S-JEPA: A Joint Embedding Predictive Architecture for Skeletal Action Recognition

Mohamed Abdelfattah* and Alexandre Alahi

EPFL, Lausanne 1015, Switzerland
firstname.lastname@epfl.ch

Abstract. Masked self-reconstruction of joints has been shown to be a promising pretext task for self-supervised skeletal action recognition. However, this task focuses on predicting isolated, potentially noisy, joint coordinates, which results in an inefficient utilization of the model capacity. In this paper, we introduce **S-JEPA**, a **S**keleton **J**oint **E**mbedding **P**redictive **A**rchitecture, which uses a novel pretext task: Given a partial skeleton sequence, predict the latent representations of the missing joints of the same sequence. Such representations serve as abstract prediction targets that direct the modelling power towards learning the high-level context and depth information, instead of unnecessary low-level details. To tackle the potential non-uniformity in these representations, we propose a simple centering operation that is found to benefit training stability, effectively leading to strong off-the-shelf action representations. Extensive experiments show that S-JEPA, combined with the vanilla transformer, outperforms previous state-of-the-art results on NTU60, NTU120, and PKU-MMD datasets. Project website: https://sjepa.github.io.

Keywords: Skeleton-based action recognition · Self-supervised learning · Representation learning

1 Introduction

Skeletons serve as powerful abstract representations of human actions, while being more compact, computationally efficient, privacy preserving, and robust to background noise, compared to RGB videos. However, existing fully-supervised skeletal action recognition methods [9, 13, 24, 25, 30, 36, 40–42, 47, 53] require an extensive amount of time-consuming and labor-intensive data annotations. In addition, limited supervision could lead to overfitting to the training data and hurting the model generalizability, especially with models with weak inductive bias and large capacity such as transformers [51]. These observations motivate the need for self-supervised skeletal action representation learning.

Self-supervised methods based on Contrastive Learning (CL) have shown promising results in learning action representations. However, CL methods [1, 16, 27, 34, 35] tend to focus on view-invariance as their pretraining objective, and

* Corresponding author.

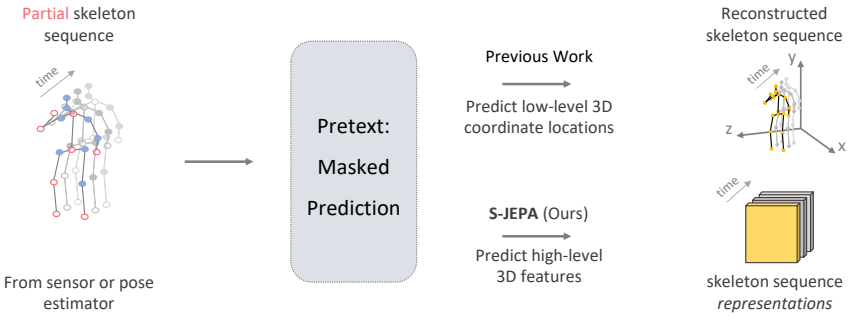


Fig. 1: Comparison between the prediction targets of previous work and S-JEPA (ours). Instead of raw 3D coordinates, S-JEPA predicts the abstract representations of 3D skeletons, embedded by a transformer encoder, effectively learning more informative high-level depth and context features for the action recognition task.

therefore, heavily rely on hand-crafted augmentations. With the introduction of Masked Autoencoders (MAEs) [17], recent methods have attempted to employ masked self-reconstruction as a pretext (pretraining) task. In these methods, the input skeleton sequence is typically masked or corrupted in some way, and the model is trained to reconstruct the original, uncorrupted sequence. By doing so, the autoencoder learns to capture important features and learn a compact and informative representation of the skeletons, as it needs to infer the prediction targets such as masked joints coordinates [46, 49] or temporal motion [32]. Another work [57] have proposed 2D-to-3D lifting as a pretext objective for action recognition: from partial 2D skeletons, predict the raw 3D joint coordinates of the same skeletons. In this way, the network learns to extract informative depth features from the input 2D skeleton data to perform the lifting task effectively. The type of prediction targets significantly influences the kind of information stored in the compact representation learned by the model. Different prediction targets require the model to focus on different aspects of the input data, leading to the extraction of distinct features and representations.

However, the prediction targets of these methods: (1) *Contain unnecessary details:* Low-level high-frequency joint coordinates contain unnecessary details from erroneous or noisy joints. This in turn results in an inefficient utilization of the model capacity by potentially modelling such discrepancies. (2) *Lack context information:* Raw joint coordinates are isolated targets that lack rich context information from neighbouring joints across the spatio-temporal dimensions. This hinders the model capacity to learn the rich joint-wise correlations that characterize every action. Motivated by these observations, we ask: *What pretext task can better optimize the model capacity for learning informative skeleton representations?*

In this work, we provide an answer to this question by proposing a novel pretext task: Given a partial skeleton sequence, predict the *latent representations* of the missing joints of the same skeleton sequence, where these representations

are embedded by a learned encoder network. We learn to match the probability distributions of the predicted representations with those of the encoded representations using standard cross-entropy loss. Interestingly, we find that applying centering and softmax operations on the target representations before the loss function results in a more stable training loss and improved performance. We term our approach **S-JEPA**, **S**keleton **J**oint **E**mbedding **P**redictive **A**rchitecture. Unlike previous methods that directly predict low-level, high-frequency 3D joint coordinates or motion, S-JEPA predicts their high-level abstract representations, as depicted in Figure 1. The vanilla transformer [12] is adopted as our backbone.

