Skeleton2vec: A Self-supervised Learning Framework with Contextualized Target Representations for Skeleton Sequence

Anonymous ECCV 2024 Submission

Paper ID #8436

Abstract. Self-supervised pre-training paradigms have been extensively explored in the field of skeleton-based action recognition. In particular, methods based on **masked prediction** have pushed the performance of pre-training to a new height. However, these methods take low-level features, such as raw joint coordinates or temporal motion, as prediction targets for the masked regions, which is suboptimal. In this paper, we show that using high-level contextualized features as prediction targets can achieve superior performance. Specifically, we propose **Skeleton2vec**, a simple and efficient self-supervised 3D action representation learning framework, which utilizes a transformer-based teacher encoder taking unmasked training samples as input to create **latent contextualized representations** as prediction targets. Benefiting from the self-attention mechanism, the latent representations generated by the teacher encoder can incorporate the global context of the entire training samples, leading to a richer training task. Additionally, considering the high temporal correlations in skeleton sequences, we propose a **Motion-Aware Multi-Tube masking strategy** which divides the skeleton sequence into multiple tubes and performs persistent masking within each tube based on motion priors, thus forcing the model to build long-range spatio-temporal connections and focus on action-semantic richer regions. Extensive experiments on NTU-60, NTU-120, and PKU-MMD datasets demonstrate that our proposed Skeleton2vec outperforms previous methods and achieves state-of-the-art results. The code will be made available after the paper is accepted for publication.

Keywords: Self-supervised 3D action recognition · Masking prediction

1 Introduction

Human action recognition has significant applications in the real world, such as security, human-robot interaction, and virtual reality. The development of depth sensors and advancements in pose estimation algorithms [4,12,42] have propelled skeleton-based action recognition into a popular research topic, owing to its computational efficiency, background robustness, and privacy preservation. A series of fully-supervised skeleton-based human action recognition methods have been developed using CNNs [10,19], RNNs [25,47], and GCNs [6,44]. Despite their

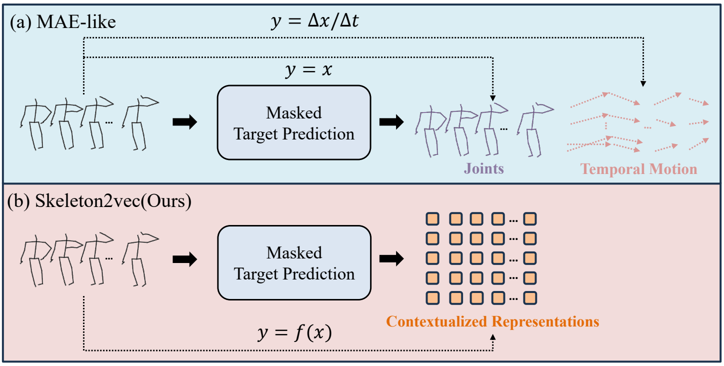


Fig. 1: A comparative illustration of the prediction targets between MAE-like methods (a) and ours Skeleton2vec (b). Skeleton2vec utilizes an teacher encoder $f(x)$ to generate globally contextualized representations as the prediction targets, instead of isolated joints or temporal motion with only local context.

promising performance, these methods rely on large amounts of manually annotated data, which is expensive, labor-intensive, and time-consuming to obtain. This circumstance motivates us to explore self-supervised representation learning for 3D actions.

Earlier works [21,30,34,48] have employed various pretext tasks, such as motion prediction, jigsaw puzzle recognition, and masked reconstruction, to learn 3D action representations. Recently, contrastive learning methods [15,22,29,31] have gained prominence. However, these methods often require carefully designed data augmentations and tend to encourage the encoder to learn more global representations, thereby neglecting local spatiotemporal information. With the rise of transformer models [38], self-supervised pre-training methods based on masked prediction tasks have become mainstream in visual representation learning [15, 22, 29, 31]. Works like SkeletonMAE [40, 43] and MAMP [27] have attempted to transfer MAE [16] methods to the field of 3D action representation learning, achieving promising results. However, these MAE-like methods inefficiently utilize model capacity by focusing on low-level high-frequency details with raw joint coordinates or temporal motion as learning targets, which is suboptimal for modeling high-level spatiotemporal structures. We believe that using higher-level prediction targets will guide the model to learn better representations and improve pre-training performance.

Motivated by this idea, we propose Skeleton2vec, a simple and efficient self-supervised framework for 3D action representation learning. Addressing the limitations of existing MAE-like methods, as illustrated in Fig. 1, Skeleton2vec leverages contextualized prediction targets. Following the work of data2vec [1,2], we employ a teacher encoder that takes unmasked training samples to generate latent contextualized representations as targets. We then use a student encoder, taking a masked version of the sample as input, combined with an asymmetric decoder to predict data representations at the masked positions. The entire model is based on the vanilla transformer architecture. The self-attention mechanism ensures that the constructed targets are contextualized, incorporating information from the entire sample, making them richer than isolated targets (*e.g.* raw joint coordinates) or targets based on local context (*e.g.* temporal motion).

Additionally, considering the strong spatiotemporal correlations in 3D skeleton sequences, we propose a Motion-Aware Multi-Tube (MAMT) masking strategy. Initially, we divide the input skeleton sequence along the temporal axis into multiple tubes, where frames within each tube share a masking map to avoid information leakage from neighboring frames. This forces the model to extract information from distant time steps for better prediction. We then guide the sampling of masked joints based on the spatial motion intensity of body joints within each tube. Joints with higher motion intensity will be masked with higher probability, allowing the model to focus more on spatiotemporal regions with rich action semantics. Compared to random masking, our method better utilizes the spatiotemporal characteristics and motion priors of 3D skeleton sequences, effectively improving pre-training performance.

