BEST: BERT Pre-training for Sign Language Recognition with Coupling Tokenization

Weichao Zhao[1], **Hezhen Hu**[1*], **Wengang Zhou**[1, 2†], **Jiaxin Shi**[3], **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 [3]Huawei Inc.
{saruka, alexhu}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn, shijiaxin3@huawei.com

Abstract

In this work, we are dedicated to leveraging the BERT pre-training success and modeling the domain-specific statistics to fertilize the sign language recognition (SLR) model. Considering the dominance of hand and body in sign language expression, we organize them as pose triplet units and feed them into the Transformer backbone in a frame-wise manner. Pre-training is performed via reconstructing the masked triplet unit from the corrupted input sequence, which learns the hierarchical correlation context cues among internal and external triplet units. Notably, different from the highly semantic word token in BERT, the pose unit is a low-level signal originally located in continuous space, which prevents the direct adoption of the BERT cross-entropy objective. To this end, we bridge this semantic gap via coupling tokenization of the triplet unit. It adaptively extracts the discrete pseudo label from the pose triplet unit, which represents the semantic gesture/body state. After pre-training, we fine-tune the pre-trained encoder on the downstream SLR task, jointly with the newly added task-specific layer. Extensive experiments are conducted to validate the effectiveness of our proposed method, achieving new state-of-the-art performance on all four benchmarks with a notable gain.

1 Introduction

Sign language is a primary communication tool for the deaf community. It is characterized by its unique grammar and lexicon, which are difficult to understand for non-sign language users. To bridge this communication gap, automatic sign language recognition (SLR) is widely studied with broad social influence. As one basic task, isolated SLR aims to recognize at the gloss-level and is a fine-grained classification problem. In this work, we focus on this task.

Due to high annotation cost, current labeled sign data sources are limited. Since common deep-learning-based methods are data-hungry, they are prone to over-fitting on SLR. To this end, several attempts have been made in SLR. For instance, considering the dominant role of hand, some methods (Hu, Zhou, and Li 2021; Albanie et al. 2020) utilize the cropped hand sequence as the auxiliary information. HMA (Hu, Zhou, and Li 2021) proposes to recognize sign language in a model-aware paradigm with the hand mesh as the intermediate constraint. However, those methods directly optimize on the target benchmark but fail to leverage the universal statistics in the sign language domain.

Notably, self-supervised pre-training techniques represented by BERT (Devlin et al. 2018) have achieved great success in Natural Language Processing (NLP). BERT builds on the strong Transformer (Vaswani et al. 2017) backbone and designs an ingenious pretext task, *i.e.,* masked language modeling (MLM). It aims to reconstruct the masked word tokens from the corrupted input sequence, whose objective is implemented by cross entropy to maximize the joint word probability distribution. However, the main obstacle to leverage its success in video SLR is the different characteristics of the input signal. In NLP, the input word token is discrete and pre-defined with high semantics. In contrast, the video signal of sign language is continuous with the spatial and temporal dimensions. This signal is quite low-level, making the original BERT objective not applicable. Besides, since the sign language video is mainly characterized by hand and body movements, the direct adoption of the BERT framework may not be optimal.

To tackle the above issue, we propose a self-supervised pre-trainable framework with a specific design for sign language, namely BEST. Focusing on the main properties of sign language, we organize the hand and body as the pose triplet unit. This triplet unit is embedded and fed into the Transformer backbone. Basically, our framework contains two stages, *i.e.,* self-supervised pre-training and downstream fine-tuning. During pre-training, we propose the masked unit modeling (MUM) pretext task to capture the context cues. The input hand or body unit embedding is randomly masked, and then the framework reconstructs the masked unit from this corrupted input sequence. Similar to BERT, self-reconstruction is optimized via the cross-entropy objective. To this end, we jointly tokenize the pose triplet unit as the pseudo label, which represents the gesture/body state. After pre-training, the pre-trained Transformer encoder is fine-tuned with the newly added prediction head to perform the SLR task.

Our contributions are summarized as follows,

- We propose a self-supervised pre-trainable framework. It leverages the BERT success, jointly with the specific design for the sign language domain.

[*]Contribute equally with the first author.

[†]Corresponding authors: Wengang Zhou and Houqiang Li.

- We organize the main hand and body movement as the pose triplet unit and propose the masked unit modeling (MUM) pretext task. To utilize the BERT objective, we generate the pseudo label for this task via coupling tokenization on the pose triplet unit.
- Extensive experiments on downstream SLR validate the effectiveness of our proposed method, achieving new state-of-the-art performance on four benchmarks with a notable gain.

2 Related Work

In this section, we will briefly review several related topics, including sign language recognition and self-supervised pre-training.

2.1 Sign Language Recognition

Sign language recognition has received much attention in recent years (Koller et al. 2018; Momeni et al. 2022; Niu and Mak 2020; Jin and Zhao 2021; Hu, Zhou, and Li 2021; Li et al. 2020b). Typically, the research works can be grouped into two categories based on the input modality, *i.e.,* RGB-based methods and pose-based methods.

RGB-based Methods. Early works (Farhadi, Forsyth, and White 2007; Fillbrandt, Akyol, and Kraiss 2003; Starner 1995) on SLR focused on hand-crafted features computed for hand shape variation and body motion. Along with the popularity of convolutional neural networks (CNNs) in computer vision, many works in SLR adopt CNNs as the backbone (Selvaraj et al. 2022; Sincan and Keles 2020; Joze and Koller 2018; Koller et al. 2018; Hu, Zhou, and Li 2021). For example, 3D CNNs are adopted due to their representation capacity for spatio-temporal dependency (Huang et al. 2018; Joze and Koller 2018; Albanie et al. 2020; Li et al. 2020b).

