PREDICT & CLUSTER: Unsupervised Skeleton Based Action Recognition

Kun Su[1], Xiulong Liu[1], and Eli Shlizerman[1, 2]

[1]Department of Electrical & Computer Engineering, University of Washington, Seattle, USA
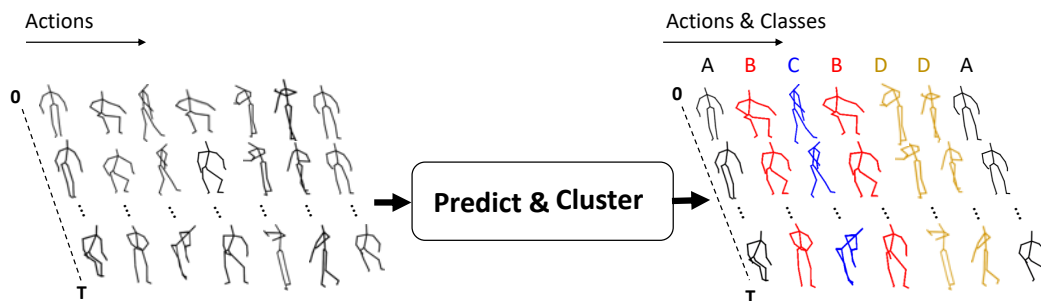[2]Department of Applied Mathematics, University of Washington, Seattle, USA

Figure 1: PREDICT & CLUSTER: Unsupervised action recognition from body keypoints. See additional examples in the supplementary video.

Abstract

We propose a novel system for unsupervised skeleton-based action recognition. Given inputs of body-keypoints sequences obtained during various movements, our system associates the sequences with actions. Our system is based on an encoder-decoder recurrent neural network, where the encoder learns a separable feature representation within its hidden states formed by training the model to perform the prediction task. We show that according to such unsupervised training, the decoder and the encoder self-organize their hidden states into a feature space which clusters similar movements into the same cluster and distinct movements into distant clusters. Current state-of-the-art methods for action recognition are strongly supervised, i.e., rely on providing labels for training. Unsupervised methods have been proposed, however, they require camera and depth inputs (RGB+D) at each time step. In contrast, our system is fully unsupervised, does not require action labels at any stage and can operate with body keypoints input only. Furthermore, the method can perform on various dimensions of body-keypoints (2D or 3D) and can include additional cues describing movements. We evaluate our system on three action recognition benchmarks with different numbers of actions and examples. Our results outperform prior unsupervised skeleton-based methods, unsupervised RGB+D based methods on cross-view tests and while being unsupervised have similar performance to supervised skeleton-based action recognition.

1. Introduction

Robust action recognition, especially human action recognition, is a fundamental capability in ubiquitous computer vision and artificial intelligence systems. While recent methods have shown remarkable success rates in recognizing basic actions in videos, current methods rely on strong supervision with large number of training examples accompanied with action labels. Collection and annotation of large scale datasets is implausible for various types of actions and applications. Furthermore, annotation is a challenging problem by itself since it is often up to the interpretation of the annotator to assign a meaningful label for a given sequence. This is particularly the case in situations where it is unclear what is the ground truth label, e.g., annotation of animal movements. Indeed, annotations challenges are common in different contextual information sources of movement, such as video (RGB), depth (+D) and keypoints tracked over time. On the one hand, keypoints include much less information compared to RGB+D data and can be challenging to work with. On the other hand, however, focusing on keypoints can often isolate the actions and provide more robust unique features for them. For human action recognition, time-series of body joints (skele-