S-JEPA directly addresses the aforementioned limitations of previous methods. By predicting the latent representations, the transformer modelling power is better directed towards learning the informative *high-level* features, successfully avoiding the overhead of learning the raw, potentially noisy, joint coordinates. Further, by leveraging *contextualized* representations encoded by a transformer backbone as prediction targets, the model can learn to encode complex temporal patterns and dependencies in skeletal action sequences. As a results, S-JEPA learns strong off-the-shelf representations that outperform MAE-like methods in linear, fine-tuning, semi-supervised, and transfer learning evaluations.

2 Background

Self-supervised learning is a paradigm in which a model learns to understand and represent the underlying structure of input data without explicit supervision from human labels. Instead, the model generates its own labels or supervisory signals from the input data itself, and the pretext objective is to learn to predict these labels given the input data. In the context of skeleton-based action recognition, existing self-supervised methods can be classified based on their pretext tasks into three categories: Generative Architectures (GA), Joint-Embedding Architectures (JEA), and Joint-Embedding Predictive Architectures (JEPA).

Generative Architectures. The core idea behind generative architectures is to learn to reconstruct a property of the skeleton sequence (*e.g.,* joint coordinates, motion) given a corrupted view of the same sequence. To that end, an encoder is conventionally used to embed the corrupted view of the skeletons into latent representations, which are then fed to a decoder or an MLP to generate the skeleton targets. Several recent generative architectures have demonstrated encouraging results. SkeletonMAE [46, 49] adopts a transformer-based Masked Autoencoder (MAE) to reconstruct masked joint locations. MAMP [32] learns to predict the explicit temporal motion of masked joints. MotionBERT [57] employs 2D-to-3D lifting by using spatial and temporal transformer blocks to capture the spatial and temporal information, respectively. LongTGAN [54] employs adversarial training to reconstruct randomly masked joints. P&C [43] utilizes an encoder-decoder network to predict future time steps of a skeleton sequence. A common design feature for GAs is that the prediction happens in the *joint space*.

Joint-Embedding Architectures. These methods focus on learning invariance based representations by enforcing similar embeddings for compatible skeletons and dissimilar embeddings for incompatible ones. The compatible skeletons are typically obtained by applying random data augmentations on the same sequence [6]. Numerous JAEs are based on contrastive learning for learning strong 3D action representations [15, 16, 18, 27, 34, 35]. CrossCLR [23] proposes using a momentum encoder with various data augmentations. AimCLR [16] uses extreme augmentations to sample positive pairs. ActCLR [27] utilizes motion-adaptive augmentations to avoid corrupting the action content. SkeAttnCLR [19] integrates local similarity and global features for skeleton-based action representations. The common pretext objective behind JEAs is to achieve skeleton representations that are invariant to the applied augmentations.

Joint-Embedding Predictive Architectures. Similar to generative architectures, the pretext objective of JEPAs is to reconstruct an aspect of the input data. However, unlike GAs, JEPAs learn to predict the *representations* of the target aspect and, therefore, the loss function is applied in the latent space, not in the input space [22]. JEPAs are also different from JEAs since the goal of JEPAs is not to enforce similar representations of compatible inputs. Instead, the pretext objective is to learn representations that are predictive of each other [2]. Hence, a key design choice in JEPAs is the usage of a predictor that learns to predict the representations of input regions, given the representations of other regions from the input data. In this work, we introduce S-JEPA, which is an instantiation of this architecture for self-supervised skeletal action recognition, as illustrated in Figure 2.

3 Method

Overview. Our Skeleton Joint Embedding Predictive Architecture (S-JEPA) consists of two stages: pretraining and fine-tuning. In the pretraining stage, the model learns informative skeleton representations without human labels by optimizing for a simple pretext task: Given a masked skeleton sequence, predict the latent representations of the missing joints of the same sequence. The missing joints representations (prediction targets) are obtained by embedding the 3D skeletons by a target encoder, as illustrated in Figure 2. The backbones are all vanilla transformer [12] networks. In the fine-tuning stage, the pre-trained model's weights are adjusted using labeled data to adapt it to action recognition. In the following, we discuss the details of each component in our framework.

Targets. The target encoder serves as a source of guidance for the model during training. It generates target representations that the predictor aims to match during the training process. Given an input 3D skeleton sequence $\mathbf{y} \in \mathbb{R}^{T \times V \times C_{in}}$, where T is the number of frames, V is the number of joints, and C_{in} is the number of coordinate axes, we first convert it into temporally non-overlapping