Pose-based Methods. Pose modality is a compact and high-level representation of human action and contains physical connection among skeleton joints (Li et al. 2018; Ng et al. 2021; Yan, Xiong, and Lin 2018). To extract the semantic representation of pose data, some works (Li et al. 2018; Du, Wang, and Wang 2015) explore graph convolutional networks (GCNs) as the backbone. These GCN-based methods (Camgoz et al. 2018; Li et al. 2018; Yan, Xiong, and Lin 2018; Min et al. 2020) show impressive performance in action recognition. Tunga *et al.* (Tunga, Nuthalapati, and Wachs 2021) combine GCN and Transformer to capture spatial-temporal information based on sign language pose sequence for sign language recognition. In this work, given the compactness of pose data, we utilize them as our input modality.

2.2 Self-Supervised Pre-training

Self-supervised pre-training methods have achieved remarkable success in Natural Language Processing (NLP) and Computer Vision (CV) fields, which make full use of large-scale unlabeled data to learn generic feature representation for a wide range of downstream tasks. In NLP, with the strong modeling capability of Transformer (Vaswani et al. 2017), many works propose to pre-train on this backbone for generic representations (Devlin et al. 2018; Lewis et al. 2019; Conneau and Lample 2019). BERT is one of the most popular methods, which designs a cleverly masked language modeling (MLM) pretext task. MLM predicts the masked word tokens from the corrupted input sentence, which aims to capture the context cues in the text corpus.

Motivated by BERT, some works attempt to leverage its success into CV tasks (Su et al. 2019; Sun et al. 2019; Zhu and Yang 2020; Bao, Dong, and Wei 2021; He et al. 2022). There exist different characteristics of the input signal between NLP and CV. Different from the semantic discrete word token, the signal in CV tasks is usually low-level and continuous, which makes the original BERT objective not applicable. One way to tackle this problem is changing its objective into regression. He *et al.* (He et al. 2022) propose masked autoencoders (MAE) to reconstruct the missing pixels of masked image patches using regression objective. Hu *et al.* (Hu et al. 2021a) propose a pre-trained model for sign language based on self-reconstruction of the hand pose data. Jiang *et al.* (Jiang, Camgoz, and Bowden 2021) attempt to utilize BERT-style refined pose for sign language. According to (Ramesh et al. 2021), the former method may focus too many short-range dependencies and hurt the downstream performance. Therefore, some methods turn to tokenizing the input signal to provide the discrete pseudo label. BEiT (Bao, Dong, and Wei 2021) propose masked image modeling (MIM) with tokenized image patches as supervision. It is originally designed for image-based tasks, and cannot be directly adopted into the video-based sign language domain, due to different task characteristics and input modalities.

3 Methods

Our proposed method consists of two stages, *i.e.,* self-supervised pre-training and downstream task fine-tuning. As shown in Figure 1, during pre-training, we first utilize a discrete variational autoencoder to learn codebooks for the upper body, left and right hand. It performs coupling tokenization on the pose triplet unit as the pseudo label for the following pretext task. Then we pre-train the BEST model via our designed masked unit modeling (MUM) pretext task to capture the context cues. Finally, we append the task-specific layer and fine-tune the pre-trained parameters on the downstream SLR task.

3.1 Tokenization in Pre-training

The tokenization provides pseudo labels for our designed pretext task during pre-training. We utilize a discrete variational autoencoder (d-VAE) to jointly convert the pose triplet unit into the triplet tokens (body, left and right hand), motivated by VQ-VAE (Van Den Oord, Vinyals et al. 2017). Our utilized pose triplet unit J_{sign} consists of two hand poses J_{left}, J_{right}, and an upper body pose J_{body}. The d-VAE contains three parts, *i.e.,* encoder, quantizer, and decoder. The encoder maps the pose triplet unit to the intermediate latent vector $z = \mathrm{Enc}(J_{sign})$. The quantizer is in charge of tokenizing each vector to be codewords coming from the hand codebook $\mathcal{V}_{hand} = \{h_k\}_{k=1}^{M_1}$ and the upper body codebook $\mathcal{V}_{body} = \{d_k\}_{k=1}^{M_2}$. The quantized vector $z_{\mathbf{q}}$ is computed as

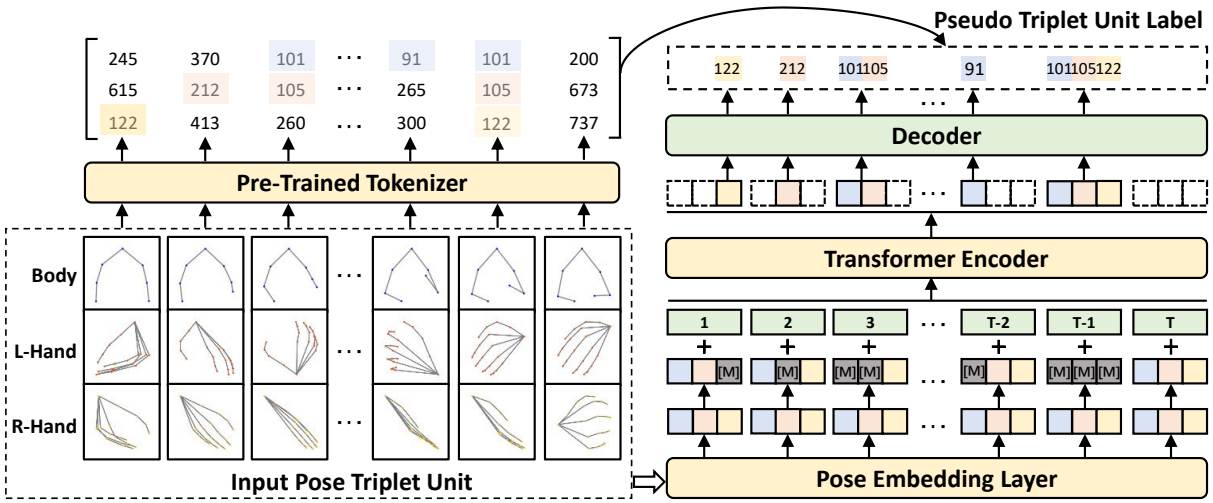

Figure 1: Illustration of our proposed BEST framework during pre-training. BEST mainly contains four components, *i.e.,* a pre-trained tokenizer, a pose embedding layer, a Transformer encoder, and a decoder. The input pose triplet unit is composed by the body pose, the left and right hand pose. Given a sequence of pose triplet units, we utilize the proposed MUM pretext task to pre-train our framework. The [M] denotes the learnable masked token for the input sequence. Meanwhile, the pre-trained tokenizer coupling discretizes the triplet units into pseudo triplet unit labels to supervise the pre-training procedure.