ton) tracked over time are indeed known as effective descriptors of actions. Here we focus on 3D skeleton timesequences and propose an unsupervised system to learn features and assign actions to classes according to them. We call our system PREDICT & CLUSTER (P&C) since it is based on training an encoder-decoder type network to simultaneously predict and cluster skeleton sequences such that the network learns an effective hidden feature representation of the actions. Indeed, an intuitive replacement of a classification supervised task by a non-classification unsupervised task is to attempt to continue (predict) or reproduce (re-generate) the given sequence such that it leads the hidden states to capture key features of the actions. In the encoder-decoder architecture, the prediction task is typically implemented as follows: given an action sequence as the encoder input, the decoder continues or generates the encoder input sequence. Since inputs are sequences, both the decoder and the encoder are recurrent neural networks (RNN) containing cells with hidden variables for each time sample in a sequence. The final hidden state of the encoder is typically being utilized to represent the action feature. While the encoder contains the final action feature, since the gradient during training flows back from the decoder to the encoder, it turns out that the decoder training strategies significantly determine the effectiveness of the representation. Specifically, there are two types of decoder training strategies proposed for such prediction/re-generation task [26]. The first strategy is a conditional strategy, where the output of the previous time-step of the decoder is used as the input to the current time-step. With such strategy, the output of the decoder is expected to be continuous. In contrast, the unconditional strategy assigns a zero input into each time-step of the decoder. Previous work showed that unconditional training of the decoder is expected to have better prediction performance since it effectively weakens the decoder and thus forces the encoder to learn a more informative representation.

In our system, we extend such strategies to enhance the encoder representation. This results in enhanced clustering and organization of actions in the feature space. In particular, we propose two decoder training strategies, Fixed Weights and Fixed States to further penalize the decoder. The implementation of these strategies guides the encoder to further learn the feature representation of the sequences that it processes. In fact, in both strategies, the decoder is a 'weak decoder', i.e., the decoder is effectively not being optimized and serves the role of propagating the gradient to the encoder to further optimize its final state. Combining these two strategies together we find that the network can learn a robust representation and our results show that this strategy can achieve significantly enhanced performance than unsupervised approaches trained without them. We demonstrate the effectiveness and the generality of our proposed methodology by evaluating our system on three extensive skeleton-based and RGB+D action recognition datasets. Specifically, we show that our P&C unsupervised system achieves high accuracy performance and outperforms prior methods.

2. Related Work

The objective of action recognition is to assign a class label to a sequence of frames with context information on the action performed, Fig. 1. Numerous approaches have been introduced particularly for human movement action recognition. Such approaches use video frames (RGB), and/or depth (RGB+D) and/or skeleton data, i.e. tracking of body joints (keypoints). Performing exclusive skeleton-based action recognition is especially advantageous since requires much less data, it is relatively easy to acquire and therefore has the potential to be performed in real-time. Furthermore, skeleton data can be used to understand the exclusive features of the actions in contrast to videos and depth that include unrelated features such as background. Indeed, in recent years, various supervised and unsupervised approaches have been introduced for human skeletonbased action recognition. Most of the skeleton-based approaches have been supervised where an annotated set of actions and labels should be provided for training. In an unsupervised setup, the problem of action recognition is much more challenging. Only a few unsupervised skeleton-based approaches have been proposed and several unsupervised approaches have been developed to use more information such as video frames and depth, i.e., unsupervised RGB+D. We review these prior approaches below and compare our results with them.