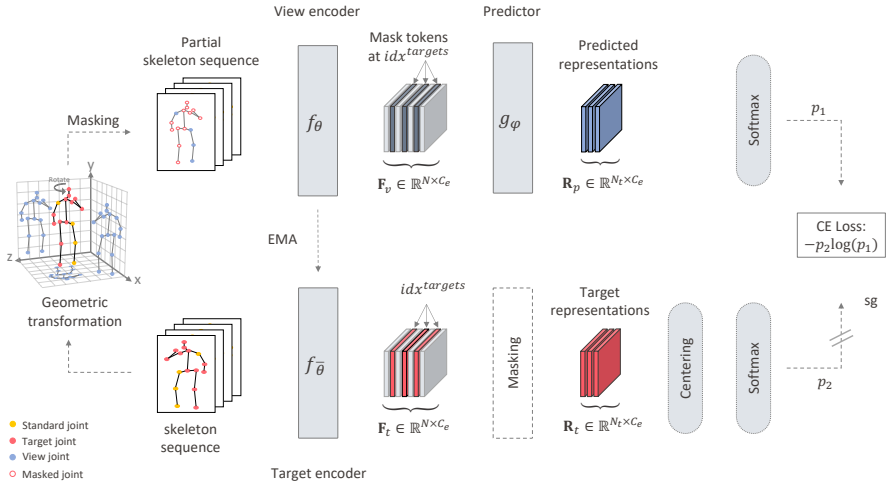


Fig. 2: Overview of S-JEPA. First, diverse skeleton views are obtained by applying geometric transformations on the 3D skeletons. The view skeletons are passed through the view encoder, after which learnable mask tokens are inserted at the locations of masked joints to get the view features $\mathbf{F}_v$. The predictor takes $\mathbf{F}_v$ as input and outputs the predicted representations $\mathbf{R}_p$ of the missing joints at the locations of the mask tokens. The target representations $\mathbf{R}_t$ are obtained by the target encoder, which takes unmasked 3D skeletons as input, and is updated through the Exponential Moving Average (EMA) of the view encoder weights after each iteration (sg denotes stop gradient). The centering and softmax operations aid in stabilizing the training loss. At fine-tuning and test times, only the target encoder weights are used.

segments $I \in \mathbb{R}^{T_e\times V\times l\times C_{in}}$, where $T_e = T/l$ and l is the segment length. For computational efficiency, joints of the same spatial position within each segment are embedded together through linear transformation, yielding a sequence of N patch embeddings $E_t \in \mathbb{R}^{N\times C_e}$ of C_e channels, $N = T_e \times V$. We feed this through the target encoder $f_{\bar{\theta}}$ with L_e Multi-Head Self-Attention (MHSA) layers to obtain the target encoder output features $\mathbf{F}_t \in \mathbb{R}^{N\times C_e}$. We leverage the motion-aware masking strategy proposed in [32] with mask ratio $r \in [0, 1]$ to sample the features corresponding to joints with significant motion with a higher probability (target joints). Hence, we obtain the indices $idx^{targets}$ of semantically rich features and use them to get the target representations $\mathbf{R}_t \in \mathbb{R}^{N_t\times C_e}$, where $N_t = N \times r$. The transformer backbone uses the self-attention mechanism which allows each joint in the skeleton sequence to attend to every other joint. This helps the model capture the overall structure and dynamics of the action, forming richer prediction targets than isolated joint positions. Therefore, it is crucial to obtain the targets $\mathbf{R}_t$ by masking the output of the target encoder, and not the input, to capture the global context information.

Views. Diversifying the views of the skeletons fed to the view encoder can significantly enhance the quality of the representations produced by our framework. The view encoder has to process different occlusions, perspectives, and orientations, resulting in representations that capture more nuanced aspects of the skeletal actions. To that end, we first aim to achieve diverse skeleton views by randomly rotating the 3D skeletons using a rotation function ϕ defined as:

$$\phi(\mathbf{p}, \alpha, \mathbf{n}) = \begin{bmatrix} R_{11}(\mathbf{n}, \alpha) & R_{12}(\mathbf{n}, \alpha) & R_{13}(\mathbf{n}, \alpha) \\ R_{21}(\mathbf{n}, \alpha) & R_{22}(\mathbf{n}, \alpha) & R_{23}(\mathbf{n}, \alpha) \\ R_{31}(\mathbf{n}, \alpha) & R_{32}(\mathbf{n}, \alpha) & R_{33}(\mathbf{n}, \alpha) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{1}$$

where $\mathbf{p} = [x, y, z]^t$ is the given keypoint coordinates from the 3D skeleton sequence, α is the rotation angle, $\mathbf{n}$ represents the axis around which we rotate, and $R(\mathbf{n}, \alpha) \in \mathbb{SO}^3$ is a rotation matrix derived from Rodrigues' rotation formula [37]. Here, we replace $\mathbf{n}$ with the vertical axis of the skeleton, estimated from the locations of pelvis and shoulders. We couple this with affine transformations such as linear translation and spatial flipping to diversify the views without corrupting the action semantics. Then, we patchify the resulting skeletons with linear transformation and drop the joint embeddings at $idx^{targets}$ to obtain the view embeddings $E_v \in \mathbb{R}^{N_v \times C_e}$, of N_v patches and C_e channels, where $N_v = N - N_t$. Finally, we pass E_v, which corresponds to the visible joints, to the view encoder f_θ, and insert learnable mask tokens in its output at the locations of the missing joints $idx^{targets}$ to get the view features $\mathbf{F}_v \in \mathbb{R}^{N \times C_e}$.

Predictions. Having the view features $\mathbf{F}_v$, we now wish to predict the probability distribution of the target representations $\mathbf{R}_t$. To that end, the predictor g_φ first applies linear transformation on $\mathbf{F}_v$ to get the embeddings $E_p \in \mathbb{R}^{N \times C_p}$ of C_p channels. Then E_p is passed through L_p MHSA layers to get the predicted representations $\mathbf{R}_p \in \mathbb{R}^{N_t \times C_e}$. The predictions are therefore, conditioned on the mask tokens inserted at the locations $idx^{targets}$ that we wish to predict.