follows,

$$
\begin{aligned}
z_{\mathbf{q}} &= \mathrm{Concat}(h_{k^l}, h_{k^r}, d_{k^b}), z = \mathrm{Concat}(z_l, z_r, z_b), \\
k^l &= \mathrm{Q\_hand}(z_l) = \underset{k\in\{1,2,\cdots,M_1\}}{\arg\min} ||z_l - h_k||, \\
k^r &= \mathrm{Q\_hand}(z_r) = \underset{k\in\{1,2,\cdots,M_1\}}{\arg\min} ||z_r - h_k||, \\
k^b &= \mathrm{Q\_body}(z_b) = \underset{k\in\{1,2,\cdots,M_2\}}{\arg\min} ||z_b - d_k||,
\end{aligned} \tag{1}
$$

where l, r and b indicate the abbreviation of the 'left', 'right' and 'upper body', respectively. We still adopt the abbreviations l, r and b in the following introduction. The latent vector z is made up of three local representations of the pose triplet unit, *i.e.,* z_l, z_r, and z_b. $\mathrm{Q\_hand}(\cdot)$ is the hand quantization encoder that maps the vector to the index of the hand codebook, which satisfies the minimum semantic distance between the input vector and the corresponding codeword. Similar to the hand quantizer, $\mathrm{Q\_body}(\cdot)$ denotes the body quantization encoder for the upper body. The quantized vector $z_{\mathbf{q}}$ consists of a cascade of three codewords from two codebooks. Based on the quantized vector, the decoder aims to reconstruct the input pose triplet unit. The output of the decoder is computed as follows,

$$
\hat{J}_{sign} = \mathrm{Dec}(z_{\mathbf{q}}), \tag{2}
$$

whose output is the reconstructed hand and body poses ($\hat{J}_{left}, \hat{J}_{right}$ and $\hat{J}_{body}$).

This tokenizer is trained before self-supervised pre-training. However, the tokenization process is non-differentiable. To backpropagate the gradient from the decoder to the encoder, we utilize the straight-through estimator (Bengio, Léonard, and Courville 2013) to copy the gradient from the decoder to the encoder. The training objective function of d-VAE is defined as follows,

$$
\begin{aligned}
\mathcal{L}_{d-VAE} &= \mathcal{L}_{hand} + \beta_1 \mathcal{L}_{body} + \beta_2 ||\mathbf{sg}[z] - z_{\mathbf{q}}||_2^2 \\
&\quad + \beta_3 ||\mathbf{sg}[z_{\mathbf{q}}] - z||_2^2, \\
\mathcal{L}_{hand} &= ||\hat{J}_{left} - J_{left}|| + ||\hat{J}_{right} - J_{right}||, \\
\mathcal{L}_{body} &= ||\hat{J}_{body} - J_{body}||,
\end{aligned} \tag{3}
$$

where $\mathcal{L}_{hand}$ and $\mathcal{L}_{body}$ denote the reconstruction losses of 2D hand and upper body pose, respectively. $\mathbf{sg}[\cdot]$ represents the stop-gradient operator, and β_1, β_2 and β_3 are the weighting factors to balance the impact of the four losses.

3.2 Framework Architecture in Pre-training

During pre-training, the framework contains four components, *i.e.,* a pose embedding layer, a pre-trained pose tokenizer, a Transformer encoder, and a decoder.

Pose Embedding Layer. This embedding layer aims to extract the feature embedding from the input pose triplet unit. Since pose has physical connection relationship, we utilize the graph convolutional network (GCN) proposed in (Cai et al. 2019) as the pose embedding layer. Specifically, given a pose sequence $V_{sign} = \{J_{sign,t}\}_{t=1}^{T}$ with T frames, it extracts the pose triplet unit $f_{sign,t} \in \mathbb{R}^D$ from the body, left and right hand in a frame-wise manner and concatenate them for the following module. Each part of the triplet unit has the same feature dimension $D_{part} = \frac{1}{3}D$.

Tokenizer. As presented in Section 3.1, given a pose sequence $\{J_{sign,t}\}_{t=1}^{T}$, we tokenize it as the pseudo labels $\{k_t\}_{t=1}^{T}$ for pre-training, of which k_t denotes the concatenation of k_t^l, k_t^r and k_t^b computed following Equation (1). The tokenizer is only adopted for pseudo label inference, with all parameters frozen during pre-training.

