Skeleton-Contrastive 3D Action Representation Learning

Fida Mohammad Thoker
University of Amsterdam
f.m.thoker@uva.nl

Hazel Doughty
University of Amsterdam
hazel.doughty@uva.nl

Cees G.M. Snoek
University of Amsterdam
cgmsnoek@uva.nl

ABSTRACT

This paper strives for self-supervised learning of a feature space suitable for skeleton-based action recognition. Our proposal is built upon learning invariances to input skeleton representations and various skeleton augmentations via a noise contrastive estimation. In particular, we propose inter-skeleton contrastive learning, which learns from multiple different input skeleton representations in a cross-contrastive manner. In addition, we contribute several skeleton-specific spatial and temporal augmentations which further encourage the model to learn the spatio-temporal dynamics of skeleton data. By learning similarities between different skeleton representations as well as augmented views of the same sequence, the network is encouraged to learn higher-level semantics of the skeleton data than when only using the augmented views. Our approach achieves state-of-the-art performance for self-supervised learning from skeleton data on the challenging PKU and NTU datasets with multiple downstream tasks, including action recognition, action retrieval and semi-supervised learning. Code is avaliable at https://github.com/fmthoker/skeleton-contrast.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition**.

KEYWORDS

skeleton action recognition; contrastive learning; self-supervision

ACM Reference Format:
Fida Mohammad Thoker, Hazel Doughty, and Cees G.M. Snoek. 2021. Skeleton-Contrastive 3D Action Representation Learning. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3474085.3475307

1 INTRODUCTION

The goal of this paper is to learn a latent feature space suitable for 3D human action understanding. Different from traditional RGB frames [2, 15], skeleton data consists of 3D coordinates representing the major joints of each person in a video [5, 20, 32]. It offers a light-weight representation that can be processed faster and in a privacy-preserving manner providing application potential in video surveillance, assisted living, gaming and human-computer interaction. Moreover, when compared to RGB, such a representation is robust to changes in background and appearance [23, 46]. However, learning a good feature space for 3D actions requires large amounts of labeled skeleton data [7, 12, 35, 36, 44–46], which is much harder to obtain than large amounts of labeled RGB video. To address this major shortcoming, we propose a new self-supervised contrastive learning method for 3D skeleton data.

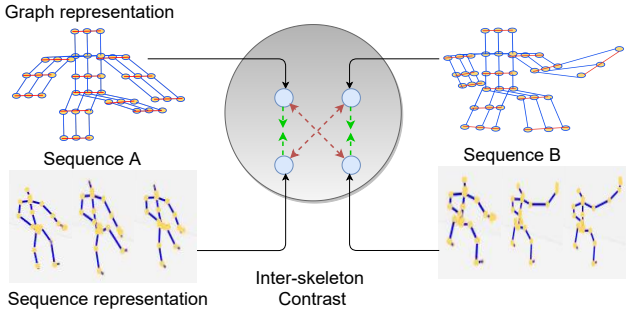

Figure 1: Inter-skeleton contrast learns high-level semantics of skeleton data in a self-supervised fashion. While contrastive methods normally learn invariance to augmentations we additionally learn invariance to the input representation. Different representations of the same sequence are encouraged to be close together in the feature space, while being far away from other sequences.

Several previous works also considered self-supervised learning for 3D skeleton data [19, 27, 39, 49]. These works design pretext tasks, such as learning to reconstruct masked input [49] and motion prediction [19], which still require the features to represent variations such as the viewpoint and skeleton scale, rather than focusing on higher-level semantic features relevant to downstream tasks. Instead, we take inspiration from recent self-supervised literature for RGB images, which aims to learn the high-level similarity between augmented forms of the same image and the dissimilarity between these and other images [3, 11, 29]. At the core of such contrastive learning is the nature of the RGB data, where each sample contains abundant pixel information, allowing for augmentations like spatial-cropping and color-jittering to easily generate subtly different versions of an image without changing its semantic content. However, skeleton sequences are much more sparse than RGB data and the augmentations commonly applied to images would not change the estimated skeleton of a person. Thus, for contrastive learning with skeleton sequences, we need skeleton-specific augmentations to encourage the learned features to encode information relating to spatio-temporal dynamics of the joints. We also want to enrich the input space which can be sampled from, to increase the variety of samples with the same semantic content, and thus increase the difficulty of the contrastive learning task.

We make three contributions. Our first contribution is to leverage multiple input-representations of the 3D-skeleton sequences. In

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475307

particular, we propose inter-skeleton contrast to learn from a pair of skeleton-representations in a cross-contrastive fashion, see Figure 1. This allows us to enrich the sparse input space and focus on the high-level semantics of the skeleton data rather than the nuances of one specific input representation. Second, we introduce several skeleton-specific spatial and temporal augmentations for generating positive pairs which encourage the model to focus on the spatio-temporal dynamics of skeleton-based action sequences, ignoring confounding factors such as viewpoint and the exact joint positions. Finally, we provide a comprehensive evaluation of our learned feature space on various challenging downstream tasks, showing considerable improvement over prior methods in all tasks.

2 RELATED WORK

Self-Supervised Learning. Self-supervised learning strives to learn feature representations without human annotation, typically by solving *pretext tasks* which exploit the structure of unlabeled data. Previous works have proposed a variety of such tasks for learning image representations, *e.g.* solving spatial jigsaw puzzles [28], rotation prediction [9], spatial context-prediction [6], image inpainting [30] and colorization [47, 48]. Similarly, pre-text tasks have been designed for learning video representations, such as spatio-temporal puzzles [14], prediction of frame-order [8], clip-order [43], speed [1], future [10] and temporal coherence [16]. Such pretext tasks rely on the rich structured nature of RGB data with the hope that by learning to solve these tasks the encoded features will rely on the high-level semantics of the image or video and are thus applicable to the downstream task(s). Unfortunately, these existing RGB-based pretext tasks are not suited for 3D-skeleton sequences which have a simple structure and are less rich in information.

Instead of designing specific pretext tasks, recent self-supervised methods rely on instance discrimination and learn the similarity between sample pairs [3, 11, 26, 29, 40]. A noise contrastive loss learns invariances to certain image or video transformation functions, resulting in good feature representations. For example, Chen *et al.* [3] show that learning invariance to simple image augmentations, such as color jitter, results in highly discriminative features. He *et al.* [11] propose a momentum contrast which is able to utilize a large number of negatives for the noise contrast by storing image features from previous batches in a dynamic queue. In this paper, we rely on contrastive estimation for 3D action representation learning. As existing works use augmentations specific to RGB images, we introduce three skeleton-specific augmentations to generate positive pairs for learning the spatio-temporal dynamics of 3D-skeleton sequences. Furthermore, we propose inter-skeleton contrastive learning which additionally aims to learn invariance to the particular input representation of the 3D-skeleton sequences.

Supervised 3D Action Recognition. Numerous methods for supervised 3D action recognition exist. While earlier methods design handcrafted features [25, 41, 42] to model geometric relationships between skeleton joints, recent approaches rely on data-driven deep neural networks. Three skeleton-representations have become popular for deep learning. Sequence-based treats the 3D-skeleton data as a multi-dimensional time-series and models it with a recurrent architecture [21, 22, 32, 35, 46] to learn the temporal dynamics of the joints. Image-based create a pseudo-image representation of the 3D-skeleton data [7, 12, 17, 23, 38] which is encoded by CNN architectures to model the co-occurrence of multiple joints and their motion. Finally, graph-based [4, 13, 18, 24, 31, 33, 37, 44] represents the 3D-skeleton data with a graph consisting of spatial and temporal edges. Graph-convolutional architectures then encode the spatio-temporal motion from the human skeleton graph. Although these methods achieve excellent performance, they are all fully supervised and require time-consuming action class annotations. We propose a self-supervised method for 3D-skeleton data that leverages the diversity of the skeleton-representations to learn highly discriminative features from unlabeled data.