Loss. We apply the softmax function on $\mathbf{R}_p$ and $\mathbf{R}_t$ with temperature hyperparameters τ_p and τ_t to get the probability distributions p_1 and p_2 for the predicted and target representations, respectively. Hence, the pretext objective is to match these distributions by minimizing the Cross-Entropy (CE) loss:

$$\mathcal{L}_{ce} = -p_2 \, log(p_1). \tag{2}$$

The parameters of the view encoder θ and predictor ϑ are updated through gradient-based optimization. The target encoder parameters $\overline{\theta}$ are updated with EMA of the view encoder weights with update rule $\overline{\theta} \leftarrow \lambda\overline{\theta} + (1 - \lambda)\theta$, where λ is the update rate hyperparameter. We find the momentum target encoder to be essential in training our framework to avoid representation collapse [21]. Previous work in other tasks such as [2, 5, 7, 55] report the same finding.

Stabilizing training. We empirically find that directly applying a similarity loss between $\mathbf{R}_p$ and $\mathbf{R}_t$ results in unstable training loss. This is due to the non-uniformity of the distribution of $\mathbf{R}_t$ in which one dimension could dominate the others. While this could be alleviated by using multiple normalizations [4, 48], we find that using only centering and sharpening [5] of $\mathbf{R}_t$ stabilizes our framework. Centering smoothens the distribution, prevents one dimension from dominating, and encourages the dimensions to be more uniform. We achieve this by subtracting the batch center c from $\mathbf{R}_t$ as follows:

$$\begin{aligned} c &\leftarrow \beta c + (1-\beta)\frac{1}{B}\sum_{i=1}^{B}\mathbf{R}_t^i, \\ \mathbf{R}_t &\leftarrow \mathbf{R}_t - c, \end{aligned} \tag{3}$$

where β is a rate parameter B is the batch size and $\mathbf{R}_t^i$ is the target representations of input sequence i. The primary purpose of centering is to align the target representations $\mathbf{R}_t$ with a stable and consistent reference. By centering $\mathbf{R}_t$, S-JEPA ensures that the predictor learns to match its representations with a reliable reference that is free from biases and variations. Centering the predictor output as well is not necessary since it could disrupt this alignment and introduce additional complexity by having overly constrained representations, making it more challenging for the predictor to effectively learn from the target encoder's guidance. In contrast, sharpening makes the distribution more peaked around the highest-scoring dimension, and is achieved by using a low value for the temperature hyperparameter τ_t in the softmax function applied on $\mathbf{R}_t$. Applying both operations balances their effect, resulting in a stable training loss.

4 Experiments

4.1 Datasets

To verify the effectiveness of S-JEPA, we evaluate it on three popular action recognition benchmarks: NTU-RGB+D 60 (NTU60) [39], NTU-RGB+D 120 (NTU120) [29], and PKU-MMD [28].

NTU60 [39]. NTU60 is large-scale dataset containing 60 action classes performed by 40 human subjects. We utilize the 57K ground truth 3D skeleton sequences provided in this dataset, and we adopt the cross-subject (XSub) and cross-view (XView) evaluation protocols. In the XSub protocol, skeleton sequences of 20 subjects are used for training and the rest are used for testing. In the XView protocol, action samples captured by 2 cameras are used for training, while the ones captured by a third camera are used for testing.

NTU120 [29]. An extension of NTU60, NTU120 has 114K samples, divided into 120 action classes, by 106 human subjects. In addition, it has a third evaluation protocol, cross-setup (XSet), in which the authors divide the samples into 32 different setups according to the background and camera distance; samples belonging to 16 setups are used for training while the rest are used for testing.

PKU-MMD [28]. PKU-MMD is based on long continuous and complex actions. We utilize the temporal annotations to trim action instances, and further divide them according to the subject IDs into training and testing sets, following [32, 44]. PKU-MMD has two phases, in which first has 23K samples categorized into 51 classes, and the second has 7K samples (41 action classes) with more noise from the larger view variation.

4.2 Implementation Details

Transformer Backbone. We leverage the vanilla transformer network, proposed in [12], as the backbone network for both the encoders and predictor in our framework. The encoder is composed of $L_e = 8$ layers while the predictor has $L_p = 5$ layers. We set the embedding dimension $C_e = C_p = 256$, the number of heads in each of the Multi-Head Self-Attention (MHSA) blocks to 8, and the feed-forward network hidden dimension to 1024 in both the encoders and predictor. The input to each transformer encoder is the high-dimensional features of each joint across the spatiotemporal dimensions. Separate learnable spatial and temporal positional embeddings are added to the embedded inputs before the first layer. The MHSA blocks output the joint-wise representations by looking at those of other joints through the self-attention mechanism [45].

Processing Skeletons. We first randomly trim a continuous segment from the input skeleton sequence with a random ratio between [0.5, 1] during training and 0.9 during testing. Next, we use bilinear interpolation to resize the trimmed segment to a fixed length $T = 120$. During pretraining, we randomly rotate each skeleton around its own axis and apply affine transformations to get diverse views, as described in Section 3. Following MAMP [32], we set the motion-aware masking ratio r to 0.9 and segment length l to 4.

Pre-training stage. We pretrain our network for 1200 epochs with effective batch size of 256. In the first 20 warm-up epochs, the learning rate is gradually increased from 0 to 1e-3 and then decreased to 5e-4 throughout the rest of the training following the cosine decay role. AdamW [31] optimizer is employed to update the parameters of the view encoder and the predictor with weight decay 0.05 and betas of (0.9, 0.95). The update rate parameter λ of target encoder weights is gradually increased from 0.9999 to 1.0 throughout the training following a cosine schedule. For the centering operation, we set the rate parameter β to 0.9. We then sharpen the predicted and target representations with softmax temperature parameters $\tau_p = 0.1$ and $\tau_t = 0.06$, respectively, before applying the cross-entropy loss. Eight NVIDIA A100 GPUs are utilized to train our model.