Transformer Encoder. Given the feature sequence from the pose embedding layer $\mathrm{F}_{sign} = \{f_{sign,t}\}_{t=1}^{T}$, we further add temporal embedding $f_{temp,t} \in \mathbb{R}^D$ implemented by the position encoding (Vaswani et al. 2017). The input sequence F_0 is computed as follows,

$$\begin{aligned} \mathrm{F}_0 &= [f_{in,1}, \cdots, f_{in,T}], \\ f_{in,t} &= f_{mask,t} + f_{temp,t}, \quad t \in \{1, \cdots, T\}, \\ \mathrm{F}_m &= \mathrm{Mask}(\mathrm{F}_{sign}) = [f_{mask,1}, \cdots, f_{mask,T}], \end{aligned} \tag{4}$$

where $\mathrm{Mask}(\cdot)$ denotes the masking operator for the embedding sequence and the masked frame positions are denoted as $\mathcal{M} \in \{1, \cdots, T\}^{\alpha \cdot T}$, in which α denotes the mask ratio. We will explain the masked modeling in Section 3.3. The input sequence F_0 is fed into the Transformer encoder. The Transformer encoder contains N layers of Transformer blocks $\mathrm{F}_l = Block(\mathrm{F}_{l-1})$, where $l = 1, \cdots, N$. The output sequence of the last layer $\mathrm{F}_N = [f_{out,1}, \cdots, f_{out,T}]$ is utilized as the encoded representation of the pose triplet unit, where $f_{out,t}$ corresponds to the t-th frame.

Decoder. Given the output sequence F_N, the masked frame positions $\mathcal{M}$ and the pseudo labels $\{k_t\}_{t=1}^{T}$, we utilize the decoder to reconstruct the pose triplet unit. Since each output feature $f_{out,t}$ contains three parts, *i.e.,* $f_{out,t}^{l}$, $f_{out,t}^{r}$ and $f_{out,t}^{b}$, we record the masked positions of each part into $\mathcal{M}^l$, $\mathcal{M}^r$ and $\mathcal{M}^b$, respectively. $\mathcal{M} = \mathcal{M}^l \cup \mathcal{M}^r \cup \mathcal{M}^b$. For each feature $f_{out,t}$, we use the softmax classifier to predict the corresponding label of each part,

$$\begin{aligned} p_{\mathrm{hand}}(k_t^l | f_{out,t}) &= \mathrm{softmax}_{k_t^l}(W_1 f_{out,t}^{l} + b_1), \quad t \in \mathcal{M}^l, \\ p_{\mathrm{hand}}(k_t^r | f_{out,t}) &= \mathrm{softmax}_{k_t^r}(W_1 f_{out,t}^{r} + b_1), \quad t \in \mathcal{M}^r, \\ p_{\mathrm{body}}(k_t^b | f_{out,t}) &= \mathrm{softmax}_{k_t^b}(W_2 f_{out,t}^{b} + b_2), \quad t \in \mathcal{M}^b, \end{aligned} \tag{5}$$

where l, r and b denote the abbreviation of left, right and upper body, respectively. $W_1 \in \mathbb{R}^{|\mathcal{V}_{hand}| \times D_{part}}$ and $W_2 \in \mathbb{R}^{|\mathcal{V}_{body}| \times D_{part}}$ denote the projection matrix. $b_1 \in \mathbb{R}^{|\mathcal{V}_{hand}|}$ and $b_2 \in \mathbb{R}^{|\mathcal{V}_{body}|}$ denote the biases.

3.3 Pretext Task in Pre-training & Objective

Our designed pretext task is MUM, which aims to exploit the hierarchical correlation context among internal and external triplet pose units. Given a pose sequence with a triplet pose unit of length T, we first randomly choose the $\alpha \cdot T$ frames to process the mask operation. For clarification, we define three parts of the pose triplet unit as $f_{sign,t}^{l}$, $f_{sign,t}^{r}$ and $f_{sign,t}^{b}$, respectively. If a unit is masked, a learnable masked token $\mathbf{e}_{mask} \in \mathbb{R}^{D_{part}}$ is utilized to replace each part of the triplet unit with 50% probability. Therefore, the masked triplet unit includes three masking cases: only hand masked, only body masked and hand-body masked. The hand masked case aims to endow our model with the capacity of capturing the local hand context. Compared with the hand counterpart, the body masked case aims to mine the context of global body movements. The hand-body masked case means that body and hand features are masked simultaneously. Since sign language conveys the full meaning with the cooperation of hand and body, we expect to exploit the correlation context cues between hand and body in this case. Therefore, we utilize the MUM to pre-train our method and conduct an ablation experiment to validate our thoughts.

The overall pre-training objective is to maximize the log-likelihood of the correct labels given the corrupted sequence, which is computed as follows,

$$\begin{aligned} \mathcal{L}_{\mathrm{pre-train}} &= \max \sum_{V_{sign} \in \mathcal{D}} \mathbb{E}_{\mathcal{M}} \left[\sum_{t \in \mathcal{M}} \log p(k_t | V_{sign}) \right] \\ \sum_{t \in \mathcal{M}} \log p(k_t | V_{sign}) &= \sum_{t \in \mathcal{M}^l} \log p_{\mathrm{hand}}(k_t^l | f_{out,t}^{l}) \\ &\quad + \sum_{t \in \mathcal{M}^r} \log p_{\mathrm{hand}}(k_t^r | f_{out,t}^{r}) \\ &\quad + \sum_{t \in \mathcal{M}^b} \log p_{\mathrm{body}}(k_t^b | f_{out,t}^{b}), \end{aligned} \tag{6}$$

where $\mathcal{D}$ is the training corpus, $\mathcal{M}$ represents the masked frame positions, $f_{out,t}$ is the masked triplet unit. $\mathcal{M}^l$ and $\mathcal{M}^r$ and $\mathcal{M}^b$ represent the masked positions for left hand, right hand and body, respectively.

3.4 Downstream Fine-Tuning

After pre-training, we directly fine-tune the parameters under the downstream SLR task. We replace the decoder with a simple MLP as the prediction head. During fine-tuning, we do not mask any triplet pose unit, and supervise the output of the prediction head with ground-truth labels. Besides, only the pose information is insufficient to convey the full meaning of sign language. We utilize a late fusion strategy to sum the prediction results of pose data and RGB data. In our experiment, we refer our method with only pose data, fusion of RGB data as **Ours** and **Ours (+R)**, respectively.

