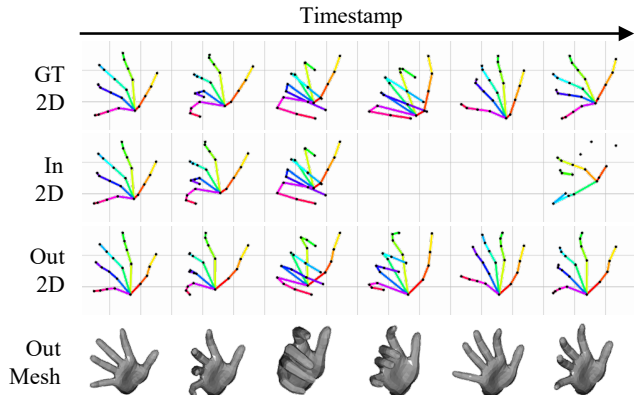

Figure 3. Visualization of the framework feasibility on HANDS17. We choose 6 continuous frames from one video. The four rows denote the ground truth (GT) pose sequence, input sequence after performing masking on GT, the reconstructed sequence and middle results of the mesh sequence, respectively. Notably, two blanks in the second row represent these poses are all masked.

formulate the input. It can be also observed that the PCK and AUC performance of the output sequence are higher than those of input, which verifies the framework capability of reconstructing from inaccurate hand joint sequence. Besides, we visualize the hand pose reconstruction in Figure 3.

Since we focus on the performance of the downstream recognition task, we perform extensive experiments on MSASL and its subsets to demonstrate the effectiveness of the masking strategies, model-aware decoder, Transformer layers N and pre-training data scale. We report per-instance and per-class Top-1 accuracy as the performance indicator.

Effectiveness of the masking strategy. As illustrated in Table 4, the first row denotes the baseline method, *i.e.,* our framework is directly trained under the video label supervision without pre-training. It is worth mentioning that compared with this baseline, our designed pre-training brings notable performance gain, with 13.08%, 12.95% and 7.69% Top-1 per-instance accuracy improvement. Both joint-level and frame-level masking strategies are beneficial for the framework capturing different levels of context, thus bringing performance improvement. When two masking strategies are both utilized, it reaches the best performance.

Effectiveness of the model-aware decoder. As shown in Table 5, we compare the effect of different pose decoders on SLR. The first two rows denote utilizing the fully-connected layers to regress the hand pose. Our decoder work in a model-aware method to import hand prior during pre-training, which eases optimization and brings performance improvement for downstream isolated SLR. Besides, the model-aware decoder has additional benefits, which inflates the 2D hand pose sequence to the 3D plane.

Effectiveness of Transformer layers N. From Table 6, the accuracy increases, when the number of Transformer layers increases. It reaches the peak when $N = 3$. The

Method	Total			Confusing			Normal		
	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5	Top-1	Top-2	Top-5
Pose-based									
ST-GCN [55]	59.9	74.7	86.8	42.2	62.3	79.4	83.4	91.3	96.7
Ours (H)	67.0	86.8	95.3	46.4	78.2	92.1	94.5	98.1	99.6
Ours (H + P)	**74.9**	**93.2**	**98.2**	**58.6**	**88.6**	**96.9**	**96.7**	**99.3**	**99.9**
RGB-based									
3D-R50 [42]	62.1	73.2	82.9	43.1	57.9	72.4	87.4	93.4	97.0
DNF [12]	55.8	69.5	82.4	33.1	51.9	71.4	86.3	93.1	97.0
I3D [8]	64.4	77.9	88.0	47.3	65.7	81.8	87.1	94.3	97.3
TSM [36]	64.5	79.5	88.7	42.9	66.0	81.0	93.3	97.5	99.0
Slowfast [17]	66.3	77.8	86.6	47.0	63.7	77.4	92.0	96.7	98.9
GLE-Net [21]	69.0	79.9	88.1	50.6	66.7	79.6	93.6	97.6	99.3
Ours (H + R)	**78.4**	**92.0**	**97.3**	**64.3**	**86.5**	**95.4**	**97.4**	**99.3**	**99.9**

Table 3. Accuracy comparison on NMFs-CSL dataset. [55] and [42] denote the pose and RGB baseline, respectively.

Mask		100		200		1000	
Joint	Frame	P-I	P-C	P-I	P-C	P-I	P-C
		63.01	62.72	57.69	57.56	41.85	38.30
✓		72.66	72.75	68.51	69.72	48.87	45.39
	✓	74.77	75.48	68.65	69.20	49.02	46.02
✓	✓	**76.09**	**76.65**	**70.64**	**70.92**	**49.54**	**46.39**

Table 4. Effectiveness of the masking strategy on MSASL dataset. The first row denotes the baseline, *i.e.,* our framework is trained without pre-training. 'Joint' and 'Frame' denote the masked joint modeling and masked frame modeling, respectively.