In summary, the main contributions of this work are three-fold:

- We propose the Skeleton2vec framework, which uses contextualized representations from a teacher encoder as prediction targets, enabling the learned representations to have stronger semantic associations.
- We introduce a motion-aware multi-tube masking strategy that performs persistent masking of joints within tubes based on spatial motion intensity, forcing the model to build better long-range spatiotemporal connections and focus on more semantic-rich regions.
- We validate the effectiveness of our method on three large-scale 3D skeleton-based action recognition datasets and achieve state-of-the-art results.

2 Related Work

2.1 Self-supervised Skeleton-based Action Recognition

Previous studies [21,34,48] on self-supervised representation learning for skeleton-based action recognition utilize various pretext tasks to capture motion context. For instance, LongTGAN [48] leverages sequence reconstruction to learn 3D action representations. P&C [34] employs a weak decoder to enhance representation learning. MS2L [21] employs motion prediction and jigsaw puzzle tasks. Yang et al. [45] introduce a skeleton cloud colorization task. Contrastive learning methods have gained prominence in 3D action representation learning [14, 15, 17, 22, 29, 31]. AS-CAL [31] and SkeletonCLR [20] utilize momentum encoder and propose various data augmentation strategies. AimCLR [15] introduces extreme augmentations. ActCLR [22] performs adaptive action modeling on different body parts. Despite their remarkable results, contrastive learning methods often overlook local spatio-temporal information, a crucial aspect for 3D action modeling.

The surge in popularity of transformers has led to the mainstream adoption of self-supervised pretraining based on masked visual modeling for visual representation learning [3, 16]. SkeletonMAE [40] and MAMP [27] apply the Masked Autoencoder (MAE) approach to 3D action representation learning. SkeletonMAE employs a skeleton-based encoder-decoder transformer for spatial coordinate reconstruction, while MAMP introduces Masked Motion Prediction to explicitly model temporal motion. In this study, we demonstrate that utilizing higher-level contextualized representations as prediction targets for masked regions yields superior performance compared to directly predicting raw joint coordinates or temporal motion.

2.2 Masked Image Modeling

BEiT [3] pioneered masked image modeling (MIM) for self-supervised pretraining of visual models, aiming to recover discrete visual tokens from masked patches. Subsequently, various prediction targets for MIM have been explored. MAE [16] and SimMIM [41] treat MIM as a denoising self-reconstruction task, utilizing raw pixels as the prediction target. MaskFeat [39] replaces pixels with HOG descriptors to enable more efficient training and achieve superior results. PeCo [8] introduces a perceptual loss during dVAE training to generate semantically richer discrete visual tokens, surpassing BEiT. These works demonstrate superior performance by utilizing higher-level and semantically richer prediction targets in MIM. To further enhance performance, data2vec [1,2] employs self-distillation to leverage latent target representations from the teacher model output at masked positions. Compared to isolated targets like visual tokens or pixels, these contextualized representations encompass relevant features from the entire image, enabling improved performance.

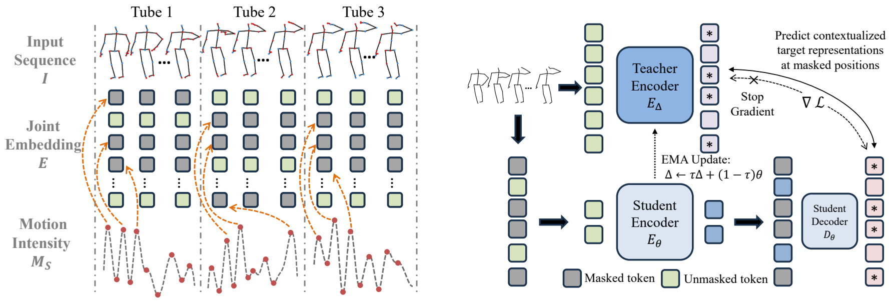
In this research, we introduce the data2vec framework into self-supervised pretraining of skeleton sequences, utilizing latent contextualized target representations from the teacher model to guide the student model in learning more effective 3D action representations.

3 Method

3.1 Overview

The overall framework of Skeleton2vec is shown in Fig. 2. It takes a skeleton sequence $I \in \mathbb{R}^{T_s \times V \times C_s}$ as input, where T_s is the the number of frames, V is the number of joints, and C_s is the the coordinates of joints. Similar to most visual transformers [9], the skeleton sequence is first divided into fixed-size patches and then linearly transformed into patch embedding $E \in \mathbb{R}^{T_e \times V \times C_e}$. After that, we employ the motion-aware multi-tube masking strategy to guide the masking of joints. The teacher model constructs the full contextualized prediction targets using unmasked training samples, while the student model receives the masked version of the samples and predicts corresponding representations at the masked positions.

As our student model, we adopt an asymmetric encoder-decoder architecture, where the encoder operates solely on non-masked tokens. The lightweight


Tube 1
Tube 2
Tube 3
Input Sequence I
Joint Embedding E
Motion Intensity M_S
Teacher Encoder E_Δ
Predict contextualized target representations at masked positions
Stop Gradient
$\nabla \mathcal{L}$
EMA Update: $\Delta \leftarrow \tau\Delta + (1-\tau)\theta$
Student Encoder E_θ
Student Decoder D_θ
Masked token
Unmasked token
*

(a) Motion-Aware Multi-Tube Masking **(b)** Skeleton2vec

Fig. 2: The overall pipeline of the proposed Skeleton2vec framework. We adopt the motion-aware multi-tube masking strategy (a) to guide the masking process, which prevents information leakage between adjacent frames and allows the model to focus more on semantically rich regions of motion. Subsequently, the teacher encoder E_Δ receives unmasked samples to construct latent contextualized targets, while the student encoder E_θ receives masked versions of the samples and predicts corresponding representations at the masked positions.

decoder inserts masked tokens into the latent representations outputted by the encoder, forming a full set for predicting the targets. The teacher encoder shares the same model structure as the student. After accomplishing the aforementioned pre-training task, the teacher encoder is retained for downstream task fine-tuning.