Self-Supervised 3D Action Recognition. Overcoming the need for large amounts of annotations has only recently received attention in the 3D action recognition community. Zheng *et al.* [49] propose a seq2seq model that learns to reconstruct masked input 3D-skeleton sequences. In particular, a GAN is trained such that the decoder attempts to regenerate the input sequences, while a discriminator measures the quality of the regenerated sequences. Similarly, Nie *et al.* [27] propose a cross-view reconstruction task that relies on a siamese denoising autoencoder to reconstruct the correct version of corrupted and rotated input skeletons. Su *et al.* [39] also propose a seq2seq model that regenerates input skeleton sequences. To encourage the encoder to learn better latent features, the decoder is weakened by fixing its weights.

Lin *et al.* [19] take a different approach and propose multi-task self-supervised learning for the sequence-based skeleton representation. Their framework solves multiple pretext tasks simultaneously, such as motion prediction and skeleton-jigsaw. Si *et al.* [34] propose an adversarial self-supervised learning approach that couples the self-supervised learning and the semi-supervised scheme via neighbor relation exploration and adversarial learning.

Different from all these works, we do not rely exclusively on a sequence-based skeleton-representation and pretext tasks such as input-reconstruction and motion prediction. Instead, we propose to exploit the diversity of skeleton-representations in an inter-contrastive learning regime and design skeleton-specific spatial and temporal augmentations for use in this contrastive method.

3 SKELETON-CONTRASTIVE LEARNING

In this section we present our inter-skeleton contrast approach for self-supervised learning of 3D action features. Contrastive methods aim to learn a good feature space by learning the similarity between augmented views of the same data. Since augmentations in existing contrastive learning works are primarily designed for RGB images [3] they are not suitable for the skeleton data that considered in this work. Therefore, we first propose several skeleton-specific augmentation functions in Section 3.1. These augmentations enable us to apply existing contrastive learning methods, such as MoCo [11], to skeleton data. We describe this is Section 3.2.

However, contrastive learning can be vulnerable to shortcuts, where simple features, irrelevant to the downstream task, may be enough to identify the different augmented views of the same data. For instance, Chen *et al.* [3] show that color distributions can be a shortcut to identify different crops from the same image. To avoid such shortcuts and make the contrastive learning task more difficult, we additionally contrast pairs of different input

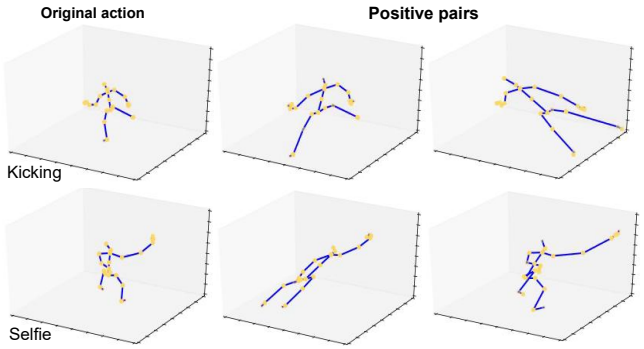

Figure 2: Spatial pose augmentation examples. A shear operation is applied to the original action so that the augmented pairs differ in viewpoint and camera distance.

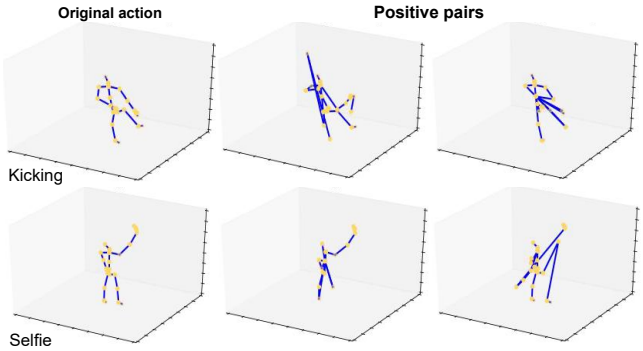

Figure 3: Spatial joint jittering examples. The augmented pairs contain a subset of common joint connections while other joint connections are randomly moved to an irregular position.

skeleton representations with each other. We call this *inter-skeleton contrastive learning* and detail our approach in Section 3.3.

3.1 Skeleton Augmentations

The goal of contrastive learning is to learn the semantic similarity between items in a dataset without labels. This is usually done by learning the similarity of two augmented views (positive pairs) of a sample X. A data augmentation function D, composed of a single or multiple transformations, creates the augmented views. Hence, the network learns features for X, which are invariant to the transformations in D. The nature of the data X and the downstream task determines the appropriate invariances that the learned features should possess. In our case, X is a 3D-skeleton sequence, where each sequence represents a particular spatial configuration of human joints and its motion over a short period of time. Thus, to learn useful representations for 3D-skeleton data, the commonly used RGB augmentations, such as color-distortion and Gaussian blurring [3], are not suitable. Instead, we need to learn invariances to transformations that encode the spatial and temporal dynamics of 3D skeleton action sequences. We introduce multiple spatial and temporal skeleton augmentation techniques to generate positive pairs for 3D-skeleton action sequences: *Pose Augmentation*, *Joint Jittering* and *Temporal Crop-Resize*. We then combine these to create our final spatio-temporal skeleton augmentation. Let us assume each raw action sequence $X \in R^{T\times J\times 3}$ consists of 3D coordinates of J body joints in T consecutive video frames. We define our individual augmentations D based on X.

3.1.1 ***Spatial Skeleton Augmentations.*** To apply our learned feature space to downstream tasks such as 3D action recognition, we require the feature encodings to rely on more discriminatory spatial semantics like joint configurations, while being invariant to factors such as viewpoint, camera distance, skeleton scale and joint perturbations. Existing augmentations for RGB images would not achieve this, thus we propose two new skeleton-specific spatial augmentations: pose augmentation and joint jittering. These can be applied to each of the T skeletons in the sequence X so a contrastive learning framework can learn invariance to these augmentations.

Pose Augmentation. With this transformation, we aim to create positive pairs which differ in viewpoint and distance to the camera, while retaining the same pose from the original sequence. To achieve this, we apply a 3D shear on the action sequence X:

$$D_{Spatial_1}(X) = X \cdot \begin{bmatrix} 1 & r_{01} & r_{02} \\ r_{10} & 1 & r_{12} \\ r_{20} & r_{21} & 1 \end{bmatrix}, \tag{1}$$

where the elements of the augmentation matrix are randomly drawn from a uniform distribution $[-1, 1]$. Figure 2 shows several examples. By applying the same shearing operation to each joint of the skeleton at each time-step in the sequence we are able to simulate changes in camera viewpoint and distance between the subject and camera. Therefore, a contrastive network which learns invariance to this transformation is forced to learn more discriminatory pose semantics of the positive pairs and ignore redundant information such as the viewpoint and proximity to the camera.

Joint Jittering. We also want a contrastive method to be invariant to noise in the estimated skeleton. Therefore we propose joint jittering to create positive pairs where some of the joint connections in X are randomly perturbed. We select j of the J joints at random and move these joints to irregular positions, while keeping other joints in their original position. The transformation is defined as:

$$D_{Spatial_2}(X) = X[:, j] \cdot \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}, \tag{2}$$

where j is a subset of the joints such that $|j| < J$, and the elements of the jitter matrix are randomly drawn from a uniform distribution $[-1, 1]$. The same jitter matrix is applied to each joint in j at each time-step T. Examples are shown in Figure 3. To learn invariance to such transformations, the contrastive task is encouraged to rely on the spatio-temporal semantics of the common joint connections and ignore the noise from the irregular joint connections.

3.1.2 ***Temporal Skeleton Augmentation.*** Besides the spatial perturbations, a good 3D skeleton feature space should also be robust to temporal modifications of the skeleton sequences, such as the speed of an action and changes to the temporal bounds of the sequence. To this end, we propose temporal crop-resize.