Decoder	100		200		1000	
	P-I	P-C	P-I	P-C	P-I	P-C
1-layer fc	73.05	72.62	67.55	68.21	47.94	45.07
2-layer fc	74.24	74.21	68.29	69.12	48.03	45.25
Ours	**76.09**	**76.65**	**70.64**	**70.92**	**49.54**	**46.39**

Table 5. Effectiveness of the model-aware decoder on MSASL dataset. We compare ours with different pose decoders.

N	100		200		1000	
	P-I	P-C	P-I	P-C	P-I	P-C
2	74.11	74.61	67.70	67.92	48.23	45.17
3	**76.09**	**76.65**	**70.64**	**70.92**	**49.54**	**46.39**
4	75.69	75.51	70.20	70.66	47.36	44.04
5	74.90	75.68	68.14	68.40	47.29	44.42

Table 6. Effectiveness of the Transformer layers N on MSASL dataset. N denotes the number of the layers in the Transformer encoder.

Ratio	100		200		1000	
	P-I	P-C	P-I	P-C	P-I	P-C
0%	63.01	62.72	57.69	57.56	41.85	38.30
25%	73.18	72.83	67.91	69.30	46.18	43.97
50%	73.18	73.42	67.18	67.71	46.57	43.79
75%	74.50	74.36	68.72	68.97	47.21	43.67
100%	**76.09**	**76.65**	**70.64**	**70.92**	**49.54**	**46.39**

Table 7. Effectiveness of the ratio of pre-training data scale on the MSASL dataset.

4.4. Comparison with State-of-the-art Methods

We compare our method with previous state-of-the-art methods on four benchmark datasets. For clarity, previous methods are grouped by their input modality, *i.e.,* pose-based and RGB-based methods.

Evaluation on NMFs-CSL. As illustrated in Table 3, we compare with methods [55, 42, 12, 8, 36, 17, 21] utilizing the pose and RGB sequence as input. GLE-Net [21] is the most challenging method, which enhances discriminative cues from global and local views. It is worth noting that our method with purely using hand pose achieves comparable performance with a majority of them. Ours (H + R) outperforms all previous methods with a notable margin.

Evaluation on SLR500. As shown in Table 10, STIP [30] and GMM-HMM [49] are traditional methods based on hand-crafted features. GLE-Net [21] still achieves the best performance. Notably, our method achieves the best performance, reaching 97.6% top-1 accuracy.

Evaluation on MSASL. MSASL brings new challenges due to unconstrained recording settings. As shown in Table 8, compared with the RGB baseline [24], ST-GCN [55] shows inferior performance. It may be caused by the failure of pose detection on sign videos, which contains the partially occluded upper body, motion blur and noisy backgrounds. Albanie *et al.* [1] and Li *et al.* [33] both use more external RGB sign data to boost the performance on MSASL or its subsets. It is worth noting that our method

difference of the best layers in BERT and our model may be due to different characteristics between sign pose and NLP domain, and the overfitting issue. Unless stated, we utilize $N = 3$ in all our experiments.

Effectiveness of the pre-training data scale. As shown in Table 7, as the ratio of pre-training data volume increases, the performance on the downstream SLR task gradually increases on the accuracy metrics. It indicated that SignBERT may benefit from larger pre-training datasets.

Method	MSASL100				MSASL200				MSASL1000			
	Per-instance		Per-class		Per-instance		Per-class		Per-instance		Per-class	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Pose-based												
ST-GCN [55]	59.84	82.03	60.79	82.96	52.91	76.67	54.20	77.62	36.03	59.92	32.32	57.15
Ours (H)	76.09	92.87	76.65	93.06	70.64	89.55	70.92	90.00	49.54	74.11	46.39	72.65
Ours (H + P)	**81.37**	**93.66**	**82.31**	**93.76**	**77.34**	**91.10**	**78.02**	**91.48**	**59.80**	**81.86**	**57.06**	**80.94**
RGB-based												
I3D [24]	-	-	81.76	95.16	-	-	81.97	93.79	-	-	57.69	81.05
TCK [33]	83.04	93.46	83.91	93.52	80.31	91.82	81.14	92.24	-	-	-	-
BSL [1]	-	-	-	-	-	-	-	-	64.71	85.59	61.55	84.43
Ours (H + R)	**89.56**	**97.36**	**89.96**	**97.51**	**86.98**	**96.39**	**87.62**	**96.43**	**71.24**	**89.12**	**67.96**	**88.40**

Table 8. Accuracy comparison on MSASL dataset. [55] and [24] denote the pose and RGB baseline, respectively.