3.2 Model Architecture

Encoder: Following MAMP [27], we first divide the raw skeleton sequence $I \in \mathbb{R}^{T_s \times V \times C_s}$ into non-overlapping segments $I' \in \mathbb{R}^{T_e \times V \times (l \cdot C_s)}$, where $T_e = T_s/l$ and l is the length of each segment. A trainable linear projection is then applied to each joint to obtain the embedding:

$$E_j = \mathrm{LinearProj}(I') \in \mathbb{R}^{T_e \times V \times C_e}, \tag{1}$$

where C_e represents the dimension of the embedding. Temporal positional embedding $E_t \in \mathbb{R}^{T_e \times 1 \times C_e}$ and spatial positional embedding $E_s \in \mathbb{R}^{1 \times V \times C_e}$ are then added to the joint embedding to yield the final input:

$$E = E_j + E_t + E_s, \tag{2}$$

For the teacher encoder, the entire set is flattened as input $E^T \in \mathbb{R}^{N_T \times C_e}$, where $N_T = T_e \times V$ represents the total number of tokens in the skeleton sequence. For the student encoder, most tokens are masked, and only the unmasked tokens are utilized as input, flattened as $E^S \in \mathbb{R}^{N_S \times C_e}$, where $N_S = T_e \times V \times (1-m)$ denotes the number of visible tokens, and m is the masking ratio. Subsequently, L_e layers of vanilla transformer blocks are applied to extract latent representations. Each block comprises a multi-head self-attention (MSA) module and a feed-forward network (FFN) module. Residual connections are employed within each module, followed by layer normalization (LN).

Decoder: The decoder input $D \in \mathbb{R}^{T_e \times V \times C_e}$ contains the full set of tokens, including the latent representations of visible encoded tokens Z_e^S and the inserted masked tokens. Each masked token is represented by a shared learnable vector $E_M \in \mathbb{R}^{C_e}$, indicating missing information to be predicted at that position. Similar to the encoder, spatial positional embedding E'_s and temporal positional embedding E'_t are added to all tokens to assist masked tokens in locating their positions. The decoder employs an additional L_d layers of transformer blocks for masked prediction.

3.3 Contextualized Target Prediction

Rather than relying on isolated raw joints or temporal motion with limited local context, we employ a transformer-based teacher encoder to construct globally contextualized prediction targets, thereby introducing a diverse training task.

Contextualized Target Representations: We extract features from the output of each FFN block in every layer of the teacher encoder and average them to form our training targets. Following data2vec 2.0 [1], the features from each layer are normalized with instance normalization [37] before averaging. Finally, the averaged features are normalized by layer normalization to serve as the prediction targets. Normalizing the targets helps prevent the model from collapsing to a trivial solution, and also prevents any single layer's features from dominating. The generation of the target representations can be formulated as:

$$\begin{aligned} Y' &= \frac{1}{L_e} \sum_{l=1}^{L_e} \mathrm{IN}(Z_l^T), \\ Y &= \mathrm{LN}(Y'), \end{aligned} \tag{3}$$

where IN and LN refer to instance normalization and layer normalization, respectively. Z_l^T denotes the output of the FFN block in the l^{th} layer of the teacher encoder.

Target Prediction: Given the output H_d of the student decoder, we employ an additional linear prediction head to regress the contextualized target representations of the teacher:

$$\hat{Y} = \mathrm{LinearPred}(H_d), \tag{4}$$

Finally, we adopt L2 loss as our learning objective, calculating loss only for the masked positions:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} ||Y_i - \hat{Y}_i||_2^2, \tag{5}$$

where $\mathcal{M}$ denotes the set of masked positions.

Teacher Parameterization: The student model weights θ are updated through backpropagation on the loss gradients. The teacher model weights Δ are initialized to be the same as the student weights and parameterized during training by taking an exponentially moving average (EMA) of the student weights:

$$\Delta \leftarrow \tau \Delta + (1-\tau)\theta, \tag{6}$$

where τ is a hyperparameter controlling the update frequency of the teacher weights using a linearly increasing schedule, gradually increasing from an initial value τ_0 to 1 throughout training.

3.4 Motion-Aware Multi-Tube Masking

We propose the motion-aware multi-tube masking strategy to address the issue of high spatiotemporal correlations in skeleton sequences.

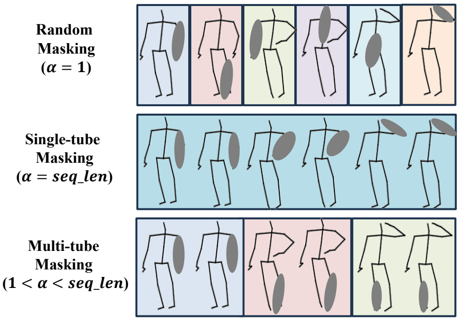

Random Masking (α = 1)
Single-tube Masking (α = *seq_len*)
Multi-tube Masking (1 < α < *seq_len*)

Fig. 3: Intuitive comparison of different masking strategies. The gray shade represents the masked region, with different colors denoting different masking maps, while *seq_len* indicates the length of the sequence.