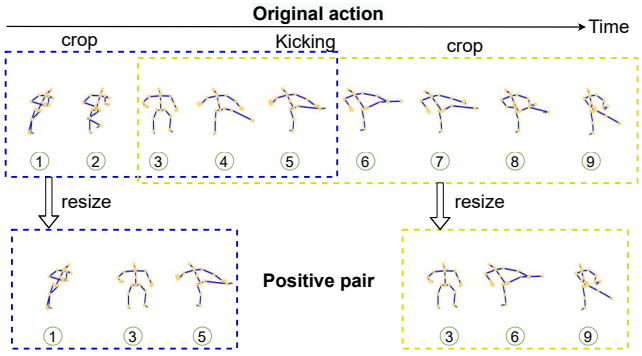

Figure 4: Temporal crop-resize. The augmented views start at different time steps and sample different temporal periods (blue and yellow boxes). Each crop is re-sampled to a fixed size, effectively altering its speed depending on the length of the temporal crop.

Temporal Crop-Resize. In this transformation, we create positive pairs with varying speed and varying starting and ending points. We sample different parts of the action sequence X via a random crop and resize this crop over the temporal dimension T:

$$D_{Temporal}(X) = \text{Interpolate}(X[L_{start} : L_{start} + TL_{ratio}]). \quad (3)$$

The length ratio L_{ratio} is first randomly sampled from distribution $[l_{min}, 1.0]$, followed by randomly selecting a starting frame L_{start} between $(0, T - TL_{ratio})$. The sub-sequence $X[L_{start} : L_{start} + TL_{ratio}]$ is then re-sampled to a fixed length. This re-sampling causes the temporal crop-resize to also alter the speed of a sequence as well as its start and end times; a shorter sub-sequence will effectively have a slower speed once re-sampled. Figure 4 shows examples of this transformation. By including this augmentation the contrastive task is forced to focus on the commonalities of the joint motion dynamics over the sampled temporal periods and be robust to changes in the exact start, end and speed of an action.

3.1.3 ***Spatio-Temporal Skeleton Augmentations.*** To learn spatial and temporal dynamics of the skeleton sequences, we propose to combine the above spatial and temporal transformations into a single augmentation function. Such composition results in strong positive pairs which vary in both spatial and temporal dynamics locally, while retaining the high-level semantics of the original action sequence. In particular, we first apply the temporal crop-resize augmentation $D_{Temporal}$ on the original action sequence X followed by a spatial augmentation $D_{Spatial_i}$ to the resulting sequence:

$$D_{Spatio\text{-}Temporal}(X) = D_{Spatial_i}(D_{Temporal}(X)) \quad (4)$$

Here, i can either be fixed to the pose augmentation or the joint jitter or randomized to select either of the spatial augmentations. As we will show in the experiments, learning invariance to spatio-temporal transformations produces a better 3D action feature space and randomizing the composition further improves the result.

3.2 Intra-Skeleton Contrast

Before describing our proposed inter-skeleton method, we first describe how the above augmentations can be incorporated into an existing contrastive method, such as MoCo [11], with a single input skeleton-representation. We call this intra-skeleton contrastive learning. Each raw action sequence $X \in R^{T \times J \times 3}$ is first augmented into two different views X_q and X_k (called query and key) via a data augmentation function D. Both views of the skeleton data are then instantiated into the same skeleton-representation, be it image-based or sequence-based or graph-based. A contrastive method such as MoCo uses two encoders, one for the query and one for the key. We refer to the query encoder as f_q and the key encoder as f_k. Let $(Z_q, Z_k) = (f_q(X_q), f_k(X_k))$ be output embeddings of the encoders for the input query-key pair. We then train the contrastive network using the noise contrastive estimation loss InfoNCE [29]:

$$\mathcal{L}(X) = -\log \frac{\exp(Z_q \cdot Z_k/\tau)}{\exp(Z_q \cdot Z_k/\tau) + \sum_{Z_n \sim \mathcal{N}} \exp(Z_q \cdot Z_n/\tau)}, \quad (5)$$

where τ is a temperature softening hyper-parameter and $\mathcal{N}$ is the current set of negatives that are stored in a dynamic queue via previous states of the key encoder f_k as in [11]. Only the query encoder is actively trained using Equation (5) and the key encoder is updated as a moving average of the query encoder. This trains the framework to learn 3D action features which are invariant to the transformations in D for the chosen skeleton-representation.

3.3 Inter-Skeleton Contrast

Up to this point, our method, like previous contrastive learning approaches [3, 11, 26], learns the similarity between different augmented forms of the same input. We now extend contrastive learning for 3D skeleton data beyond these augmentations and propose inter-skeleton contrast which aims to learn invariance to the *input representation* of the skeleton sequence. Three 3D-skeleton representations are common: *image-based* as a $T \times J$ pseudo-image where the 3D coordinates of each joint are the image channels, *sequence-based* as a multi-dimensional time series, or *graph-based* as a spatio-temporal graph. Each requires a different network architecture and encodes different characteristics of the sequence. For example, RNNs treat skeleton sequences as a time series and explicitly model the temporal evolution of joints, while GCNs treat sequences as a graph with both spatial and temporal edges and thus explicitly encode human pose as well as each joint's temporal motion. While the action depicted by the skeleton sequence is the same, the way the input sequence is represented and encoded is different. To learn invariance to the input representation the contrastive framework has to learn the similarities between the characteristics of these different representations as well as our data augmentations which will result in more discriminative features.

The overall network is depicted in Figure 5. The raw skeleton sequence is first augmented into two views as in Section 3.2. Each view is then represented in two ways, in this case with a graph-based representation and a sequence-based representation. We refer to the different representations of the raw action sequence X as X^{IMG} for image-based, X^{SEQ} for seq-based and X^{STG} for graph-based. For the rest of this section we will take the example of the pair X^{SEQ} and X^{STG} as displayed in Figure 5. We adapt our model to contrast the different input representations by using a pair of momentum contrastive models together, one for each input-representation X^{SEQ}

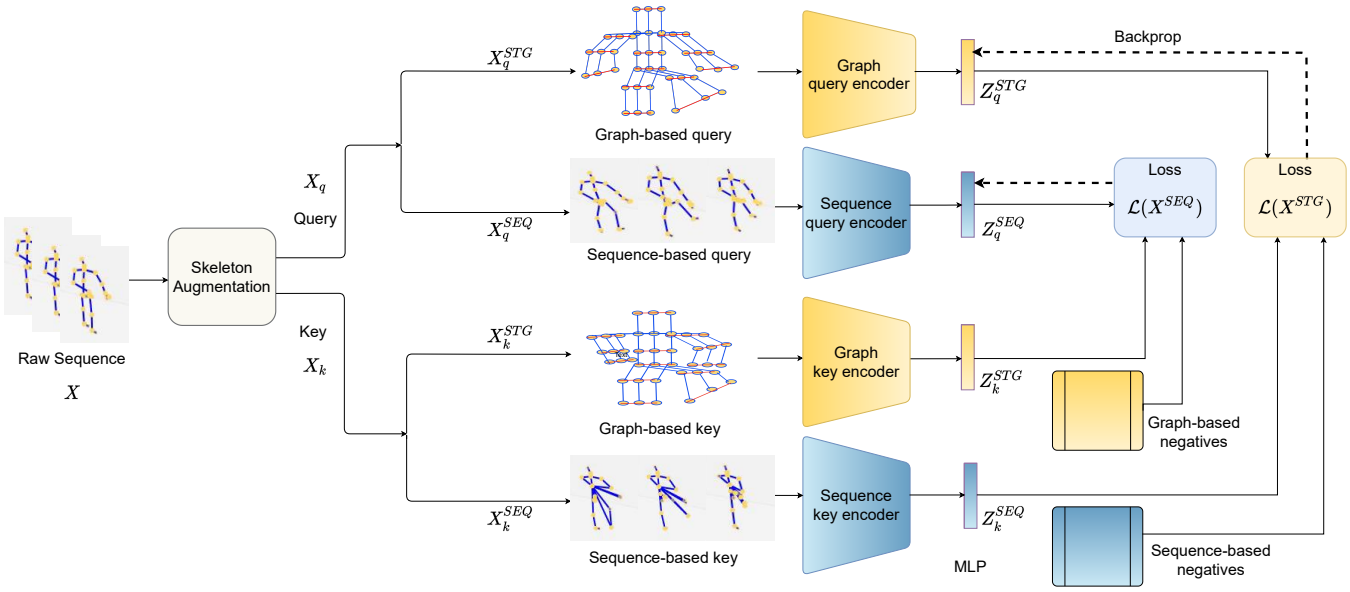

Figure 5: Inter-skeleton contrast. We learn invariances to input skeleton representations, as well data augmentations, in a cross-contrastive manner. We first augment the input sequence into two different views called the query and key using our proposed spatio-temporal augmentations. Each of these views is then represented with two different input skeleton-representations, here graph-based and sequence-based. We encourage the embedding for the graph-based query to be similar to the embedding of the sequence-based key while being dissimilar to the current set of sequence-based negatives. The same applies for the sequence-based query and graph-based key and negatives.