Method	WLASL100				WLASL300				WLASL2000			
	Per-instance		Per-class		Per-instance		Per-class		Per-instance		Per-class	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Pose-based												
ST-GCN [55]	50.78	79.07	51.62	79.47	44.46	73.05	45.29	73.16	34.40	66.57	32.53	65.45
Pose-TGCN [34]	55.43	78.68	-	-	38.32	67.51	-	-	23.65	51.75	-	-
PSLR [51]	60.15	83.98	-	-	42.18	71.71	-	-	-	-	-	-
Ours (H)	76.36	91.09	77.68	91.67	62.72	85.18	63.43	85.71	39.40	73.35	36.74	72.38
Ours (H + P)	**79.07**	**93.80**	**80.05**	**94.17**	**70.36**	**88.92**	**71.17**	**89.36**	**47.46**	**83.32**	**45.17**	**82.32**
RGB-based												
I3D [34]	65.89	84.11	67.01	84.58	56.14	79.94	56.24	78.38	32.48	57.31	-	-
TCK [33]	77.52	91.08	77.55	91.42	68.56	89.52	68.75	89.41	-	-	-	-
BSL [1]	-	-	-	-	-	-	-	-	46.82	79.36	44.72	78.47
Ours (H + R)	**82.56**	**94.96**	**83.30**	**95.00**	**74.40**	**91.32**	**75.27**	**91.72**	**54.69**	**87.49**	**52.08**	**86.93**

Table 9. Accuracy comparison on WLASL dataset. ST-GCN [55] and I3D [34] denote the pose and RGB baseline, respectively.

Method	Accuracy
Pose-based	
ST-GCN [55]	90.0
Ours (H)	94.5
Ours (H + P)	**96.6**
RGB-based	
STIP [30]	61.8
GMM-HMM [49]	56.3
3D-R50 [42]	95.1
GLE-Net [21]	96.8
Ours (H + R)	**97.6**

Table 10. Accuracy comparison on SLR500 dataset. [55] and [42] denote the pose and RGB baseline, respectively.

achieves noticeable performance improvement when compared with both pose-based and RGB-based methods.

Evaluation on WLASL. Compared with MSASL, WLASL contains fewer samples and double vocabulary size. It can be observed that Ours (H + P), which only utilizes pose as the input modality, even outperforms the most challenging RGB-based method [1]. Besides, Ours (H + R) further outperforms the best competitor by 7.87% per-instance top-1 accuracy improvement on WLASL2000. With incorporated hand prior and self-supervised pre-training, our method is more effective under the benchmark with limited samples.

5. Conclusion

In this paper, we introduce the *first* self-supervised pre-trainable SLR framework with model-aware hand prior incorporated, namely SignBERT. We involve both hands and view hand pose as a visual token. The visual token is embedded with gesture state, temporal and hand chirality information before feeding into the framework. We first perform self-supervised pre-training on a large volume of hand poses by masking and reconstructing the hand tokens. During pre-training, our framework consists of the Transformer encoder and hand-model-aware decoder. Jointly with incorporated hand prior by the decoder, we elaborately design several masking strategies for better capturing hierarchical contextual information. Then our pre-trained framework is fine-tuned to perform recognition. We perform extensive experiments on four popular benchmark datasets. Experiment results demonstrate the effectiveness of our method, achieving new state-of-the-art performance on all benchmarks with a notable margin.

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 *ECCV*, pages 35–53, 2020.

[2] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, pages 640–653, 2012.

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3D hand shape and pose from images in the wild. In *CVPR*, pages 10843–10852, 2019.

[4] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In *ICCV*, pages 2272–2281, 2019.

[5] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. SubUNets: End-to-end hand shape and continuous sign language recognition. In *ICCV*, pages 3075–3084, 2017.

[6] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language Transformers: Joint end-to-end sign language recognition and translation. In *CVPR*, pages 10023–10033, 2020.

[7] Congqi Cao, Cuiling Lan, Yifan Zhang, Wenjun Zeng, Hanqing Lu, and Yanning Zhang. Skeleton-based action recognition with gated convolutional neural networks. *TCSVT*, 29(11):3247–3257, 2018.

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, pages 1691–1703, 2020.