Multi-Tube Division: The tube masking strategy, initially introduced by VideoMAE [36], considers the entire video sequence along the temporal axis as a single tube, sharing the same masking map across different frames. This mitigates the information leakage issue between adjacent frames. Although the skeleton sequence is derived from the video, directly applying this single-tube masking strategy to skeleton data is suboptimal due to the inherent structural differences. In video data, the basic units for masking are image patches in each frame. Due to scene motion or camera viewpoint changes, a masked body part like the hand in the first frame may find its correspondence in unmasked regions in later frames far apart, which facilitates long-range dependency modeling. In contrast, the basic units for masking in skeleton sequences are the joints in each skeleton frame, where the same-order joints have explicit correspondence across frames. As a result, a body part masked in the first skeleton frame will remain masked in all frames, causing a complete loss of information for that part (as shown in the second row of Fig. 3), which makes the masked prediction task overly difficult and harms the model's learning capability. To address this, as illustrated in Fig. 2a and Fig. 3, we empirically divide the skeleton sequence along the time axis into multiple tubes instead of one tube. Each tube shares the same masking map to force the model to extract information from farther time steps, while different tubes use different masking maps to avoid joints being masked throughout. The tube division can be represented as:

$$E' = \text{Reshape}(E) \in \mathbb{R}^{N \times \alpha \times V \times C_e}, \tag{7}$$

where α is tube length and $N = \frac{T_e}{\alpha}$ is number of tubes. It can be observed in Fig. 3 that random masking samples a different masking map for each frame, single-tube masking shares a single masking map across all frames, while our multi-tube masking is a compromise between the two, sharing the masking map only among several adjacent frames. This approach mitigates information leakage while encouraging the model to establish long-range spatiotemporal relationships.

Motion-Aware Sampling: Regions with larger motion intensity intuitively contain richer semantic information about actions. Therefore, we utilize the spatial motion intensity of each human body joint within a tube as empirical guidance to generate the masking map.

Specifically, we first extract the corresponding motion sequence $M \in \mathbb{R}^{T_s \times V \times C_s}$ from the input skeleton sequence $I \in \mathbb{R}^{T_s \times V \times C_s}$ by calculating temporal differences of corresponding joint coordinates between adjacent frames:

$$M_{i,:,:} = \begin{cases} I_{i+1,:,:} - I_{i,:,:}, & i \in 0, \dots, T_s - 1 \\ 0, & i = T_s \end{cases} \tag{8}$$

Similar to joint embedding in the encoder, we reshape M into non-overlapping segments $M' \in \mathbb{R}^{T_e \times V \times (l \cdot C_s)}$ to match the shape of input sequence I'. We then calculate the motion intensity of each joint within a segment as:

$$S_{i,:} = \sum_{k=0}^{l \cdot C_s} |M'_{i,:,k}| \in \mathbb{R}^{T_e \times V}, \quad i = 0, \dots, T_e \tag{9}$$

Afterwards, we compute the spatial motion intensity of each body joint within a tube, normalizing it along the spatial dimension:

$$\begin{aligned} T_{i,:} &= \sum_{j=i}^{i+\alpha} S_{j,:} \in \mathbb{R}^{N \times V}, && i = 0, \dots, N \\ T'_{i,:} &= T_{i,:} / \max(T_{i,:}), && i = 0, \dots, N \end{aligned} \tag{10}$$

Finally, we utilize the normalized spatial motion intensity to generate a unique masking map for each tube:

$$\begin{aligned} p &= \eta + \beta \cdot T', && \eta \sim U(0,1) \\ \mathcal{M}_i &= \text{argsort}(p_{i,:})[-K :], && i = 0, \dots, N \end{aligned} \tag{11}$$

where η is random noise drawn from a uniform distribution between 0 and 1, β is a hyperparameter controlling the influence of spatial motion intensity on sampling, $\mathcal{M}_i$ is the masking map for i^{th} tube, $K = V \times (1 - m)$ is the number of joints to be masked, and m is the masking ratio. By customizing motion-aware masking maps for each tube, the model is encouraged to focus more on semantically richer regions, leading to improved spatiotemporal representations.

4 Experiments

4.1 Datasets

We evaluate our method on three large-scale 3D skeleton-based action recognition datasets: NTU RGB+D 60, NTU RGB+D 120, and PKU Multi-Modality Dataset (PKUMMD).

NTU RGB+D 60 [33] contains 56,880 skeleton sequences across 60 action categories performed by 40 subjects. We follow the recommended cross-subject and cross-view evaluation protocols. For cross-subject, sequences from 20 subjects are used for training and the rest are used for testing. For cross-view, training samples are from cameras 2 and 3, while testing samples are from camera 1.

NTU RGB+D 120 [24] is an extension of NTU RGB+D 60 with 114,480 skeleton sequences across 120 action categories performed by 106 subjects. The authors also propose a more challenging cross-setup evaluation protocol, where sequences are divided into 32 setups based on camera distance and background. Samples from 16 setups are used for training and the rest are used for testing.

PKUMMD [23] contains nearly 20,000 skeleton sequences across 52 action categories. We adopt the cross-subject protocol, where training and testing sets are split based on subject ID. PKUMMD consists of two parts: PKU-I and PKU-II. PKU-II is more challenging due to larger view variations that introduce more skeleton noise. For PKU-II, there are 5,332 sequences for training and 1,613 for testing.

4.2 Settings

Data Processing: We employed the data preprocessing method from DG-STGCN [11] to apply uniform sampling to a given skeleton sequence, generating subsequences as training samples. The number of frames T_s for sampling is set to 90. During the training, we applied random rotation as data augmentation on the sampled subsequences to enhance robustness against view variation. During the testing, we averaged the scores of 10 subsequences to predict the class.

Network Architecture: We adopted the same network architecture setting as MAMP [27], with the encoder layers L_e set to 8, decoder layers L_d set to 3, embedding dimension set to 256, the number of heads in the multi-head self-attention module set to 8, and the hidden dimension of the feed-forward network set to 1024. For Joint Embedding, the length l of each segment is set to 3.