4.3 Comparison with State-of-the-art Methods

Linear Evaluation. Table 1 shows the performance comparison under the linear evaluation protocol. The model weights are frozen, and a linear classifier is trained on top with supervision for 300 epochs with a batch size of 256 and an SGD optimizer with a momentum of 0.9. The learning rate start at 0.1 and gradually decays to 0 with the cosine decay rule. As shown, we compare

Table 1: Accuracy comparison under the linear evaluation protocol. J indicates using joints only as input and J+B+M indicates using joint, bone, and motion.

Method	Input	NTU60 [39]		NTU120 [29]		PKU-MMD [28]	
		XSub	XView	XSub	XSet	Phase I	Phase II
Other pretext task:							
LongTGAN [54]	J	39.1	48.1	-	-	67.7	26.0
P&C [43]	J	50.7	75.3	42.7	41.7	59.9	25.5
Contrastive learning:							
CrosSCLR [23]	J+B+M	77.8	83.4	67.9	66.7	84.9	21.2
AimCLR [16]	J+B+M	78.9	83.8	68.2	68.8	87.4	39.5
CPM [52]	J	78.7	84.9	68.7	69.6	88.8	-
PSTL [56]	J+B+M	79.1	83.8	69.2	70.3	89.2	52.3
CMD [33]	J	79.4	86.9	70.3	71.5	-	43.0
HaLP [38]	J	79.7	86.8	71.1	72.2	-	43.5
HiCo-former [11]	J	81.1	88.6	72.8	74.1	89.3	49.4
SkeAttnCLR [19]	J+B+M	82.0	86.5	77.1	80.0	89.5	55.5
ActCLR [27]	J+B+M	84.3	88.8	74.3	75.7	-	-
Masked prediction:							
SkeletonMAE [49]	J	74.8	77.7	72.5	73.5	82.8	36.1
MAMP [32]	J	84.9	89.1	78.6	79.1	**92.2**	**53.8**
S-JEPA (Ours)	J	**85.3**	**89.8**	**79.6**	**79.9**	**92.2**	53.5

with some of the latest methods that are based on different pre-text objectives. Notably, S-JEPA outperforms the best contrastive learning method, ActCLR [27], by 1.0 percentage point on both NTU60 [39] XSub and XSet datasets. As for the masked prediction methods, we observe that our method achieves superior performance to SkeletonMAE [49] and MAMP [32], both of which are based on self-reconstruction in the joint space as their pretext task. With only joints as input, S-JEPA surpasses the performance of MAMP on four out of six dataset subsets, and slightly lags behind it on PKU Phase II [28]. Its superior performance demonstrates the effectiveness of learning the high-level latent representations instead of low-level joint coordinates, resulting in higher quality representations.

Fine-tuning Evaluation. We fine-tune the pretrained encoder network together with an MLP head for 300 epochs with a batch size of 256. In the first 5 warm-up epochs, we linearly increase the learning rate from 0 to 3e-4. Then, we gradually decrease it to 1e-5 following the cosine decay schedule, with the layer-wise lr decay [10], following [3]. First, we compare with the vanilla transformer when trained from scratch (fully supervised) without pre-training. We observe a performance improvement of +10% and +13.5% on NTU60-XSub and NTU120-XSub, respectively, after pre-training with the S-JEPA framework. This suggests that

Table 2: Accuracy comparison under the fine-tuning evaluation protocol.

Method	Input	Backbone	NTU60 [39]		NTU120 [29]	
			XSub	XView	XSub	XSet
Other pretext task:						
Colorization [50]	J+B+M	DGCNN	88.0	94.9	-	-
Hi-TRS [8]	J+B+M	Transformer	90.0	95.7	85.3	87.4
Contrastive learning:						
CPM [52]	J	ST-GCN	84.8	91.1	78.4	78.9
CrosSCLR [23]	J+B+M	ST-GCN	86.2	92.5	80.5	80.4
AimCLR [16]	J+B+M	ST-GCN	86.9	92.8	80.1	80.9
ActCLR [27]	J+B+M	ST-GCN	88.2	93.9	82.1	84.6
HYSP [14]	J+B+M	ST-GCN	89.1	95.2	84.5	86.3
Masked prediction:						
SkeletonMAE [49]	J	STTFormer	86.6	92.9	76.8	79.1
SkeletonMAE [49]	J	Transformer	88.5	94.7	87.0	88.9
SkeletonMAE [46]	J	STRL	92.8	96.5	84.8	85.7
MotionBERT [57]	J	DSTformer	93.0	97.2	84.8	86.4
MAMP [32]	J	Transformer	**93.1**	97.5	90.0	**91.3**
S-JEPA (Ours)	J	Transformer	**93.1**	**97.6**	**90.3**	**91.3**

S-JEPA helps mitigate the problem of overfitting commonly found in transformers that are trained with insufficient amounts of data. Furthermore, in Table 2, we compare against different approaches on NTU60 [39] and NTU120 [29]. S-JEPA outperforms all recent contrastive learning methods, with a margin of 5.8% and 5.0% on NTU120-XSub and XSet datasets, respectively, compared to HYSP [14]. S-JEPA is also ahead of MotionBERT [57], with 5.5 and 4.9 more percentage points achieved on NTU120-XSub and XSet datasets, respectively, highlighting the importance of predicting 3D skeleton features as opposed to raw joint coordinates. Additionally, S-JEPA achieves superior performance to the recent SOTA methods such as MAMP [32], and SkeletonMAE [46, 49], which use the same vanilla transformer backbone and skeleton format, except on NTU60-XSub [39] and NTU120-XSet [29], in which the performance is tied with MAMP.