and X^{STG}. In particular, the model now consists of two query encoders f_q^{SEQ} and f_q^{STG} and two key encoders f_k^{SEQ} and f_k^{STG}. A query-key pair (X_q, X_k) is obtained by augmenting a raw action sequence X with D as before. We instantiate two different skeleton-representation pairs (X_q^{SEQ}, X_k^{SEQ}) and (X_q^{STG}, X_k^{STG}). Then, for the query in each input representation, we generate the positives and negatives from the key encoder of the *other* input representation and vice versa. The encoders (f_q^{SEQ}, f_q^{STG}) are trained jointly using a cross-contrastive loss function:

$$\mathcal{L}(X^{SEQ}, X^{STG}) = \mathcal{L}(X^{SEQ}) + \mathcal{L}(X^{STG}), \tag{6}$$

$$\mathcal{L}(X^{SEQ}) = -\log \frac{\exp(Z_q^{SEQ} \cdot Z_k^{STG}/\tau)}{\exp(Z_q^{SEQ} \cdot Z_k^{STG}/\tau) + \sum_{Z_n \sim \mathcal{N}^{STG}} \exp(Z_q^{SEQ} \cdot Z_n^{STG}/\tau)}, \tag{7}$$

$$\mathcal{L}(X^{STG}) = -\log \frac{\exp(Z_q^{STG} \cdot Z_k^{SEQ}/\tau)}{\exp(Z_q^{STG} \cdot Z_k^{SEQ}/\tau) + \sum_{Z_n \sim \mathcal{N}^{SEQ}} \exp(Z_q^{STG} \cdot Z_n^{SEQ}/\tau)}, \tag{8}$$

where $Z_q^{SEQ}{=}f_q^{SEQ}(X_q^{SEQ})$ is the embedding of the sequence-based query and $\mathcal{N}^{SEQ}$ is the current set of negative sequence-based embeddings. These are defined similarly for the other representations and augmentations of X. This formulation serves two purposes. First the input space of the contrastive task is enriched to learn from multiple representations of the same sequence, in addition to the multiple 'views' the data augmentation D provides. Second, different from Equation (5), the cross-contrastive loss *i.e.* Equation (6) forces the framework to rely on mutual information between the embeddings of the two skeleton representations. Thus the contrastive framework is encouraged to focus on higher-level semantics and avoid resorting to shortcut solutions to identify the similarity between query-key pairs.

4 EXPERIMENTS

We first describe the datasets and implementation details. We then demonstrate the effectiveness of our contrastive learning approach on several 3D action understanding downstream tasks. Finally, we ablate the effects of our proposed skeleton augmentations and inter-skeleton contrast.

4.1 Datasets and Evaluation

NTU RGB+D 60 [32]. This is the most commonly used dataset for 3D action recognition. All actions are captured in indoor scenes with three cameras concurrently. The dataset contains 40 different subjects and 60 action classes. Each action sequence is performed by an individual or pair of actors with each actor represented by the 3D coordinates of 25 skeleton joints. The dataset consists of 56,880 video samples and is evaluated under the two standard protocols as suggested by [32]. The first is *cross-view*, where samples from two angles (0°, 45°) are used for training (37,920 samples) and a third angle (−45°) is used for testing (18,960 samples). The second is *cross-subject*, where the actors in the training and testing sets are different, with 40,320 training and 16,560 testing samples.

NTU RGB+D 120 [20]. This is an extension to NTU RGB-D 60 and is currently the largest benchmark for 3D action recognition

	NTU RGB+D 60		NTU RGB+D 120		PKU-MMD I	PKU-MMD II
	x-view	x-sub	x-setup	x-sub	x-sub	x-sub
Zheng *et al.* [49]	56.4	52.1	39.7	35.6	68.7	26.5
Lin *et al.* [19]	–	52.5	–	–	64.8	27.6
Su *et al.* [39]	59.3	56.1	44.1	41.1	59.9	25.5
Nie *et al.* [27]	79.7	–	–	–	–	–
This paper	**85.2**	**76.3**	**67.9**	**67.1**	**80.9**	**36.0**

Table 1: 3D action recognition. Our method learns better 3D-action features from unlabeled data than alternatives, no matter the dataset or evaluation protocol. All results of Zheng *et al.* and Su *et al.* obtained with code provided by Su *et al.*

with 114,480 samples over 120 action classes. Actions are captured with 106 subjects in a multi-view setting using 32 different setups (varying camera distances and background). Each action sample has 1 or 2 subjects, again each is represented by 25 3D-skeleton joints. The dataset is challenging due to the variation in subject, background, viewpoint and fine-grained actions captured. For evaluation, two recommended protocols [20] are used: *cross-setup*, where even-numbered setups are used for training (54,471 samples) and odd-numbered setups are used for testing (59,477 samples), and again *cross-subject*, with 63,026 training and 50,922 testing samples.
PKU-MMD [5]. This dataset was originally proposed for action detection but has also been used for action recognition [19]. It contains 52 human action classes. Each action is represented by the 3D coordinates of the 25 joints of each actor involved in the action. The dataset consists of two parts: ***PKU-MMD I*** and ***PKU-MMD II***, with almost 20,000 and 7,000 action instances. Both parts are challenging for action recognition, as the number of action classes is large while the training sets are relatively small, however PKU-MMD II is more challenging due to the large view variation causing more skeleton noise. We split both sets into a training and a testing set using the recommended *cross-subject* protocol [5]. The training sets of PKU-MMD I & II contain 18,841 and 5,332 samples, while the testing sets contain 2,704 and 1,613 samples.
Evaluation Criteria. For all datasets, protocols and downstream tasks we report the top-1 accuracy.

4.2 Implementation Details

Network Architectures. We instantiate the pair of encoders (f_q, f_k) based on the skeleton-representations used. For the sequence-based encoder f^{SEQ} we rely on a 3-Layer BI-GRU with H=1024 units per layer [39]. For the image representation encoder f^{IMG}, we adopt the CNN based Hierarchical Co-occurrence Network (HCN) [17]. For the graph representation encoder f^{STG}, a joint based graph-convolutional network A-GCN [33] is used. We represent each skeleton sequence X as two people, with the second actor being all zeros for single actor actions. The augmented forms of the raw skeleton sequence X (X_q and X_k) have temporal length 64. Unless mentioned otherwise we use $|j| = 15$ for the joint jitter augmentation and $l_{min} = 0.1$ for the temporal crop-resize augmentation.
Self-Supervised Pretraining. Our inter-skeleton contrastive network is based on MOCO [11] and is trained on the training data without any labels. A projection head (an MLP) is appended to each encoder to produce embeddings of a fixed size of 128. The embeddings are L2-normalized before computing the contrastive loss. We train the whole network with a temperature value of τ=0.07, an SGD optimizer, a learning rate of 0.01 and a weight decay of 0.0001. For NTU RGB+D 60 & 120, the size of the set negatives $\mathcal{N}$ is 16,384 and the model is pre-trained for a total of 450 epochs. For PKU-MMD I & II, the size of $\mathcal{N}$ is set to 8,192 and 2,048, and we pre-train for 600 epochs. The training and evaluation details of the downstream tasks are discussed in the supplementary material.