Pre-training: In the pre-training, the initial value of the EMA parameter τ is set to 0.9999. The masking ratio m of the input sequence is set to 90%. The tube length α for motion-aware multi-tube masking is set to 5, and the sampling parameter β is set to 0.1. We utilized the AdamW optimizer with weight decay of 0.05 and betas (0.9, 0.95). The model was trained for a total of 600 epochs, with the learning rate linearly increasing to 1e-3 during the first 20 warmup epochs, and then decaying to 1e-5 according to a cosine decay schedule. Our model was trained on 2 RTX 4090 GPUs, with a total batch size of 128.

Table 1: Performance comparison in linear evaluation protocol on NTU 60, NTU 120, and PKU MMD datasets. *Single-stream* refers to Joint, while *Three-stream* denotes Joint+Motion+Bone.

Method	Input	NTU 60		NTU 120		PKU II
		XSub	XView	XSub	XSet	XSub
Other pretext tasks:						
LongTGAN [48]	Single-stream	39.1	48.1	-	-	26.0
P&C [34]	Single-stream	50.7	75.3	42.7	41.7	25.5
Contrastive Learning:						
CrosSCLR [20]	Three-stream	77.8	83.4	67.9	66.7	21.2
AimCLR [15]	Three-stream	78.9	83.8	68.2	68.8	39.5
CPM [46]	Single-stream	78.7	84.9	68.7	69.6	-
PSTL [49]	Three-stream	79.1	83.8	69.2	70.3	52.3
CMD [28]	Single-stream	79.4	86.9	70.3	71.5	-
HaLP [32]	Single-stream	79.7	86.8	71.1	72.2	43.5
HiCo-Transformer [7]	Single-stream	81.1	88.6	72.8	74.1	49.4
SkeAttnCLR [18]	Three-stream	82.0	86.5	77.1	80.0	55.5
ActCLR [22]	Three-stream	84.3	88.8	74.3	75.7	-
Masked Prediction:						
SkeletonMAE [43]	Single-stream	74.8	77.7	72.5	73.5	36.1
MAMP [27]	Single-stream	84.9	89.1	78.6	79.1	53.8
Skeleton2vec(Ours)	Single-stream	**85.7**	**90.3**	**79.7**	**81.3**	**55.6**

4.3 Evaluation and Comparison

Linear Evaluation: In this protocol, the parameters of the pre-trained encoder are fixed to extract features. A trainable linear classifier is then applied for classification. We train for 100 epochs in total using SGD optimizer with momentum of 0.9 and batch size of 256. Our results are evaluated on NTU-60, NTU-120, and PKU-MMD. Comparison with the latest methods reveals the superiority of our proposed Skeleton2vec, as illustrated in Tab. 1. Notably, in contrast to contrastive learning methods, Skeleton2vec, employing the masked prediction approach, demonstrates significant advantages. Furthermore, Skeleton2vec outperforms other masked prediction methods across all datasets. Particularly, on the NTU-60 XView and NTU-120 XSet datasets, Skeleton2vec exhibits superior performance over the previously state-of-the-art method MAMP by 1.2% and 2.2%, highlighting the strength of our contextualized prediction targets.

Semi-supervised Evaluation: In the semi-supervised protocol, we add an MLP head to the pre-trained encoder and then fine-tune the entire network using only 1% and 10% of the training data. We use the AdamW optimizer with a weight decay of 0.05. The learning rate starts at 0 and linearly increases to 3e-4 for the first 5 epochs, then decreases to 1e-5 according to a cosine decay schedule. We train the network for a total of 100 epochs with a batch size of 48. Evaluations on the NTU-60 dataset and comparisons with state-of-the-art approaches such as HYSP [13], SkeAttnCLR [18], and MAMP [27] are conducted. As depicted in Tab. 2, Skeleton2vec demonstrates significant superiority over these methods, particularly when utilizing only 1% of the training data. Specifically, on the XSub and XView settings, Skeleton2vec outperforms MAMP by 9.7% and 7.5%, respectively, affirming the superiority of the proposed Skeleton2vec pretraining framework.

Table 2: Performance comparison in the semi-supervised protocol on NTU 60 datasets. We averaged the results of five runs as the final performance.

Method	Input	NTU 60			
		XSub		XView	
		(1%)	(10%)	(1%)	(10%)
Other pretext tasks:					
LongTGAN [48]	Single-stream	35.2	62.0	-	-
MS2L [21]	Single-stream	33.1	65.1	-	-
Contrastive Learning:					
3s-CrosSCLR [20]	Three-stream	51.1	74.4	50.0	77.8
3s-Hi-TRS [5]	Three-stream	49.3	77.7	51.5	81.1
3s-AimCLR [15]	Three-stream	54.8	78.2	54.3	81.6
3s-CMD [28]	Three-stream	55.6	79.0	55.5	82.4
CPM [46]	Three-stream	56.7	73.0	57.5	77.1
3s-HYSP [13]	Three-stream	-	80.5	-	85.4
3s-SkeAttnCLR [18]	Three-stream	59.6	81.5	59.2	83.8
Masked Prediction:					
SkeletonMAE [40]	Single-stream	54.4	80.6	54.6	83.5
MAMP [27]	Single-stream	66.0	88.0	68.7	91.5
Skeleton2vec(Ours)	Single-stream	**75.7**	**89.2**	**76.2**	**92.9**

Fine-tuning Evaluation: Under the fine-tuning protocol, we employ the same training settings as the semi-supervised protocol, but fine-tuned using 100% of the training data. Evaluation of the fine-tuning results on the NTU-60 and NTU-120 datasets is presented in Tab. 3. Our proposed Skeleton2vec consistently outperforms previous methods based on the masked prediction task, including SkeletonMAE [43] and MotionBERT [50], across all datasets. Additionally, our approach demonstrated comparable or even superior performance compared to the state-of-the-art method MAMP [27] on most datasets, particularly on the NTU-60 XView dataset. However, the advantage of our method under the fine-tuning protocol seems less pronounced compared to the semi-supervised protocol. This is primarily because the model achieves a good fit when fine-tuned using a sufficient amount of training data (100%), thus mitigating the performance differences introduced by various pre-training methods. This phenomenon is also evident in the results of the semi-supervised protocol (Tab. 2), where the performance improvement from using 10% of the training data is much smaller than that from using 1% of the training data.