Semi-supervised Evaluation. We evaluate our method on the NTU60 dataset [39] using the XSub and XView subsets, employing the same fine-tuning settings but with only 1% and 10% of the available training data labels. Following previous studies [16, 23, 32, 48, 52], we report the mean accuracy of five runs in Table 3 to account for the randomness in training data selection.

Our results show that pre-training with S-JEPA significantly improves performance compared to training a vanilla transformer from scratch, with gains of 28.7 and 17.6 percentage points on NTU60-XSub when using 1% and 10% of the data,

Table 3: Accuracy comparison under the semi-supervised evaluation protocol.

Method	Input	NTU60-XSub [39]		NTU60-XView [39]	
		1%	10%	1%	10%
Other pretext task:					
Colorization [50]	J+B+M	48.3	71.7	52.5	78.9
Hi-TRS [8]	J+B+M	49.3	77.7	51.5	81.1
Contrastive learning:					
ISC [44]	J	35.7	65.9	38.1	72.5
CPM [52]	J	56.7	73.0	57.5	77.1
CrosSCLR [23]	J+B+M	51.1	74.4	50.0	77.8
AimCLR [16]	J+B+M	54.8	78.2	54.3	81.6
CMD [33]	J+B+M	55.6	79.0	55.5	82.4
SkeAttnCLR [19]	J+B+M	59.6	81.5	59.2	83.8
Masked prediction:					
SkeletonMAE [49]	J	54.4	80.6	54.6	83.5
MAMP [32]	J	66.0	88.0	68.7	**91.5**
S-JEPA (Ours)	J	**67.5**	**88.4**	**69.1**	91.4

respectively. Moreover, S-JEPA outperforms the leading contrastive learning method, SkeAttnCLR [19], by 6.9 and 7.6 percentage points on XSub and XView subsets, respectively, with 10% of the labeled data. Compared to MAMP [32], pre-training with S-JEPA shows improvements of 1.5 and 0.4 percentage points with 1% of the XSub and XView training data, respectively. With 10% of the data, S-JEPA exceeds MAMP by 0.4 percentage points on NTU60-XSub but slightly lags behind by 0.1 percentage points on NTU60-XView.

Transfer Learning Evaluation. To assess the generalizability of the learned action representations, we pretrain our model on a source dataset and fine-tune it on a different target dataset. Consistent with previous studies [32, 48, 49], we use PKU-MMD II [28] as the target dataset, and pretrain on NTU60 [39], NTU120 [29], and PKU-MMD I [28].

The results, shown in Table 4, indicate that our framework yields the most transferable representations across the three source datasets compared to previous state-of-the-art methods. Specifically, S-JEPA outperforms SkeletonMAE [49], which reconstructs raw joint coordinates in the joint space, by 13.0%, 13.2%, and 8.4% when using NTU60, NTU120, and PKU-MMD I as source datasets, respectively. Additionally, S-JEPA surpasses MAMP, which relies on masked motion prediction, by 0.8, 1.0, and 0.8 percentage points with the three datasets, respectively. These results highlight the effectiveness of predicting depth- and context-rich latent features in achieving more transferable representations.

Table 4: Accuracy on PKU-II [28] under the transfer learning evaluation. The source datasets are NTU60-XSub [39], NTU120-XSub [29], and PKU-I [28].

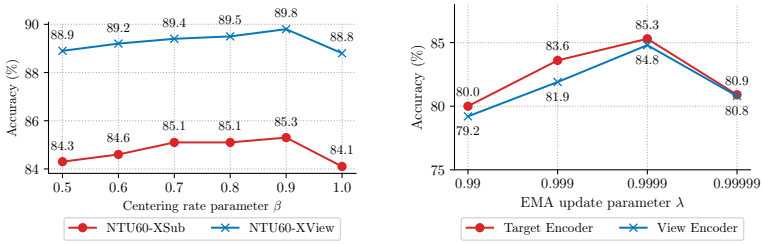
Method	To PKU-II [28]		
	NTU60 [39]	NTU120 [29]	PKU-I [28]
LongTGAN [54]	44.8	-	43.6
MS^{2}L [26]	45.8	-	44.1
ISC [44]	51.1	52.3	45.1
CMD [33]	56.0	57.0	-
SkeletonMAE [49]	58.4	61.0	62.5
MAMP [32]	70.6	73.2	70.1
S-JEPA (Ours)	**71.4**	**74.2**	**70.9**

Table 5: Importance of each component in our framework. EMA: momentum update of target encoder, GT: Geometric Transformations, CE: Cross-Entropy, MSE: Mean Square Error. The performance is evaluated on the XSub sets of NTU60 [39] and NTU120 [29] under the linear evaluation protocol.