For *supervised* skeleton-based action recognition, prior to deep learning methods, classical approaches were proposed to map the actions from Lie group to its Lie algebra and to perform classification using a combination of dynamic time warping, Fourier temporal pyramid representation and linear SVM (e.g. LARP [28]). Deep learning approaches have been developed to classify skeleton data as well, in particular, models based on RNN designed to work with sequences. For example, Du et al. [3] used hierarchical RNN (HBRNN-L) for action classification and Shahroudy et al. [21] proposed the Part-aware LSTM (P-LSTM) as the baseline for large scale skeleton action recognition with NTU RGB+D dataset. Since skeleton data is noisy, largely due to variance in camera views, prior works proposed a view-adaptive RNN (VA-RNN) which learns a transformation from original skeleton data to a general pose [35]. CNN based approaches for supervised skeleton based recognition have been proposed as well. Such approaches require a representation of body joints that can be processed by CNN. Spatio-temporal skeleton sequence can be represented as a matrix in which each column vector rep-
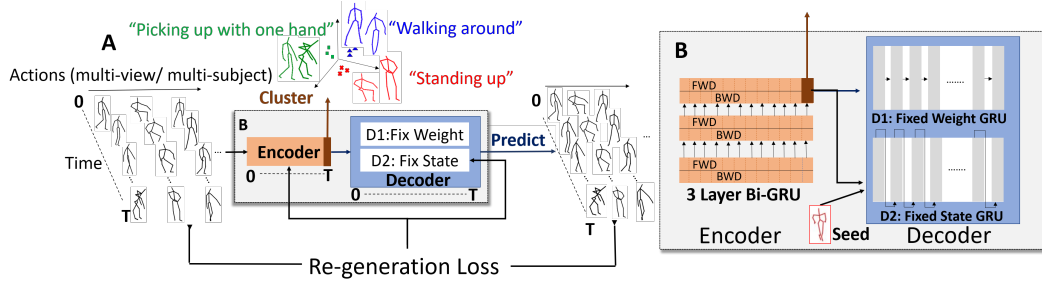
Figure 2: PREDICT & CLUSTER system summary. **A:** System overview. **B:** Encoder-Decoder architecture.

resents an instant in time and elements of the vector are the coordinates of the joints. Such a matrix can be then converted to an image on which CNN is trained for classification [2]. In addition, enhanced skeleton visualization methods have been developed in conjunction with CNN classification for view invariant human action recognition [13]. Recently, Graph Neural Networks (GNN) gained popularity in skeleton-based action recognition. The data structure in these networks is a graph which corresponds to the human skeleton, i.e., vertices correspond to joints and edges to bones. With the graph structure, the network learns representations constrained by the human skeleton anatomy. In the supervised action recognition setup, GNNs are implemented using Graph Convolutional Networks (GCN). In particular, Spatio-Temporal GCN was designed to learn both spatial and temporal patterns from skeleton data [33]. A recent extension of the approach showed that directed GCN with two streams can encode skeleton representation and adaptively learn the graph [22, 23]. Furthermore, Actional-Structural GCN which incorporates a *prediction* module was proposed to enhance action recognition [11].

While recent supervised approaches show robust performance on action recognition, the *unsupervised* setup is advantageous since it does not require labeling of sequences and may not require re-training when additional actions not included in the initial training set are introduced. Unsupervised methods typically aim to obtain an effective feature representations by predicting future frames of input action sequences or by re-generating the sequences. Unsupervised approaches were mostly proposed for videos of actions or videos with additional information such as depth or optical flow. Specifically, Srivastava et al. [26] proposed a recurrent-based sequence to sequence (Seq2Seq) model as an autoencoder to learn the representation of a video. Such an approach is at the core of our method for body joints input data. However, as we show, the approach will not be able to achieve efficient performance without particular training strategies that we develop to weaken the decoder and strengthen the encoder. Subsequently, Luo et al. [14] developed a convolutional LSTM to use depth and optical

flow information such that the network encodes depth input and uses the decoder to predict the optical flow of future frames. Furthermore, Li et al. [10] proposed to employ a generative adversarial network (GAN) with a camera-view discriminator to assist the encoder in learning better representations.

As in unsupervised RGB+D approaches, skeleton-based approaches utilize the task of human motion prediction as the underlying task to learn action feature representation. For such a task, RNN-based Seq2Seq models [15] were shown to achieve improved accuracy in comparison to non-Seq2Seq RNN models such as ERD [5] and S-RNN [7]. GNN turned out to be effective network architecture as well, e.g., unsupervised GNN was shown to effectively learn the dynamics in real motion capture and sports tracking data [8]. Recently, networks incorporating GANs have achieved improved performance on this task by utilizing the predictor network being RNN Seq2Seq and the discriminator [6].