4.3 Downstream Tasks

In this section, we evaluate the 3D action features learned by our inter-skeleton contrast for various downstream tasks in comparison with the respective state-of-the-art in self-supervised learning. For a fair comparison we follow the setups of prior works and only train and evaluate downstream tasks with the sequence-based input representation X^{SEQ}. In particular, we pre-train our inter-skeleton contrast network with X^{SEQ} and X^{STG} skeleton representations as this gives the best result (see Section 4.4) and evaluate only the sequence-based query encoder f_q^{SEQ}. We also show some qualitative results in the supplementary material.
3D Action Recognition. We compare our method to prior works in self-supervised learning for skeleton data by training a linear classifier on top of the frozen features from our inter-skeleton contrast. We compare with the proposed methods of Zheng *et al.* [49], Su *et al.* [39] and Nie *et al.* [27], all of which use reconstruction of the skeleton sequence as a pretext task. We also compare to the multi-task self-supervised method by Lin *et al.* [19], which uses skeleton-jigsaw and motion prediction as auxiliary tasks.

We present results on the NTU RGB+D 60, NTU-120 and PKU-MMD (I and II) datasets in Table 1. It is evident our inter-skeleton contrast outperforms all methods by a considerable margin on each benchmark. We conclude the self-supervised feature space learned by our method is state-of-the-art for 3D action recognition.
3D Action Retrieval. We follow the setup introduced by Su *et al.* [39]. We apply the kNN classifier (k=1) to the pre-trained features of the training set to assign classes. We match each test sample to the most similar training class using cosine similarity. Besides comparison with Su *et al.* [39], we also compare with Zheng *et al.* [49], using numbers and code provided by Su *et al.* We present results for NTU RGB+D 60 and NTU RGB+D 120 in Table 3. For both datasets, our method outperforms the alternatives, especially for the more challenging cross-subject and cross-setup protocols. Both [39, 49] rely on an input reconstruction pretext-task for learning their feature space, which easily captures varying viewpoints. However,

	NTU RGB+D 60								PKU-MMD I	
	x-view				x-sub				x-sub	
	(1%)	(5%)	(10%)	(20%)	(1%)	(5%)	(10%)	(20%)	(1%)	(10%)
Zheng *et al.* [49]	-	-	-	-	35.2	-	62.0	-	34.4	69.5
Lin *et al.* [19]	-	-	-	-	33.1	-	65.1	-	36.4	70.3
Si *et al.* [34]	-	63.6	69.8	74.7	-	57.3	64.3	68.0	-	-
This paper (supervised only)	21.7 ±1.0	47.6 ±1.0	59.8 ±0.5	69.1 ±0.5	17.6 ±0.5	42.8 ±0.5	51.6 ±1.0	59.5 ±1.0	22.5 ±1.0	55.4 ±1.0
This paper	**38.1** ±1.0	**65.7** ±0.5	**72.5** ±0.4	**78.2** ±0.3	**35.7** ±0.5	**59.6** ±0.5	**65.9** ±1.0	**70.8** ±1.0	**37.7** ±1.0	**72.1** ±1.0

Table 2: Semi-supervised 3D action recognition. We report average accuracy of five runs with random subsets of labeled samples. Pre-training with our inter-skeleton shows improvement over prior semi-supervised works as well as training only with the labeled subset.

	NTU RGB+D 60		NTU RGB+D 120	
	x-view	x-sub	x-setup	x-sub
Zheng *et al.* [49]	48.1	39.1	35.5	31.5
Su *et al.* [39]	76.3	50.7	41.8	39.5
This paper	**82.6**	**62.5**	**52.3**	**50.6**

Table 3: 3D action retrieval. Results for Zheng *et al.* and Su *et al.* in [39] obtained with code provided by Su *et al.* Our method learns best features for retrieval than prior self-supervised methods.

	Transfer to PKU-MMD II	
	PKU-MMD I	NTU RGB+D 60
Zheng *et al.* [49]	43.6	44.8
Lin *et al.* [19]	44.1	**45.8**
This paper	**45.1**	**45.9**

Table 4: Transfer learning for 3D action recognition. All results by Zheng *et al.* provided by Lin *et al.* in [19]. Knowledge gained via inter-skeleton contrastive pretraining transfers well, especially when source and target datasets are more similar.

with a simple reconstruction, it is difficult to capture variation with respect to subjects and setups as our inter-skeleton contrast can.

Semi-Supervised 3D Action Recognition. In the semi-supervised setting, a network utilizes both labeled and unlabeled data during the training process. Following prior work for semi-supervised learning in 3D action recognition, we first train our encoder on our unsupervised inter-skeleton contrastive learning task. Then, we fine-tune the final classification layer and the pre-trained encoder together using a portion of the data labeled with the action class. Again, we compare with Zheng *et al.* [49] and Lin *et al.* [19] as well as the method of Si *et al.* [34] on NTU RGB+D 60 and the PKU-MMD I datasets. To compare with prior works, we report results when using 1%, 5%, 10% and 20% of the training data with labels for NTU RGB+60 and when using 1% and 10% of the labels for PKU-MMD I. The rest of the training set is used as the unlabeled data.

The results in Table 2 reveal that our method outperforms all previous methods on each benchmark. We also demonstrate a large improvement over supervised only training, *i.e.* training with only the available labeled data from randomly initialized weights. From these results we can see that our inter-skeleton contrastive learning is especially suited to learn from both unlabeled and labeled skeleton data in order to boost the performance of 3D action recognition.

Transfer Learning for 3D Action Recognition. To evaluate if knowledge gained from a source dataset generalizes to a different target dataset, we also consider transfer learning. In this setting, an encoder network is first trained on the source dataset for our inter-skeleton contrastive task, followed by jointly finetuning the pretrained encoder and a classifier on a target dataset for action recognition. As in Lin *et al.* [19], we use NTU RGB+D 60 and PKU-MMD I as the source datasets and PKU-MMD II as the target dataset. Table 4 shows our features are just as or more transferable than those of Zheng *et al.* [49] and Lin *et al.* [19], especially for transfer from PKU-MMD I to PKU-MMD II which are from same domain. Thus, the knowledge gained by our method from a source dataset can improve action classification accuracy on a different target set, especially one with a similar domain.

4.4 Ablation Studies

We now ablate the effect of each of our skeleton augmentations and demonstrate the effectiveness of our inter-skeleton contrastive learning. These ablations are performed on the cross-view protocol of NTU RGB+D 60 for the downstream task of 3D action recognition. As before, after pre-training the models with our contrastive self-supervision methods, we train a linear classifier with action labels on top of the frozen features of the query encoder f_q.

Benefit of Skeleton Augmentation. First, we show the benefit of each of the proposed skeleton augmentations when learning from a single input skeleton representation. We choose as skeleton augmentation function D, either pose augmentation, joint jitter, temporal crop-resize or combinations thereof, and train an intra-skeleton contrastive model as described in Section 3.2.

Table 5 shows the accuracy of our augmentations with each input representation. We find that all of the proposed spatial and temporal skeleton augmentations individually perform better than using no augmentation. Thereby, reinforcing our claim that learning invariances to spatial changes like viewpoints, scale and joint perturbations, or, temporal changes such as delay and speed result in learning good action features. The composition of augmentations further improves the accuracy by a considerable margin for all input representations, with the best combination being the inclusion of all three augmentation functions. For example, the final accuracy with the X^{IMG} representation is a ~10% increase over using only pose augmentation and ~28% over using no augmentation.

Augmentations			Downstream Representation		
Temporal Crop-resize	Pose	Joint Jitter	X^{IMG}	X^{STG}	X^{SEQ}
-	-	-	51.0	51.4	50.0
✓	-	-	62.5	53.5	64.1
-	✓	-	69.8	63.8	71.7
-	-	✓	74.6	66.1	75.2
✓	✓	-	73.2	69.3	73.8
✓	-	✓	77.0	68.3	80.0
✓	✓	✓	**79.6**	**72.5**	**82.5**

Table 5: Benefit of skeleton augmentation. We ablate the effect of our augmentations with 3D action recognition on NTU RGB+D 60. Combining all three augmentations generates strong positive pairs for increased accuracy, no matter the 3D action representation.