4 Experiments

4.1 Datasets and Metrics

Datasets. We conduct experiments on four public sign language datasets, *i.e.,* NMFs-CSL (Hu et al. 2021b), SLR500 (Huang et al. 2018), WLASL (Li et al. 2020a) and MSASL (Joze and Koller 2018). The training sets of all datasets participate in the pre-training stage. Table 1 presents an overview of the above-mentioned datasets.

NMFs-CSL is a large-scale Chinese sign language (CSL) dataset with a vocabulary size of 1,067. All samples are split into 25,608 and 6,402 samples for training and testing, respectively. SLR500 is another CSL dataset including 500 daily words performed by 50 signers. It contains a total of 125,000 samples, of which 90,000 and 35,000 samples are utilized for training and testing, respectively. The above two datasets collect samples in the controlled lab scene.

WLASL is a large-scale American sign language (ASL) dataset, containing 2000 words performed by over 100 signers. It totally consists of 21,083 samples. In particular, it selects the top-K most frequent words with

Name	Language	Vocab.	Videos	Signers	Source
WLASL	ASL	2000	21.1k	119	Web, lexicons
NMF-CSL	CSL	1067	32.0k	10	lab
MSASL	ASL	1000	25.5k	222	Web, lexicons
SLR500	CSL	500	125k	50	lab

Table 1: Statistics of public isolated SLR datasets. ASL denotes American Sign Language, and CSL denotes Chinese Sign Language.

$K = \{100, 300\}$, and organize them as two subsets, namely WLASL100 and WLASL300. MSASL is another popular ASL dataset with a total of 25,513 samples and a vocabulary size of 1000. Similar to WLASL, it also provides two subsets, named MSASL100 and MSASL200, respectively. Different from NMFs-CSL and SLR500, WLASL and MSASL collect data from the Web and bring more challenges due to the unrestricted real-life scenario.

Metrics. For evaluation, we report the classification accuracy, *i.e.,* Top-1 (**T-1**) and Top-5 (**T-5**) for the downstream SLR task. We adopt both per-instance (**P-I**) and per-class (**P-C**) accuracy metrics following (Li et al. 2020a; Joze and Koller 2018). Per-instance accuracy is computed over whole test data. Per-class accuracy is the average of the sign categories present in the test set. For WLASL and MSASL, we report both per-instance and per-class accuracy due to the unbalanced samples per class of the two datasets. For NMFs-CSL and SLR500, we only report per-instance accuracy with an equal number of samples for each class.

4.2 Experiment Settings

Data Preparation and Processing. Our proposed method utilizes the pose modality to represent hand and body movements. Since no available pose annotation is provided in sign language datasets, we utilize the off-the-shelf pose detector MMPose to extract the 2D pose keypoints. In each frame, the extracted 2D pose includes 49 joints, containing 7 upper body joints and 42 hand joints. Furthermore, considering the different scales among triplet units, we crop the body, left and right hand pose separately and rescale them to 256×256.

Model Hyper-Parameters. For the tokenizer, the vocabulary size of hand codebook M_1 and body codebook M_2 are 1000 and 500, respectively. The dimension of each codeword in two codebooks is 512. The weighting factors β_1, β_2 and β_3 in equation 3 are set to 0.1, 1.0 and 0.9, respectively. During pre-training, the Transformer encoder contains 8 heads with the input size of the Transformer encoder D as 1536 and position-wise feed-forward dimension as 2048.

Training Setup. The Adam (Kingma and Ba 2014) optimizer is employed in our experiments. For tokenizer training, we set the initial learning rate as 0.001 and decrease it with a factor of 0.1 per 10 epochs. For pre-training, the weight decay and momentum are set to 0.01 and 0.9, respectively. The learning rate is set to 0.0001, with a warmup of 6 epochs, and linear learning rate decay. For the downstream SLR task, the learning rate is initialized to 0.0001 and decreases by a factor of 0.1 per 10 epochs. we disturb the coordinates of the input pose sequence with a random perturbation matrix to relieve overfitting during training. Following (Hu et al. 2021a), we temporally select 32 frames using random and center temporal sampling during training and testing, respectively. All experiments are implemented by PyTorch and trained on NVIDIA RTX 3090.

4.3 Comparison with State-of-the-Art Methods

In this section, we compare our method with previous state-of-the-art methods on four public datasets. Following (Hu et al. 2021a), we group the previous methods by the input modality, *i.e.,* pose-based and RGB-based methods.

NMFs-CSL Dataset. As shown in Table 5, we compare with the previous methods. SignBERT (Hu et al. 2021a) is a self-supervised pre-training method with hand prior to model the hand sequence. Compared with SignBERT, our method still achieves the best performance in top-1 accuracy under different settings. GLE-Net (Hu et al. 2021b) is the state-of-the-art method with discriminative clues from global and local views. Compared with GLE-NET, Ours (+R) outperforms it, achieving 79.2% top-1 accuracy.

MSASL Dataset. As shown in Table 2, ST-GCN (Yan, Xiong, and Lin 2018) shows inferior performance compared with RGB-based methods, which may be attributed to the failure of pose detection. Compared with ST-GCN (Yan, Xiong, and Lin 2018), self-supervised learning methods, *i.e.,* SignBERT (Hu et al. 2021a) and Ours, relieve this issue by leveraging the stored statistics during pre-training. Our method outperforms SignBERT (Hu et al. 2021a) with 4.89%, 5.96% and 9.28% Top-1 per-instance accuracy improvement on MSASL100, MSASL200 and MSASL1000, respectively. Notably, our method even achieves comparable performance with RGB-based methods. When fused with the RGB method, the performance is further improved.