Transfer Learning Evaluation: In the transfer learning evaluation protocol, pretraining is initially performed on the source dataset and subsequently fine-tuned on the target dataset. The source datasets used in our experiments are NTU-60 and NTU-120, with the target dataset being PKU-MMD II. As illustrated in Tab. 4, our proposed Skeleton2vec surpasses the state-of-the-art method MAMP by 2.4% and 1.9% when using NTU-60 and NTU-120 as source datasets, respectively. This underscores the robustness of features learned through the Skeleton2vec framework.

4.4 Ablation Study

We conducted an extensive ablation study on NTU-60 XSub dataset and NTU-60 XView dataset to analyze the proposed Skeleton2vec framework. Unless other-

Table 3: Performance comparison in fine-tuning protocol on NTU 60 and NTU 120 datasets. The best results are shown in bold, and the second-best results are highlighted with an underline.

Method	Input	Backbone	NTU 60		NTU 120	
			XSub	XView	XSub	XSet
Other pretext tasks:						
Colorization [45]	Three-stream	DGCNN	88.0	94.9	-	-
Hi-TRS [5]	Three-stream	Transformer	90.0	95.7	85.3	87.4
Contrastive Learning:						
CPM [46]	Single-stream	ST-GCN	84.8	91.1	78.4	78.9
CrosSCLR [20]	Three-stream	ST-GCN	86.2	92.5	80.5	80.4
AimCLR [15]	Three-stream	ST-GCN	86.9	92.8	80.1	80.9
ActCLR [22]	Three-stream	ST-GCN	88.2	93.9	82.1	84.6
HYSP [13]	Three-stream	ST-GCN	89.1	95.2	84.5	86.3
Masked Prediction:						
SkeletonMAE [40]	Single-stream	STTFormer	86.6	92.9	76.8	79.1
SkeletonMAE [43]	Single-stream	STRL	92.8	96.5	84.8	85.7
MotionBERT [50]	Single-stream	DSTformer	93.0	97.2	-	-
MAMP [27]	Single-stream	Transformer	93.1	97.5	**90.0**	**91.3**
Skeleton2vec(Ours)	Single-stream	Transformer	**93.2**	**97.8**	89.8	**91.3**

wise specified, we pre-train the model for 200 epochs and report the results under the linear evaluation protocol.

Teacher vs. Student: Compared to offline pre-training a teacher network using additional labeled data or pretext tasks to guide a student, our Skeleton2vec employs online updating of the teacher's parameters through EMA of the student's parameters. This approach does not require labeled data or additional training stages, making it a highly cost-effective choice. To demonstrate the teacher's ability to guide the student through online EMA updates, we illustrate in Fig. 4 the performance evolution of both teacher and student on the NTU-60 XSub dataset throughout the training process. It is evident that the teacher consistently outperforms the student, indicating that the teacher has learned high-level semantics and can effectively guide the student.

Teacher Weight Update: We regulate the update frequency of teacher's weights by adjusting the parameter τ_0 in the exponential moving average. In Fig. 5, we compared the impact of four different values of τ_0 on the pre-training performance of the model. It is observed that employing smaller τ_0 values (0.99, 0.999) leads to a rapid performance improvement in the early stages of training (first 100 epochs). However, as training progresses, the performance growth diminishes, and in some cases, a decline is observed. Conversely, overly large values of τ_0 (0.99999) significantly slow down the convergence of training, incurring impractical time costs. Through experimentation, we found that using an appropriate τ_0 value (0.9999) achieves a balanced convergence speed and growth potential, resulting in optimal performance.

Masking Strategy: Tab. 5a presents a performance comparison between our proposed Motion-Aware Multi-Tube (MAMT) masking strategy and other methods. It is evident that compared to random masking and single-tube masking, multi-tube masking yields a significant performance improvement, validating the effectiveness of dividing into multiple tubes. Moreover, motion-aware sampling

Table 4: Performance comparison in the transfer learning protocol. The source datasets are NTU-60 and NTU-120, and the target dataset is PKU-II.

Method	To PKU-II	
	NTU 60	NTU 120
ISC [35]	51.1	52.3
CMD [28]	56.0	57.0
HaLP+CMD [32]	56.6	57.3
SkeletonMAE [40]	58.4	61.0
MAMP [27]	70.6	73.2
Skeleton2vec(Ours)	**73.0**	**75.1**

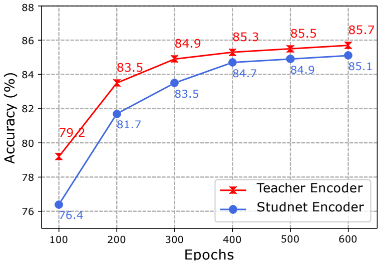


Fig. 4: The performance comparison between teacher and student on the NTU-60 XSub dataset across the training process.

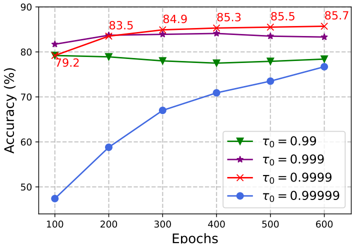


Fig. 5: Ablation study of EMA parameter τ_0 on the NTU-60 XSub dataset under linear protocol.