[10] Ka Leong Cheng, Zhaoyang Yang, Qifeng Chen, and Yu-Wing Tai. Fully convolutional networks for continuous sign language recognition. In *ECCV*, pages 697–714, 2020.

[11] MMPose Contributors. OpenMMLab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[12] Runpeng Cui, Hu Liu, and Changshui Zhang. A deep neural framework for continuous sign language recognition by iterative training. *TMM*, 21(7):1880–1891, 2019.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2018.

[15] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015.

[16] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *ECCV*, pages 670–688, 2020.

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019.

[18] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. DeepCap: Monocular human performance capture using weak supervision. In *CVPR*, pages 5052–5063, 2020.

[19] Hezhen Hu, Weilun Wang, Wengang Zhou, Weichao Zhao, and Houqiang Li. Model-aware gesture-to-gesture translation. In *CVPR*, pages 16428–16437, 2021.

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

[21] Hezhen Hu, Wengang Zhou, Junfu Pu, and Houqiang Li. Global-local enhancement network for NMFs-aware sign language recognition. *TOMM*, 17(3):1–18, 2021.

[22] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. Attention based 3D-CNNs for large-vocabulary sign language recognition. *TCSVT*, 29(9):2822–2832, 2019.

[23] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI*, pages 2257–2264, 2018.

[24] Hamid Reza Vaezi Joze and Oscar Koller. MS-ASL: A large-scale data set and benchmark for understanding american sign language. *BMVC*, pages 1–16, 2019.

[25] Ladislav Kavan and Jiří Žára. Spherical blend skinning: a real-time deformation of articulated models. In *ACM I3D*, pages 9–16, 2005.

[26] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *NeurIPS*, pages 3294–3302, 2015.

[27] Oscar Koller. Quantitative survey of the state of the art in sign language recognition. *arXiv*, pages 1–40, 2020.

[28] Oscar Koller, Cihan Camgoz, Hermann Ney, and Richard Bowden. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *TPAMI*, 42(9):2306–2320, 2020.

[29] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid cnn-hmms. *IJCV*, 126(12):1311–1325, 2018.

[30] Ivan Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005.

[31] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH*, pages 165–172, 2000.

[32] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. pages 786–792, 2018.

[33] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Transferring cross-domain knowledge for video sign language recognition. In *CVPR*, pages 6205–6214, 2020.

[34] 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 *WACV*, pages 1459–1469, 2020.

[35] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pages 11336–11344, 2020.

[36] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019.

[37] Yuecong Min, Yanxiao Zhang, Xiujuan Chai, and Xilin Chen. An efficient pointlstm for point clouds based gesture recognition. In *CVPR*, pages 5761–5770, 2020.

[38] Iason Oikonomidis, Manolis IA Lourakis, and Antonis A Argyros. Evolutionary quasi-random search for hand articulations tracking. In *CVPR*, pages 3422–3429, 2014.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 1–12, 2019.

[40] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[41] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *CVPR*, pages 1106–1113, 2014.

[42] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatiotemporal representation with pseudo-3D residual networks. In *ICCV*, pages 5533–5541, 2017.

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *arxiv*, pages 1–12, 2018.

[44] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ToG*, 36(6):1–17, 2017.

[45] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, pages 4263–4270, 2017.

[46] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*, pages 2456–2463, 2013.

[47] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*, pages 1–16, 2020.

[48] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *ICCV*, pages 7464–7473, 2019.

[49] Ao Tang, Ke Lu, Yufei Wang, Jie Huang, and Houqiang Li. A real-time hand posture recognition system using deep neural networks. *ACM TIST*, 6(2):1–23, 2015.

[50] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ToG*, 35(6):1–11, 2016.

[51] Anirudh Tunga, Sai Vidyaranya Nuthalapati, and Juan Wachs. Pose-based sign language recognition using GCN and BERT. In *WACV Workshop*, pages 31–40, 2020.

[52] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 118(2):172–193, 2016.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5999–6009, 2017.

[54] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*, pages 1–23, 2016.

[55] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018.

[56] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 1–18, 2019.

[57] Shanxin Yuan, Qi Ye, Guillermo Garcia-Hernando, and Tae-Kyun Kim. The 2017 hands in the million challenge on 3D hand pose estimation. *arXiv*, pages 1–7, 2017.

[58] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *ICIP*, pages 982–986, 2017.

[59] Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. Improving sign language translation with monolingual data by sign back-translation. In *CVPR*, pages 1316–1325, 2021.

[60] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *CVPR*, pages 8746–8755, 2020.

[61] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, pages 4903–4911, 2017.