WLASL Dataset. Similar to MSASL, WLASL is a challenging dataset with unconstrained recording conditions and unbalanced sample distribution. As shown in Table 3, Pose-TGCN (Li et al. 2020a) and PSLR (Tunga, Nuthalapati, and Wachs 2021) show inferior performance caused by the erroneous estimation of poses. HMA (Hu, Zhou, and Li 2021) utilizes a hand statistic model to refine the pose and improve the performance. BSL (Albanie et al. 2020) and TCK (Li et al. 2020b) utilize external RGB data to enhance the model robustness and boost their performance. Compared with them, our method achieves the best performance on all subsets.

SLR500 Dataset. As shown in Table 4, compared with STIP (Laptev 2005) and GMM-HMM (Tang et al. 2015) based on hand-craft features, deep-learning-based methods (Yan, Xiong, and Lin 2018; Qiu, Yao, and Mei 2017; Hu et al. 2021a) achieve notable performance gain. Our method still achieves new state-of-the-art performance among both pose-based and RGB-based methods.

4.4 Ablation Study

In this section, we conduct ablation studies to validate the effectiveness of our proposed approach and select proper

Method	MSASL100 P-I T-1	MSASL100 P-I T-5	MSASL100 P-C T-1	MSASL100 P-C T-5	MSASL200 P-I T-1	MSASL200 P-I T-5	MSASL200 P-C T-1	MSASL200 P-C T-5	MSASL1000 P-I T-1	MSASL1000 P-I T-5	MSASL1000 P-C T-1	MSASL1000 P-C T-5
Pose-based												
ST-GCN (Yan, Xiong, and Lin 2018)	59.84	82.03	60.79	82.96	52.91	76.67	54.20	77.62	36.03	59.92	32.32	57.15
SignBERT (Hu et al. 2021a)	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	**80.98**	**95.11**	**81.24**	**95.44**	**76.60**	**91.54**	**76.75**	**91.95**	**58.82**	**81.18**	**54.87**	**80.05**
RGB-based												
I3D (Carreira and Zisserman 2017)	-	-	81.76	95.16	-	-	81.97	93.79	-	-	57.69	81.05
TCK (Li et al. 2020b)	83.04	93.46	83.91	93.52	80.31	91.82	81.14	92.24	-	-	-	-
BSL (Albanie et al. 2020)	-	-	-	-	-	-	-	-	64.71	85.59	61.55	84.43
HMA (Hu, Zhou, and Li 2021)	73.45	89.70	74.59	89.70	66.30	84.03	67.47	84.03	49.16	69.75	46.27	68.60
Ours (+R)	**89.56**	**96.96**	**90.08**	**97.07**	**86.83**	**95.66**	**87.45**	**95.72**	**71.21**	**88.85**	**68.24**	**87.98**

Table 2: Comparison with state-of-the-art methods on MSASL dataset. Our proposed method fused with another RGB-based method, I3D (Carreira and Zisserman 2017), is represented by Ours (+R).

Method	WLASL100 P-I T-1	WLASL100 P-I T-5	WLASL100 P-C T-1	WLASL100 P-C T-5	WLASL300 P-I T-1	WLASL300 P-I T-5	WLASL300 P-C T-1	WLASL300 P-C T-5	WLASL2000 P-I T-1	WLASL2000 P-I T-5	WLASL2000 P-C T-1	WLASL2000 P-C T-5
Pose-based												
ST-GCN (Yan, Xiong, and Lin 2018)	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 (Li et al. 2020a)	55.43	78.68	-	-	38.32	67.51	-	-	23.65	51.75	-	-
PSLR (Tunga, Nuthalapati, and Wachs 2021)	60.15	83.98	-	-	42.18	71.71	-	-	-	-	-	-
SignBERT (Hu et al. 2021a)	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	**77.91**	**91.47**	**77.83**	**92.50**	**67.66**	**89.22**	**68.31**	**89.57**	**46.25**	**79.33**	**43.52**	**77.65**
RGB-based												
I3D (Carreira and Zisserman 2017)	65.89	84.11	67.01	84.58	56.14	79.94	56.24	78.38	32.48	57.31	-	-
TCK (Li et al. 2020b)	77.52	91.08	77.55	91.42	68.56	89.52	68.75	89.41	-	-	-	-
BSL (Albanie et al. 2020)	-	-	-	-	-	-	-	-	46.82	79.36	44.72	78.47
HMA (Hu, Zhou, and Li 2021)	-	-	-	-	-	-	-	-	37.91	71.26	35.90	70.00
Ours (+R)	**81.01**	**94.19**	**81.63**	**94.67**	**75.60**	**92.81**	**76.12**	**93.07**	**54.59**	**88.08**	**52.12**	**87.28**

Table 3: Comparison with state-of-the-art methods on WLASL dataset. Our proposed method fused with another RGB-based method, I3D (Carreira and Zisserman 2017), is represented by Ours (+R).

Method	Accuracy
Pose-based	
ST-GCN (Yan, Xiong, and Lin 2018)	90.0
SignBERT (Hu et al. 2021a)	94.5
Ours	**95.4**
RGB-based	
STIP (Laptev 2005)	61.8
GMM-HMM (Tang et al. 2015)	56.3
3D-R50 (Qiu, Yao, and Mei 2017)	95.1
HMA (Hu, Zhou, and Li 2021)	95.9
GLE-Net (Hu et al. 2021b)	96.8
Ours (+R)	**97.7**

Table 4: Comparison with state-of-the-art methods on SLR500 dataset. 3D-R50 (Qiu, Yao, and Mei 2017) is utilized for fusion with our method.

hyper-parameters for our framework. For fair comparison, experiments are performed on the MSASL dataset and we report the top-1 accuracy under the per-instance and per-class metrics as the indicator.