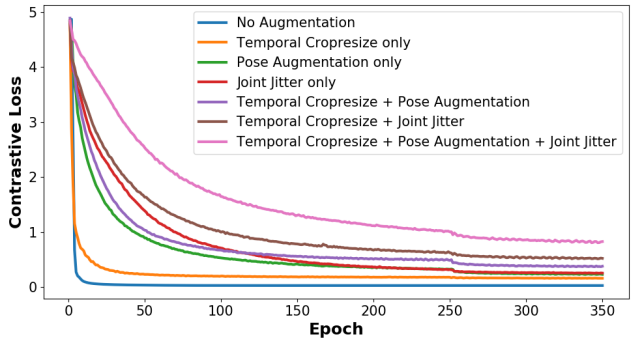

Figure 6: Skeleton augmentation loss curves. Our proposed spatial and temporal skeleton augmentations make the contrastive task more difficult which prevents early saturation of the loss. The network is forced to focus more on commonalities in pose and joint motion dynamics to learn the similarities.

The benefit of our proposed skeleton augmentations are also reflected in the contrastive pre-training plots in Figure 6, which demonstrate that without augmentation the contrastive task is too easy, resulting in early saturation of the loss and poor features. With our spatial and temporal augmentations the contrastive task becomes more difficult as the network is encouraged to focus more on the pose and spatio-temporal movements of the joints, thereby improving downstream accuracy. Thus the combination of all our augmentations result in learning our best 3D action features.

Intra-Skeleton *vs.* Inter-Skeleton. Next, we examine the effectiveness of learning two skeleton representations together in our inter-skeleton framework over learning from each input representation separately (intra-skeleton). While our inter-skeleton network pre-trains two input skeleton representations alongside one another, to allow for fair comparison to the intra-skeleton network we train and test the downstream action recognition model with each input representation separately. The results of combining multiple representations in downstream tasks are presented in supplementary.

Table 6 shows the accuracy of our inter-skeleton contrast compared to the intra-skeleton baseline for each skeleton representation.

	Downstream Representation		
Pretraining	X^{IMG}	X^{STG}	X^{SEQ}
Intra (X^{IMG} only)	79.6	-	-
Intra (X^{STG} only)	-	72.5	-
Intra (X^{SEQ} only)	-	-	82.5
Inter (X^{IMG}, X^{STG})	80.0	78.0	-
Inter (X^{IMG}, X^{SEQ})	**81.7**	-	83.0
Inter (X^{SEQ}, X^{STG})	-	78.9	**85.2**
Inter (X^{IMG}, X^{SEQ}, X^{STG})	**81.2**	**81.6**	**85.4**

Table 6: Intra-skeleton *vs.* Inter-skeleton. Training alongside a second input representation in our inter-skeleton contrast results in better features for all input representations, regardless of the pair used. Note that a representation can only be used in the downstream task when it is present in pre-training. Ablation performed on 3D action recognition with NTU RGB+D 60.

We first observe that pre-training with any two skeleton representations side by side in our inter-skeleton contrast is considerably better than only learning with a single representation as in the intra-skeleton contrast. For example, the accuracy with X^{STG} increases by 6% when pre-trained together with X^{SEQ} in our inter-skeleton contrast model. A similar increase of 5% occurs when pre-training alongside X^{IMG}. We find this to be the case with each skeleton representation; regardless of the second representation it is trained alongside in the inter-skeleton contrast, there is an increase in performance. We also tried training all three skeleton representations together. While this does give the best result, the improvement is outweighed by the computational cost of training all three representations simultaneously. Overall, these results reinforce our claim that learning invariance to skeleton augmentations alone leads to sub-optimal features and learning additional invariance to skeleton-representations results in a better feature space.

5 CONCLUSION

In this work, we presented a method for self-supervised learning of 3D skeleton data. We design a contrastive learning framework that relies on novel skeleton augmentations and multiple skeleton-representations to learn spatio-temporal dynamics of the skeleton sequences. Our comprehensive evaluation with different skeleton augmentations and skeleton-representation pairs reveal that learning invariance to our spatio-temporal augmentations and contrasting sequence-based and graph-based representations with each other results in best action features. The final model achieves considerable performance gains and outperforms prior state-of-the-art in self-supervised learning for multiple downstream tasks on NTU RGB+D 60 & 120 and PKU-MMD.

ACKNOWLEDGMENTS

This work is part of the research programme Perspectief EDL with project number P16-25 project 3, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering/ Sciences (TTW).

REFERENCES

[1] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. 2020. SpeedNet: Learning the Speediness in Videos. In *CVPR*.

[2] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

[4] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. 2020. Skeleton-Based Action Recognition with Shift Graph Convolutional Network. In *CVPR*.

[5] Liu Chunhui, Hu Yueyu, Li Yanghao, Song Sijie, and Liu Jiaying. 2017. PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding. In *ACM Multimedia workshop*.

[6] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. 2015. Unsupervised Visual Representation Learning by Context Prediction. In *ICCV*.

[7] Yong Du, Yun Fu, and Liang Wang. 2015. Skeleton based action recognition with convolutional neural network. In *ACPR*.

[8] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. 2017. Self-Supervised Video Representation Learning with Odd-One-Out Networks. In *CVPR*.

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*.

[10] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Memory-augmented Dense Predictive Coding for Video Representation Learning. In *ECCV*.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

[12] Alejandro Hernandez Ruiz, Lorenzo Porzi, Samuel Rota Bulò, and Francesc Moreno-Noguer. 2017. 3D CNNs on Distance Matrices for Human Action Recognition. In *ACM Multimedia*.

[13] Zhen Huang, Xu Shen, Xinmei Tian, Houqiang Li, Jianqiang Huang, and Xian-Sheng Hua. 2020. Spatio-Temporal Inception Graph Convolutional Networks for Skeleton-Based Action Recognition. In *ACM Multimedia*.

[14] Dahun Kim, Donghyeon Cho, and In So Kweon. 2019. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *AAAI*.

[15] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. 2011. HMDB: a large video database for human motion recognition. In *ICCV*.

[16] Zihang Lai, Erika Lu, and Weidi Xie. 2020. MAST: A Memory-Augmented Self-Supervised Tracker. In *CVPR*.

[17] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. 2018. Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. In *IJCAI*.

[18] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2019. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*.

[19] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu. 2020. MS2L: Multi-Task Self-Supervised Learning for Skeleton Based Action Recognition. In *ACM Multimedia*.

[20] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. 2020. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 10 (2020), 2684—2701.

[21] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. 2016. Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. In *ECCV*.

[22] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. 2017. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*.

[23] Mengyuan Liu, Hong Liu, and Chen Chen. 2017. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition* 68 (2017), 346–362.

[24] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. 2020. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. In *CVPR*.

[25] Fengjun Lv and Ramakant Nevatia. 2006. Recognition and Segmentation of 3-d Human Action Using HMM and Multi-Class Adaboost. In *ECCV*.

[26] Ishan Misra and Laurens van der Maaten. 2020. Self-Supervised Learning of Pretext-Invariant Representations. In *CVPR*.

[27] Qiang Nie, Ziwei Liu, and Yunhui Liu. 2020. Unsupervised 3D Human Pose Representation with Viewpoint and Pose Disentanglement. In *ECCV*.

[28] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised Learning of Visual Representions by solving Jigsaw Puzzles. In *ECCV*.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[30] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *CVPR*.

[31] Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. 2020. Mix dimension in poincaré geometry for 3d skeleton-based action recognition. In *ACM Multimedia*.

[32] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *CVPR*.

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*.

[34] Chenyang Si, Xuecheng Nie, Wei Wang, Liang Wang, Tieniu Tan, and Jiashi Feng. 2020. Adversarial Self-Supervised Learning for Semi-Supervised 3D Action Recognition. In *ECCV*.

[35] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In *AAAI*.

[36] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. 2020. Stronger, Faster and More Explainable: A Graph Convolutional Baseline for Skeleton-Based Action Recognition. In *ACM Multimedia*.

[37] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. 2021. Richly Activated Graph Convolutional Network for Robust Skeleton-based Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2021). In press.