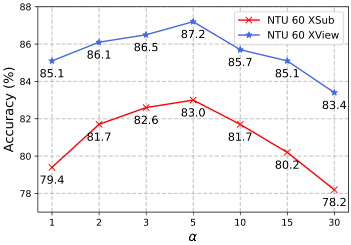


Fig. 6: Ablation study of the tube length α on the NTU-60 XSub and XView datasets.

further improves performance, highlighting the value of guiding the model to focus on semantically rich action regions. A detailed analysis of hyperparameters in MAMT masking will be presented in subsequent sections.

Tube Length: We investigated the impact of the length α of each tube on pre-training performance. As depicted in Fig. 6, excessively short tube lengths result in information leakage between adjacent frames, leading to a performance decline. On the other hand, overly long tube lengths pose excessively challenging pre-training tasks, impairing the model's learning capacity, as discussed in Sec. 3.4. Hence, selecting an appropriate tube length is crucial. Considering the results from Fig. 6, we identified a tube length of $\alpha = 5$ as optimal, achieving the best balance and performance.

Motion-aware Sampling: We compared the performance of learned representations under different motion-aware sampling parameters β. As shown in Tab. 5b, selecting an appropriate sampling parameter enhances pre-training performance compared to not using motion prior information ($\beta = 0$). However, excessively large sampling parameters can result in overly fixed sampling of joints, leading to a loss of diversity and a subsequent performance decline. We empirically found that a sampling parameter of $\beta = 0.1$ yields the best results.

Masking Ratio: In Tab. 5c, we compared the influence of different masking ratios on the results. It is evident that excessively large or small masking ratios

Table 5: Ablation study on the **Motion-Aware Multi-Tube (MAMT)** masking strategy. α represents the length of each tube, while β denotes the parameter of motion-aware sampling. m denotes the masking ratio.

Strategy	α	β	NTU 60	
			XSub	XView
Random	1	0.0	79.4	85.1
Single-tube	30	0.0	78.2	83.4
Multi-tube	5	0.0	83.0	87.2
MAMT(Ours)	5	0.1	**83.5**	**87.7**

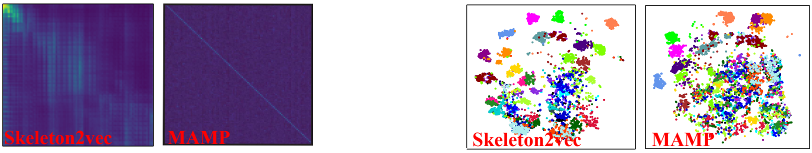
(a) Masking strategy.

β	NTU 60	
	XSub	XView
0.0	83.0	87.2
0.1	**83.5**	**87.7**
0.2	82.1	87.0
0.3	79.5	86.3

(b) Motion-aware sampling

m	NTU 60	
	XSub	XView
0.80	83.1	86.7
0.85	83.3	87.3
0.90	**83.5**	**87.7**
0.95	77.1	82.1

(c) Masking ratio



(a) Average Multi-Head Self-Attention Matrices **(b)** t-SNE Visualization of Feature Embeddings

Fig. 7: Qualitative Comparison between Skeleton2vec and MAMP

can impair the final performance. We ultimately selected a masking ratio of 90% to achieve optimal results.

Qualitative Results: To illustrate the effectiveness of contextualized representations as self-supervised targets in Skeleton2vec, we compared the average multi-head self-attention matrices and output feature embeddings of pre-trained encoders between Skeleton2vec and MAMP [27]. The visualization results are depicted in Fig. 7, where the feature embeddings are derived from samples of 30 classes selected from the NTU-60 XSub test set and the visualization is performed using t-SNE [26]. Compared to MAMP, Skeleton2vec shows a more uniform and global attention distribution, thanks to its use of globally contextualized representations as prediction targets rather than local motion context alone. Furthermore, Skeleton2vec's feature outputs exhibit significantly improved separability, confirming the efficacy of our approach.

5 Conclusion

In this work, we propose Skeleton2vec, a novel self-supervised learning framework for 3D skeleton-based action recognition. We demonstrated the superiority of utilizing global contextualized representations built by a teacher model as the prediction target for the masked prediction task, compared to isolated raw joints or temporal motion with local context. Furthermore, considering the high spatiotemporal correlation in skeleton sequences, we proposed the motion-aware multi-tube masking strategy to compel the model into effective long-range motion modeling. Extensive experiments conducted on three large-scale prevalent benchmarks validated the effectiveness of our approach. The experimental results showcased outstanding performance of our proposed Skeleton2vec, achieving state-of-the-art results across multiple testing protocols.

References

1. Baevski, A., Babu, A., Hsu, W.N., Auli, M.: Efficient self-supervised learning with contextualized target representations for vision, speech and language. In: International Conference on Machine Learning. pp. 1416–1429. PMLR (2023) 2, 4, 6
2. Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M.: Data2vec: A general framework for self-supervised learning in speech, vision and language. In: International Conference on Machine Learning. pp. 1298–1312. PMLR (2022) 2, 4
3. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021) 4
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE TPAMI (2018) 1
5. Chen, Y., Zhao, L., Yuan, J., Tian, Y., Xia, Z., Geng, S., Han, L., Metaxas, D.N.: Hierarchically self-supervised transformer for human skeleton representation learning. In: ECCV. pp. 185–202. Springer (2022) 11, 12
6. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: ICCV. pp. 13359–13368 (2021) 1
7. Dong, J., Sun, S., Liu, Z., Chen, S., Liu, B., Wang, X.: Hierarchical contrast for unsupervised skeleton-based action representation learning. In: AAAI (2023) 10
8. Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., Yu, N., Guo, B.: Peco: Perceptual codebook for bert pre-training of vision transformers. In: AAAI. vol. 37, pp. 552–560 (2023) 4
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 4
10. Du, Y., Fu, Y., Wang, L.: Skeleton based action recognition with convolutional neural network. In: 2015 3rd IAPR Asian conference on pattern recognition (ACPR). pp. 579–583. IEEE (2015) 1
11. Duan, H., Wang, J., Chen, K., Lin, D.: Dg-stgcn: dynamic spatial-temporal modeling for skeleton-based action recognition. arXiv preprint arXiv:2210.05895 (2022) 9
12. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: ICCV. pp. 2334–2343 (2017) 1
13. Franco, L., Mandica, P., Munjal, B., Galasso, F.: Hyperbolic self-paced learning for self-supervised skeleton-based action representations. In: Int. Conf. Learn. Represent. (2023), https://openreview.net/forum?id=3Bh6sRPKS3J 10, 11, 12
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS **33**, 21271–21284 (2020) 3
15. Guo, T., Liu, H., Chen, Z., Liu, M., Wang, T., Ding, R.: Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In: AAAI. vol. 36, pp. 762–770 (2022) 2, 3, 10, 11, 12
16. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022) 2, 4
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020) 3