Different Tokenizers. In Table 6, we compare the impact of different tokenizers on the downstream SLR task. The K-Means method is a simple model-free clustering algorithm. Different from the learnable tokenizer, K-Means iteratively transfers the 2D pose data to a series of clustering centers for hand and body, respectively. Then we directly utilize the index of clustering centers as the pseudo labels. For fair comparison, the number of clustering centers is the same as the vocabulary size of our utilized codebook. VQ and Ours stand for separate and coupled vector quantizers, respectively. VQ learns the hand and body codebooks separately, while our proposed tokenizer jointly optimizes the hand and body codebooks. It can be observed that our proposed tokenizer shows the best performance to validate our assumption. Since the body and hand are strongly correlated, our coupling tokenization utilizes this cue for better codebook learning. Moreover, we illustrate the qualitative results of our proposed coupling tokenizer in Figure 2. Our tokenizer can successfully cluster similar pose triplet units under the unconstrained settings, *i.e.,* individual pose variance and inaccurate pose detection.

Masking Strategy. As shown in Table 7, we compare three different settings on our proposed MUM, *i.e.,* only mask-

Method	Total	Confusing	Normal
	T-1	T-1	T-1
Pose-based			
ST-GCN (Yan, Xiong, and Lin 2018)	59.9	42.2	83.4
SignBERT (Hu et al. 2021a)	67.0	46.4	94.5
Ours	**68.5**	**49.0**	**94.6**
RGB-based			
3D-R50 (Qiu, Yao, and Mei 2017)	62.1	43.1	87.4
DNF (Cui, Liu, and Zhang 2019)	55.8	51.9	86.3
I3D (Carreira and Zisserman 2017)	64.4	47.3	87.1
TSM (Lin, Gan, and Han 2019)	64.5	42.9	93.3
Slowfast (Feichtenhofer et al. 2019)	66.3	47.0	92.0
GLE-Net (Hu et al. 2021b)	69.0	50.6	93.6
HMA (Hu, Zhou, and Li 2021)	64.7	42.3	94.6
Ours (+R)	**79.2**	**65.5**	**97.5**

Table 5: Comparison with state-of-the-art methods on NMFs-CSL dataset. Our proposed method fused with another RGB-based method, 3D-R50 (Qiu, Yao, and Mei 2017), is represented by Ours (+R).

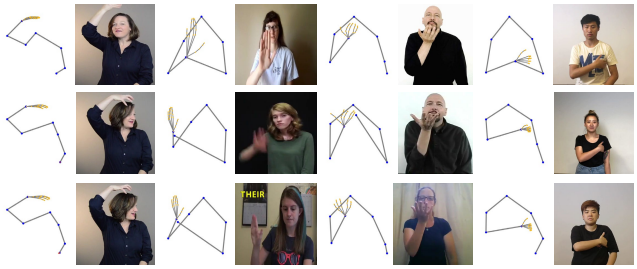

Figure 2: Qualitative results of our proposed coupling tokenization. Each two adjacent columns are organized as a group. In the same row of each group, the pose corresponds to the RGB image. Each group belongs to the same triplet unit label.

ing hand, only masking body and masking both parts (hand and body). The first row denotes the baseline without pre-training. Only masked hand setting shows better performance than only masked body setting due to the dominance of hand during sign language expression. Our MUM adopts the third masking setting, which achieves the best performance. Compared with the baseline, our proposed MUM achieves performance gain, *i.e.,* 7.54%, 5.81% and 6.86% for per-instance Top-1 accuracy improvement, respectively.

Pre-Training Data Scale. In Table 8, we investigate the effect of the pre-training data scale. The first row denotes that our proposed framework is the method without pre-training. It is clearly observed that the performance gradually increases with the increment in the proportion of pre-training data. The result demonstrates that our proposed method is applicable to the pre-training for large-scale data.

5 Conclusion

In this paper, we propose a self-supervised pre-trainable framework namely BEST, which leverages the success of BERT with the specific design to the sign language domain. Focusing on the main properties during sign language expression, we organize the hand and body movements as the pose triplet unit. During pre-training, we propose the masked unit modeling (MUM) pretext task to exploit the hierarchical context cues among internal and external triplet units. To make the original BERT objective applicable, we attempt to bridge the semantic gap of pseudo label via coupling tokenization on the triplet unit. Extensive experiments validate the effectiveness of our method, achieving new state-of-the-art performance on all four benchmarks.

Quantizer	MSASL100		MSASL200		MSASL1000	
	P-I	P-C	P-I	P-C	P-I	P-C
K-Means	74.63	74.03	72.62	74.20	54.29	50.74
VQ	77.54	77.44	74.47	75.47	55.87	52.16
Ours	**80.98**	**81.24**	**76.60**	**76.75**	**58.82**	**54.87**

Table 6: Comparison with different pose tokenizers on MSASL dataset. K-Means denotes a model-free clustering algorithm. VQ and Ours stand for separate and coupled vector tokenizers, respectively.

Mask		MSASL100		MSASL200		MSASL1000	
Hand	Body	P-I	P-C	P-I	P-C	P-I	P-C
		73.44	73.75	70.79	71.46	51.96	48.50
✓		76.75	77.22	71.37	72.07	54.60	51.63
	✓	68.82	68.82	67.11	67.84	47.72	43.87
✓	✓	**80.98**	**81.24**	**76.60**	**76.75**	**58.82**	**54.87**

Table 7: Effect of the masking strategy on MSASL dataset. The first row denotes the baseline method without pre-training. "Hand" and "Body" denote the only masking on the hand and body part, respectively.

Percent	MSASL100		MSASL200		MSASL1000	
	P-I	P-C	P-I	P-C	P-I	P-C
0%	73.44	73.75	70.79	71.46	51.96	48.50
25%	74.50	75.20	71.89	72.69	53.67	50.81
50%	77.54	78.06	73.43	75.03	54.69	51.47
75%	78.47	78.77	74.54	75.32	56.06	53.16
100%	**80.98**	**81.24**	**76.60**	**76.75**	**58.82**	**54.87**

Table 8: Effect of the data scale during pre-training on MSASL dataset. The "Percent" denotes the proportion of pre-training data.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Contract U20A20183 and 62021001. It was also supported by GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

Albanie, S.; Varol, G.; Momeni, L.; Afouras, T.; Chung, J. S.; Fox, N.; and Zisserman, A. 2020. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *ECCV*, 35–53.