	EMA	GT	Loss	Centering	NTU60 [39]	NTU120 [29]
1	✓	✓	CE	✓	85.3	79.6
2	✗	✓	CE	✓	1.6	0.83
3	✓	✗	CE	✓	84.6	78.9
4	✓	✓	MSE	✓	83.7	76.2
5	✓	✓	CE	✗	84.5	79.4

4.4 Ablation Study

Importance of each component in our framework. In Table 5, we analyze the effect of each component in our framework on the XSub sets of NTU60 [39] and NTU120 [29] under the linear evaluation protocol. As shown in row 2, when we remove the momentum update of the target encoder (*i.e.,* the target encoder weights become identical to those of the view encoder), the model collapses to trivial solutions (predicting a constant value regardless of the input). Further, row 3 shows the results when removing the Geometric Transformations (GT) (*i.e.,* using the same skeleton view for both encoders). Removing GT hurts the model performance by 0.7 percentage points on both NTU60 and NTU120, which we attribute to the loss in diversity of the skeleton views. Additionally, we experiment with replacing CE loss with MSE (row 4), and we observe that MSE is not optimal since it negatively affects the quality of the learned representations. Finally, removing the centering operation (row 5) results in a performance drop of 0.8 and 0.2 points on NTU60 and NTU120, respectively. Further, in Figure 3a, We explore the effect of different values of the centering rate hyperparameter β on accuracy achieved on the two subsets of NTU60 under the linear evaluation



(a) Ablation on batch centering rate hyperparameter β of the target representations vs the linear evaluation accuracy on the two subsets of NTU60.

(b) Comparison between the performance of the target and view encoders on NTU60-XSub under linear evaluation against different values of λ.

Fig. 3: Ablation on (a) centering effect on loss and (b) centering rate β

protocol. We observe that $\beta = 0.9$ leads to optimal performance and is, therefore, chosen in our framework.

Effect of EMA update and prediction targets. In Figure 3b, we experiment with four different initial values of EMA update rate λ. Intuitively, λ determines the speed at which the parameters of the target encoder are updated. We empirically find that $\lambda = 0.9999$ leads to the best performance. Further, we compare the performance of the view and target encoders under the linear evaluation protocol and we observe that the target encoder consistently outperforms the view encoder across different λ values, as shown in Figure 3b. The momentum update of the target encoder can be interpreted as form of model ensembling [20]. To clarify, building the target encoder with momentum update means aggregating the values of many previous versions of the view encoder weights through the EMA update rule. The momentum update, therefore, smooths out the optimization trajectory, enabling faster convergence and improved generalization. Hence, this model ensembling, *i.e.,* the target encoder, is used to guide the predictor and the overall training of the network.

Masking input vs output of the target encoder. We study the effect of masking the target encoder input, instead of the output, on the linear evaluation performance in Table 6. We observe that masking the input significantly hurts the model performance by 7.2 percentage points compared to masking the output. We attribute this to the loss of context information in target representations when masking the input, since the self-attention mechanism of the transformer encoder only captures the correlations between the target joints, overlooking those between the masked joints. Instead, when masking the output, the target representations encode the joint-wise correlations from all joints across different time steps, serving as richer prediction targets. Hence, masking the *output* of the target encoder is essential to achieving good performance in our framework.

Table 6: Ablation on masking the target encoder output. Results reflect the performance on NTU60-XSub [39] under the linear evaluation protocol.

Target Masking	Epochs	Accuracy
Input	800	78.1
Output	800	**85.3**

Table 7: Ablation on the design of the encoder and the predictor. Results reflect the performance on NTU60 [39] under the linear evaluation protocol.

L_e	NTU60 XSub	NTU60 XView
7	85.0	89.4
8	**85.3**	**89.8**
9	85.2	89.7
10	**85.3**	89.5

(a) Encoder depth.

C_e	NTU60 XSub	NTU60 XView
64	84.0	88.1
128	85.1	89.3
256	**85.3**	**89.8**
512	**85.3**	89.7

(b) Encoder width.

L_p	NTU60 XSub	NTU60 XView
4	85.0	89.5
5	**85.3**	**89.8**
6	85.1	89.5
7	85.1	89.3

(c) Predictor depth.

C_p	NTU60 XSub	NTU60 XView
64	84.6	88.5
128	84.8	89.1
256	**85.3**	**89.8**
512	85.1	**89.8**

(d) Predictor width.

Design of encoder and predictor. In Table 7, We experiment with different design choices for the encoder and predictor in S-JEPA. In Tables 7a and 7b, we study the effect of the encoder depth L_e (number of layers) and width C_e (feature dimension) on the linear probing accuracy on the two subsets of NTU60 [39]. Our framework exhibits the best performance at $L_e = 8$ and $C_e = 256$. As for the predictor (Tables 7c and 7d), we find that predictor depth $L_p = 5$ and feature dimension $C_p = 256$ bring the best performance. In case there's a tie in performance, we opt for the lower depth or width in our framework. Overall, we adopt an encoder depth and width of 8 and 256, and a predictor depth and width of 5 and 256, respectively by default in S-JEPA.

5 Conclusion

We introduce S-JEPA, a non-generative approach for the self-supervised learning of strong off-the-shelf action representations. We show that by taking the masked prediction pretext task from the joint space to the latent space, the model learns richer skeleton representations that encode high-level contextual action information. Additionally, we propose a batch centering operation that stabilizes the pretraining loss, enabling the model to learn higher quality representations by facilitating better convergence within the same number of training epochs. We verify through extensive evaluations on three popular action recognition datasets the superiority of S-JEPA compared to previous SOTA methods, demonstrating the effectiveness of the proposed pretext task.

Acknowledgement.

This research is supported by the Swiss National Science Foundation through a Sinergia grant for the interdisciplinary project *Narratives from the Long Tail: Transforming Access to Audiovisual Archives*, grant number CRSII5_198632, see https://www.futurecinema.live/project/%7D for a project description. We deeply thank Kaouther Messaoud for her insightful feedback.