[38] Tae Soo Kim and Austin Reiter. 2017. Interpretable 3d human action analysis with temporal convolutional networks. In *CVPR workshop*.

[39] Kun Su, Xiulong Liu, and Eli Shlizerman. 2020. Predict & cluster: Unsupervised skeleton based action recognition. In *CVPR*.

[40] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849* (2019).

[41] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In *CVPR*.

[42] Raviteja Vemulapalli and Rama Chellapa. 2016. Rolling Rotations for Recognizing Human Actions From 3D Skeletal Data. In *CVPR*.

[43] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. 2019. Self-supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*.

[44] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *AAAI*.

[45] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. 2020. Dynamic GCN: Context-enriched Topology Learning for Skeleton-based Action Recognition. In *ACM Multimedia*.

[46] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. 2017. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *ICCV*.

[47] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful Image Colorization. In *ECCV*.

[48] Richard Zhang, Phillip Isola, and Alexei A Efros. 2017. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. In *CVPR*.

[49] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. 2018. Unsupervised Representation Learning with Long-Term Dynamics for Skeleton Based Action Recognition. In *AAAI*.

Method	NTU RGB+D 60		NTU RGB+D 120	
	x-view	x-sub	x-setup	x-sub
PA-LSTM [32]	52.8	50.1	26.3	25.5
ST-LSTM [21]	77.7	69.2	57.9	55.7
GCA-LSTM [22]	84.0	76.1	59.2	58.3
VA-LSTM [46]	87.7	79.4	-	-
ST-GCN [44]	88.3	81.5	73.2	70.7
Shift-GCN [4]	96.5	90.7	85.9	87.6
MS-G3D Net [24]	96.2	91.5	86.9	88.4
This paper (supervised-only)	87.8	72.9	68.2	66.3
This paper (with-pretraining)	90.4	79.3	75.4	73.1

Table 7: Comparison with supervised only training for 3D action recognition. Pre-training with our inter-skeleton contrast improves the performance over supervised only training, especially for the more challenging cross-subject and cross-setup protocols.

Augmentation	Number of jittered joints $\|j\|$				
	2	5	10	15	20
Spatial-Jittering	65.6	67.5	69.4	**74.6**	70.6

Table 8: Effect of number of joints to jitter on the downstream task of 3D action classification on cross-view protocol of NTU RGB+D 60. Increasing the number of joints to jitter improves the downstream performance.

Augmentation	l_{min}		
	0.1	0.3	0.5
Temporal Crop-Resize	**62.5**	62.0	60.8

Table 9: Effect of temporal length ratio on the downstream task of 3D action classification on cross-view protocol of NTU RGB+D 60. The bigger the range, the better the downstream performance.

A APPENDIX

In this Appendix we provide details on the training procedure for each downstream task in Section A.1 and provide a comparison of our method to supervised-approaches for skeleton-based action recognition in Section A.2. We examine the effect of the hyperparameters of our proposed augmentations in Section A.3. Finally, we show the performance of combining multiple-skeleton representations for the downstream task of action recognition in Section A.4 and provide some qualitative results of our method in Section A.5.

A.1 Downstream Training Details

For the downstream tasks we follow Chen *et al.* [3] and remove the projection head of the pre-trained query encoder, as the projection head tends to focus mostly on information specific to the pretext task. For the 3D action recognition tasks, we then append a classifier to the pre-trained query encoder, while for 3D action retrieval we directly use the feature space without adding a classification head. The dimensionality of the feature space is dependent on the input skeleton-representation used in the downstream task. It is either 4096 (for X^{IMG}), 2048 (for X^{SEQ}) or 256 (for X^{STG}). For downstream tasks we use a temporal crop of length 64. During training this is sampled randomly, while for evaluation we sample a center crop.

3D Action Recognition. For this task, the weights of the pre-trained encoder are frozen and only the linear classifier is trained as in [19, 27]. An SGD optimizer is used with a momentum of 0.9 and learning rate of 0.1. The linear classifier is trained for a total of 80 epochs and learning rate is reduced by a factor of 10 after the 50th and 70th epoch.

3D Action Retrieval For this task we follow [39] to extract the encoder features of the training set. Then, we apply a kNN classifier with k=1 using these features and their corresponding action labels to assign action classes. Finally, during testing we assign to the unseen sample the action class of the closest neighbour in the training set.

Semi-Supervised 3D Action Recognition. For this task, we finetune both the classifier and the pre-trained encoder weights jointly as in [19]. An Adam optimizer is used to train the network for a total of 50 epochs with a learning rate of 0.0001, which is reduced by a factor of 10 after both the 30th and 40th epoch.

Transfer Learning for 3D Action Recognition. For this task we again follow [19] and finetune the classifier and the pre-trained encoder together. An Adam optimizer is used to train the network for a total of 50 epochs with a learning rate of 0.0001 which is reduced by a factor of 10 after 30th and 40th epoch.

A.2 Supervised Approaches

While our method outperforms prior self-supervised learning works for 3D action recognition, it is also useful to know how this compares to state-of-the-art supervised approaches. Table 7 shows the performance of various supervised approaches on the NTU 60 & 120 datasets. We compare these results to the performance of our sequence-based query encoder f^{SEQ} (a simple 3-layer Bi-GRU) trained end-to-end from randomly initialized weights (supervised-only) and finetuned end-to-end from the weights learnt from our inter-skeleton contrastive learning approach (with pre-training). Note that this setting is different to the experiment performed in the main paper, which only finetunes the final layer in order to demonstrate the raw performance of the features, rather than the boost they can provide to supervised training. It is evident from the table our method is competitive with many supervised approaches, even though the encoder we use is not state-of-the-art. It is also clear that our contrastive pre-training can boost the performance over supervised-only training. It is likely our inter-skeleton contrastive pre-training can also be used to boost the performance of more complex state-of-the-art encoders too.

A.3 Augmentation Hyperparameter Ablations

In this section we study the impact of hyperparameters $|j|$ and L_{ratio} of the spatial joint jittering and temporal crop-resize augmentations on the downstream performance. We use X^{IMG} skeleton representation and evaluate on the cross-view protocol of NTU RGB+D 60 for the downstream task of 3D action classification. We first pre-train an intra-contrastive framework using X^{IMG} representation with only the relevant augmentation and then train a

Downstream Reps.	Intra	Inter	# Inference Params.
X^{IMG}	79.6	81.7	1.0M
X^{STG}	72.5	78.9	3.0M
X^{SEQ}	82.5	**85.2**	10.0M
X^{IMG} + X^{STG}	80.3	81.8	4.0M
X^{IMG} + X^{SEQ}	80.3	82.6	11.0M
X^{SEQ} + X^{STG}	84.5	**86.0**	14.0M

Table 10: Combining representations for 3D action recognition. We show the trade-off between accuracy and number of parameters involved in the downstream task when using two representations to fine-tune the representations learnt from both intra and inter-skeleton pretraining. Pretraining with our inter-skeleton contrast learns better features for each representation whether used individually or combined.

linear classifier with action labels on top of the frozen features of the query encoder f_q.

A.3.1 Effect of number joints to jitter $|j|$. Here, we ablate over the number of joints to jitter $|j|$ in our joint jittering augmentation. This parameter controls the number of joints to be jittered for the augmented view. Table 8 shows the downstream 3D action classification performance of different values of $|j|$. We found that jittering around half the joints ($|j|$=10, 15) performed best. Using very small or a large values for $|j|$ *e.g.* 2 or 20 is sub-optimal as with too few jittered joints the augmented views become highly similar, while with many jittered joints there remains little commonality between the augmented sequences. For all our experiments we use $|j|$=15 in our joint jittering augmentation as it achieve best downstream performance.

A.3.2 Effect of temporal length ratio L_{ratio} . We next ablate over the distribution from which temporal length ratio $L_{ratio} \in [l_{min}, 1.0]$ is sampled in our temporal crop-resize augmentation, see Equation (3). The parameter l_{min} controls the minimum length of the temporal crop, which can be sampled for the augmented view. Table 9 shows the 3D action classification performance with different minimum samples lengths l_{min}. A smaller l_{min}, and thus a larger temporal range improves the downstream performance. We therefore use l_{min}=0.1, *i.e.*. $L_{ratio} \in [0.1, 1.0]$, for all our experiments.