18. Hua, Y., Wu, W., Zheng, C., Lu, A., Liu, M., Chen, C., Wu, S.: Part aware contrastive learning for self-supervised action recognition. In: Int. J. Comput. Vis. (2023) 10, 11
19. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: 2017 IEEE international conference on multimedia & expo workshops (ICMEW). pp. 597–600. IEEE (2017) 1
20. Li, L., Wang, M., Ni, B., Wang, H., Yang, J., Zhang, W.: 3d human action representation learning via cross-view consistency pursuit. In: CVPR. pp. 4741–4750 (2021) 3, 10, 11, 12
21. Lin, L., Song, S., Yang, W., Liu, J.: Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In: ACM MM. pp. 2490–2498 (2020) 2, 3, 11
22. Lin, L., Zhang, J., Liu, J.: Actionlet-dependent contrastive learning for unsupervised skeleton-based action recognition. In: CVPR. pp. 2363–2372 (2023) 2, 3, 10, 12
23. Liu, C., Hu, Y., Li, Y., Song, S., Liu, J.: Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. arXiv preprint arXiv:1703.07475 (2017) 9
24. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. IEEE Trans. Pattern Anal. Mach. Intell. **42**(10), 2684–2701 (2019) 9
25. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 816–833. Springer (2016) 1
26. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 14
27. Mao, Y., Deng, J., Zhou, W., Fang, Y., Ouyang, W., Li, H.: Masked motion predictors are strong 3d action representation learners. In: ICCV. pp. 10181–10191 (2023) 2, 4, 5, 9, 10, 11, 12, 13, 14
28. Mao, Y., Zhou, W., Lu, Z., Deng, J., Li, H.: Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. In: ECCV. pp. 734–752. Springer (2022) 10, 11, 13
29. Moliner, O., Huang, S., Åström, K.: Bootstrapped representation learning for skeleton-based action recognition. In: CVPR. pp. 4154–4164 (2022) 2, 3
30. Nie, Q., Liu, Z., Liu, Y.: Unsupervised 3d human pose representation with viewpoint and pose disentanglement. In: ECCV. pp. 102–118. Springer (2020) 2
31. Rao, H., Xu, S., Hu, X., Cheng, J., Hu, B.: Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. Information Sciences **569**, 90–109 (2021) 2, 3
32. Shah, A., Roy, A., Shah, K., Mishra, S., Jacobs, D., Cherian, A., Chellappa, R.: Halp: Hallucinating latent positives for skeleton-based self-supervised learning of actions. In: CVPR. pp. 18846–18856 (2023) 10, 13
33. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: CVPR. pp. 1010–1019 (2016) 9
34. Su, K., Liu, X., Shlizerman, E.: Predict & cluster: Unsupervised skeleton based action recognition. In: CVPR. pp. 9631–9640 (2020) 2, 3, 10
35. Thoker, F.M., Doughty, H., Snoek, C.G.: Skeleton-contrastive 3d action representation learning. In: ACM MM. pp. 1655–1663 (2021) 13
36. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS **35**, 10078–10093 (2022) 7

37. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016) 6
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017) 2
39. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. In: CVPR. pp. 14668–14678 (2022) 4
40. Wu, W., Hua, Y., Zheng, C., Wu, S., Chen, C., Lu, A.: Skeletonmae: Spatial-temporal masked autoencoders for self-supervised skeleton action recognition. In: 2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW). pp. 224–229. IEEE (2023) 2, 4, 11, 12, 13
41. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: CVPR. pp. 9653–9663 (2022) 4
42. Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., Zhang, W.: Deep kinematics analysis for monocular 3d human pose estimation. In: CVPR. pp. 899–908 (2020) 1
43. Yan, H., Liu, Y., Wei, Y., Li, Z., Li, G., Lin, L.: Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training. In: ICCV. pp. 5606–5618 (2023) 2, 10, 11, 12
44. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI. vol. 32 (2018) 1
45. Yang, S., Liu, J., Lu, S., Er, M.H., Kot, A.C.: Skeleton cloud colorization for unsupervised 3d action representation learning. In: ICCV. pp. 13423–13433 (2021) 3, 12
46. Zhang, H., Hou, Y., Zhang, W., Li, W.: Contrastive positive mining for unsupervised 3d action representation learning. In: ECCV. pp. 36–51. Springer (2022) 10, 11, 12
47. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE international conference on computer vision. pp. 2117–2126 (2017) 1
48. Zheng, N., Wen, J., Liu, R., Long, L., Dai, J., Gong, Z.: Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In: AAAI. vol. 32 (2018) 2, 3, 10, 11
49. Zhou, Y., Duan, H., Rao, A., Su, B., Wang, J.: Self-supervised action representation learning from partial spatio-temporal skeleton sequences. In: AAAI (2023), https://api.semanticscholar.org/CorpusID:257019654 10
50. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: A unified perspective on learning human motion representations. In: ICCV. pp. 15085–15099 (2023) 11, 12