References

1. Abdelfattah, M., Hassan, M., Alahi, A.: Maskclr: Attention-guided contrastive learning for robust action representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18678–18687 (2024)
2. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15619–15629 (2023)
3. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. ICLR (2022)
4. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
7. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. in 2021 ieee. In: CVF International Conference on Computer Vision (ICCV). pp. 9620–9629 (2021)
8. 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: European Conference on Computer Vision. pp. 185–202. Springer (2022)
9. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 183–192 (2020)
10. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. ICLR (2020)
11. Dong, J., Sun, S., Liu, Z., Chen, S., Liu, B., Wang, X.: Hierarchical contrast for unsupervised skeleton-based action representation learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 525–533 (2023)
12. 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)

13. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1110–1118 (2015)
14. Franco, L., Mandica, P., Munjal, B., Galasso, F.: Hyperbolic self-paced learning for self-supervised skeleton-based action representations. Int. Conf. Learn. Represent. (2023)
15. 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. Advances in neural information processing systems **33**, 21271–21284 (2020)
16. 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: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 762–770 (2022)
17. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
19. Hua, Y., Wu, W., Zheng, C., Lu, A., Liu, M., Chen, C., Wu, S.: Part aware contrastive learning for self-supervised action recognition. Int. J. Comput. Vis. (2023)
20. Jean, S., Cho, K., Memisevic, R., Bengio, Y.: On using very large target vocabulary for neural machine translation. arXiv preprint arXiv:1412.2007 (2014)
21. Jing, L., Vincent, P., LeCun, Y., Tian, Y.: Understanding dimensional collapse in contrastive self-supervised learning. In Proceedings of the 10th International Conference on Learning Representations (ICLR) (2022)
22. LeCun, Y.: A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. Open Review **62** (2022)
23. Li, L., Wang, M., Ni, B., Wang, H., Yang, J., Zhang, W.: 3d human action representation learning via cross-view consistency pursuit. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4741–4750 (2021)
24. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3595–3603 (2019)
25. Li, T., Ke, Q., Rahmani, H., Ho, R.E., Ding, H., Liu, J.: Else-net: Elastic semantic network for continual action recognition from skeleton data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13434–13443 (2021)
26. Lin, L., Song, S., Yang, W., Liu, J.: Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2490–2498 (2020)
27. Lin, L., Zhang, J., Liu, J.: Actionlet-dependent contrastive learning for unsupervised skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
28. 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)

29. 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 transactions on pattern analysis and machine intelligence **42**(10), 2684–2701 (2019)
30. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 143–152 (2020)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. Proceedings of the International Confer- ence on Learning Representations (ICLR) (2019)
32. Mao, Y., Deng, J., Zhou, W., Fang, Y., Ouyang, W., Li, H.: Masked motion predictors are strong 3d action representation learners. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10181–10191 (2023)
33. Mao, Y., Zhou, W., Lu, Z., Deng, J., Li, H.: Cmd: Self-supervised 3d action representation learning with cross-modal mutual distillation. In: European Conference on Computer Vision. pp. 734–752. Springer (2022)
34. Moliner, O., Huang, S., Åström, K.: Bootstrapped representation learning for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4154–4164 (2022)
35. 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)
36. Rida, M., Abdelfattah, M., Alahi, A., Khovalyg, D.: Toward contactless human thermal monitoring: A framework for machine learning-based human thermophysiology modeling augmented with computer vision. Building and Environment **245**, 110850 (2023)
37. Rodrigues, O.: On the geometric laws governing the displacement of a solid body in space, and on the variation of coordinates resulting from these displacements considered independently of the causes that may produce them. Journal of Pure and Applied Mathematics **5**, 380–440 (1840), https://gallica.bnf.fr/ark:/12148/bpt6k4335701
38. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18846–18856 (2023)
39. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1010–1019 (2016)
40. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7912–7921 (2019)
41. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Adasgn: Adapting joint number and model size for efficient skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13413–13422 (2021)
42. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1227–1236 (2019)
43. Su, K., Liu, X., Shlizerman, E.: Predict & cluster: Unsupervised skeleton based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9631–9640 (2020)

44. Thoker, F.M., Doughty, H., Snoek, C.G.: Skeleton-contrastive 3d action representation learning. In: Proceedings of the 29th ACM international conference on multimedia. pp. 1655–1663 (2021)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
46. 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)
47. Xiong, W., Bertoni, L., Mordan, T., Alahi, A.: Simple yet effective action recognition for autonomous driving. In: 11th Triennial Symposium on Transportation Analysis Conference (TRISTAN XI) (2022)
48. Xu, R., Huang, L., Wang, M., Hu, J., Deng, W.: Skeleton2vec: A self-supervised learning framework with contextualized target representations for skeleton sequence. arXiv preprint arXiv:2401.00921 (2024)
49. Yan, H., Liu, Y., Wei, Y., Li, Z., Li, G., Lin, L.: Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5606–5618 (2023)
50. Yang, S., Liu, J., Lu, S., Er, M.H., Kot, A.C.: Skeleton cloud colorization for unsupervised 3d action representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13423–13433 (2021)
51. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021)
52. Zhang, H., Hou, Y., Zhang, W., Li, W.: Contrastive positive mining for unsupervised 3d action representation learning. In: European Conference on Computer Vision. pp. 36–51. Springer (2022)
53. Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., Zheng, N.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1112–1121 (2020)
54. 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: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
55. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. International Conference on Learning Representations (2022)
56. Zhou, Y., Duan, H., Rao, A., Su, B., Wang, J.: Self-supervised action representation learning from partial spatio-temporal skeleton sequences. AAAI (2023)
57. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: A unified perspective on learning human motion representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)