Bao, H.; Dong, L.; and Wei, F. 2021. BEiT: BERT pre-training of image transformers. In *ICLR*, 1–18.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv*, 1–12.

Cai, Y.; Ge, L.; Liu, J.; Cai, J.; Cham, T.-J.; Yuan, J.; and Thalmann, N. M. 2019. Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In *ICCV*, 2272–2281.

Camgoz, N. C.; Hadfield, S.; Koller, O.; Ney, H.; and Bowden, R. 2018. Neural sign language translation. In *CVPR*, 7784–7793.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 6299–6308.

Conneau, A.; and Lample, G. 2019. Cross-lingual language model pretraining. In *NeurIPS*, 1–11.

Cui, R.; Liu, H.; and Zhang, C. 2019. A deep neural framework for continuous sign language recognition by iterative training. *IEEE TMM*, 21(7): 1880–1891.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 4171–4186.

Du, Y.; Wang, W.; and Wang, L. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 1110–1118.

Farhadi, A.; Forsyth, D.; and White, R. 2007. Transfer learning in sign language. In *CVPR*, 1–8.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *ICCV*, 6202–6211.

Fillbrandt, H.; Akyol, S.; and Kraiss, K.-F. 2003. Extraction of 3D hand shape and posture from image sequences for sign language recognition. In *ISO4*, 181–186.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *CVPR*, 16000–16009.

Hu, H.; Zhao, W.; Zhou, W.; Wang, Y.; and Li, H. 2021a. SignBERT: Pre-Training of Hand-Model-Aware Representation for Sign Language Recognition. In *ICCV*, 11087–11096.

Hu, H.; Zhou, W.; and Li, H. 2021. Hand-model-aware sign language recognition. In *AAAI*, 1558–1566.

Hu, H.; Zhou, W.; Pu, J.; and Li, H. 2021b. Global-local enhancement network for NMF-aware sign language recognition. *ACM TOMM*, 17(3): 1–19.

Huang, J.; Zhou, W.; Li, H.; and Li, W. 2018. Attention-based 3D-CNNs for large-vocabulary sign language recognition. *IEEE TCSVT*, 29(9): 2822–2832.

Jiang, T.; Camgoz, N. C.; and Bowden, R. 2021. Skeletor: Skeletal transformers for robust body-pose estimation. In *CVPR*, 3394–3402.

Jin, T.; and Zhao, Z. 2021. Contrastive Disentangled Meta-Learning for Signer-Independent Sign Language Translation. In *ACM MM*, 5065–5073.

Joze, H. R. V.; and Koller, O. 2018. MS-ASL: A large-scale data set and benchmark for understanding american sign language. In *BMVC*, 1–16.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv*, 1–15.

Koller, O.; Zargaran, S.; Ney, H.; and Bowden, R. 2018. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *IJCV*, 126(12): 1311–1325.

Laptev, I. 2005. On space-time interest points. *IJCV*, 64(2): 107–123.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 7871–7880.

Li, C.; Zhong, Q.; Xie, D.; and Pu, S. 2018. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, 786–792.

Li, D.; Rodriguez, C.; Yu, X.; and Li, H. 2020a. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *WACV*, 1459–1469.

Li, D.; Yu, X.; Xu, C.; Petersson, L.; and Li, H. 2020b. Transferring cross-domain knowledge for video sign language recognition. In *CVPR*, 6205–6214.

Lin, J.; Gan, C.; and Han, S. 2019. TSM: Temporal shift module for efficient video understanding. In *ICCV*, 7083–7093.

Min, Y.; Zhang, Y.; Chai, X.; and Chen, X. 2020. An efficient pointlstm for point clouds based gesture recognition. In *CVPR*, 5761–5770.

Momeni, L.; Bull, H.; Prajwal, K.; Albanie, S.; Varol, G.; and Zisserman, A. 2022. Automatic dense annotation of large-vocabulary sign language videos. In *ECCV*, 671–690.

Ng, E.; Ginosar, S.; Darrell, T.; and Joo, H. 2021. Body2hands: Learning to infer 3D hands from conversational gesture body dynamics. In *CVPR*, 11865–11874.

Niu, Z.; and Mak, B. 2020. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In *ECCV*, 172–186.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3D residual networks. In *ICCV*, 5533–5541.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831.

Selvaraj, P.; Nc, G.; Kumar, P.; and Khapra, M. 2022. OpenHands: Making Sign Language Recognition Accessible with Pose-based Pretrained Models across Languages. In *ACL*, 2114–2133.

Sincan, O. M.; and Keles, H. Y. 2020. Autsl: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access*, 8: 181340–181355.

Starner, T. E. 1995. Visual Recognition of American Sign Language Using Hidden Markov Models. Technical report, MIT.

Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; and Dai, J. 2019. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*, 1–16.

Sun, C.; Myers, A.; Vondrick, C.; Murphy, K.; and Schmid, C. 2019. VideoBERT: A joint model for video and language representation learning. In *ICCV*, 7464–7473.

Tang, A.; Lu, K.; Wang, Y.; Huang, J.; and Li, H. 2015. A real-time hand posture recognition system using deep neural networks. *ACM TIST*, 6(2): 1–23.

Tunga, A.; Nuthalapati, S. V.; and Wachs, J. 2021. Pose-Based Sign Language Recognition Using GCN and BERT. In *WACV*, 31–40.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. In *NeurIPS*, 1–11.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 6000–6010.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 1–10.

Zhu, L.; and Yang, Y. 2020. ActBERT: Learning global-local video-text representations. In *CVPR*, 8746–8755.