A.4 Multi-representation Downstream

In this section, we examine the effect of combining skeleton representations when finetuning for the downstream task of 3D action recognition. All of our previous results use only one representation in the downstream task for efficiency, even when representations are trained together in our inter-skeleton contrast. Here we report the results of combining representations in the downstream task for both intra and inter-skeleton contrast. For intra-skeleton, each skeleton representation is first pretrained separately (see Section 3.2) and then their query encoders are combined for the downstream task. For inter-skeleton two skeleton representations are pretrained together (see Section 3.3) with their query encoders also combined for the downstream task. Table 10 shows the results of these experiments alongside the results when using only one representation during the downstream task from Table 6. We again evaluate on the cross-view protocol of NTU RGB+D 60 by training a linear classifier on frozen features. In Table 10 we also highlight the number of parameters needed for each representation in this downstream task. The downstream encoders (i.e query encoders) for the skeleton representations are as a 3-Layer BI-GRU with H=1024 units) for X^{SEQ}, an HCN [17] model for X^{IMG} and f^{STG} , a joint-based A-GCN [33] network for X^{STG} (see Implementation details Section 4.2).

From Table 10, we first observe when combining representations in the downstream task pretraining with inter-skeleton contrast outperforms the intra-skeleton pretraining for all combinations. In this setting both the computational costs required for inference and pretraining of the intra and inter-skeleton contrast are same as training two representations separately requires the same computation as training them together (inter), thereby showing the superiority of our inter-skeleton contrast.

As we saw in the main paper, inter-skeleton contrast shows considerable improvement in performance over the intra-skeleton contrast for the each single representation downstream evaluation, with X^{SEQ} obtaining the best results. However, it is worth noting that while the number of parameters required for inference are the same, the inter-skeleton does require additional computational resources for pretraining since each representation is required to be pre-trained with one of the other skeleton representations while in intra-skeleton each representation is pretrained alone.

We also observe that combining representations in the downstream task improves over using a single representation in the majority of cases, with the combination X^{SEQ} and X^{STG} showing the best results. Note that this improvement comes with an additional cost of model size during the inference time. With these results we can conclude that our model can be used for all skeleton representations individually or in combination based on the trade off between the pretraining computational cost, the inference model size and the performance.

A.5 Qualitative Results

A.5.1 Visualization of learned features. First we visualize the features by our inter-skeleton contrastive learning in comparison to those learned by Su *et al.* [39], one of the best performing methods on both the 3D action recognition and retrieval tasks. We randomly select 10 of the 60 action classes, so as not to overcrowd the figure, and plot their features using t-SNE. This is repeated three times for three different subsets of action classes. We observe from the Figure 7 that the features learned by our method form better clusters and are therefore more discriminatory and more suitable for the downstream tasks of action recognition and retrieval.

A.5.2 3D Action retrieval results. In Figure 8, we visualize the results of 3D action retrieval. For a given query video we retrieve the top four nearest neighbours in the feature space learned by Su *et al.* [39] and by our inter-skeleton contrastive learning. We observe from Figure 8 that the nearest neighbours in the feature space are generally more relevant to the query when using our method. The videos retrieved by Su *et al.* tend to be from different actions, which contain similar body poses. For instance 'kicking', 'staggering' and 'hop on one leg' all contain poses with one leg off the floor. Instead, our method is able to better focus on the motion of the query action and retrieve other instances of the same action.

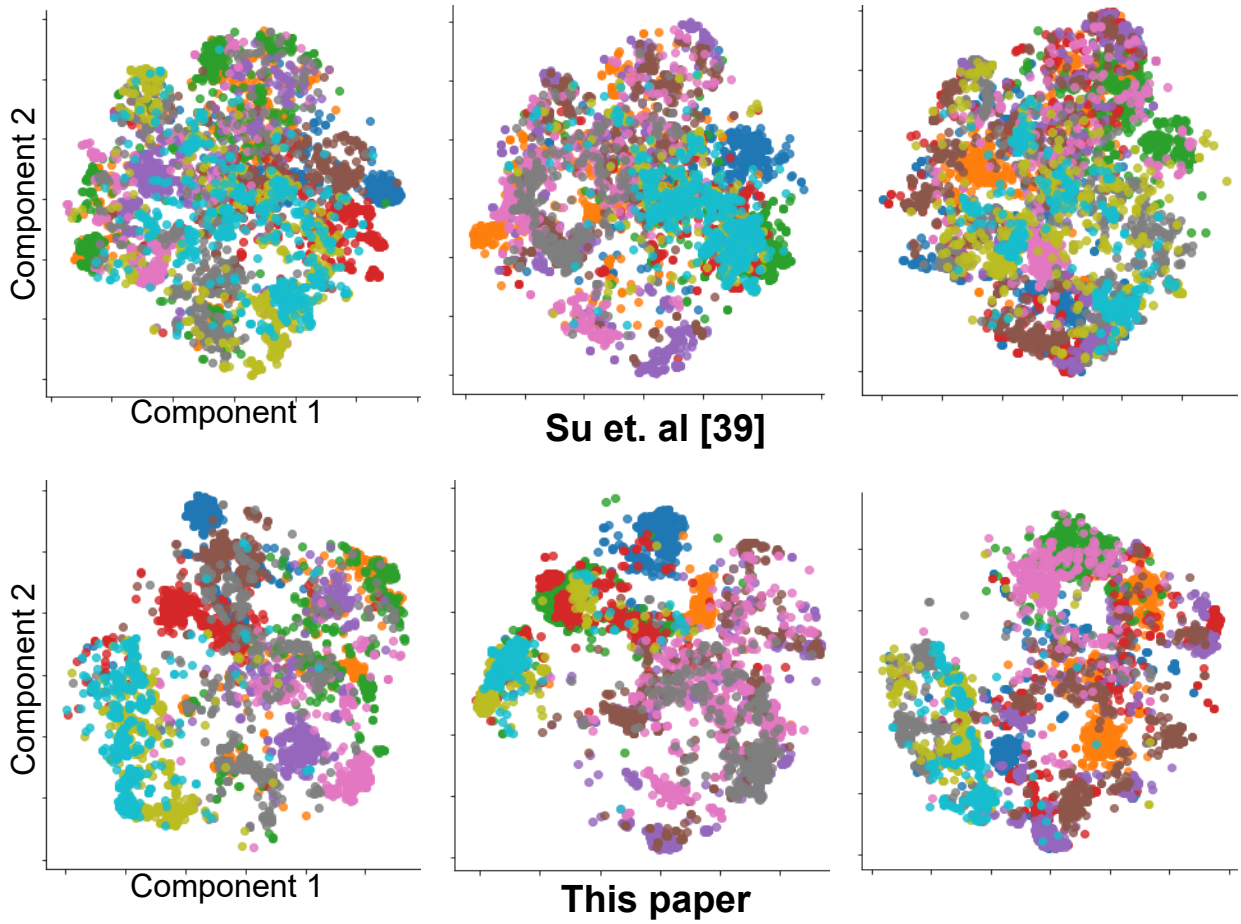

Figure 7: t-SNE visualization of learned features on NTU RGB+D 60 dataset. Each plot shows the features of 10 randomly selected action classes. Top row shows the features learned by Su *et al.* [39] and bottom row shows the corresponding features learned by our inter-skeleton contrastive learning. Our methods learns a more discriminatory feature space forming better clusters which are more dense with most samples from same action class and distant from other clusters as compared to [39].

Figure 8: 3D Action retrieval results on NTU RGB+D 60 dataset. For each query, the first row shows nearest neighbours learned by Su *et al.* [39] and the second row shows the nearest neighbours in the feature space learnt by our inter-skeleton contrastive learning. For our method most neighbours belong to the same action classes. All results were obtained using 3D skeleton data, however, for the ease of visualization/interpretation we show the corresponding RGB videos instead of the skeleton sequences.