Unsupervised approaches for *skeleton-based action recognition* are scarce since obtaining effective feature representations from coordinate positions of body joints is challenging. Indeed, prior non-deep work have shown that representations play a significant role in modeling and predicting 3D human skeleton movements [34, 18, 19, 1]. Recently, deep approaches have been proposed for human motion prediction. In particular, Zheng et al. [36] (LongT GAN) proposed a GAN encoder-decoder such that the decoder attempts to re-generate the input sequence and the discriminator is used to discriminate whether the re-generation is accurate. The feature representation used for action recognition is taken from the final state of the encoder hidden representation. During training, masked ground truth input is provided to the decoder. The method was tested on motion-capture databases, e.g., CMU Mocap, HDM05[17] and Berkeley MHAD[18]. Such datasets were captured by physical sensors (markers) and thus are much cleaner than marker-less data collected by depth cameras and do not test for multi-view variance which significantly affects action recognition performance. Our baseline network architecture is similar to the structure of LongT GAN since we use
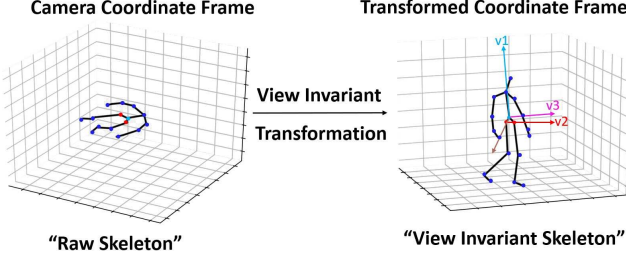
Figure 3: Pre-processing of body keypoints sequences according to view-invariant transformation.
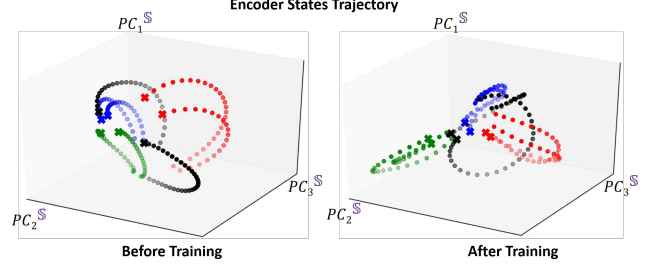

Figure 4: Encoder states trajectories visualized by projection to 3 PCA space. Each color represents one type of action (blue: donning, red: sit down, green: carry, black: stand up). The cross symbol denotes the final state. Left: before training; Right: after training.

an encoder and decoder and we also use the final state of the encoder as the feature representation of the action sequences. However, as we show, it is required to develop extended training strategies for the system to be applicable to larger scale multi-view and multi-subject datasets. Specifically, instead of using the masked ground truth as an input into the decoder, we propose methods to improve learning of the encoder and to weaken the decoder.

3. Methods

Pre-processing of body keypoints: Body keypoints data is a sequence X_T^V of T frames captured from a particular view, where each frame represents $N = 3$D coordinates of J joint keypoints

$$X_T^V = \{x_1, x_2, \ldots, x_T\}, x_t \in \mathbf{R}^{J \times N}.$$

Action sequences are captured from different views by depth cameras, such as Microsoft Kinect. 3D human joint positions are extracted from a single depth image by a real-time human skeleton tracking framework [24]. We align the action sequences by implementing a view-invariant transformation which transforms keypoints coordinates from original coordinate system into a view-invariant coordinate system $X^V \rightarrow X$. The transformed skeleton joint coordinates are given by

$$x_t^j = R^{-1}(x_t^j - d_R), \forall j \in J, \forall t \in T,$$

where $x_t^j \in \mathbf{R}^{3 \times 1}$ are the coordinates of the j-th joint of the t-th frame, R is the rotation matrix and d_R is the origin of rotation. These are computed according to

$$R = \left[\frac{v_1}{\|v_1\|} \,\middle|\, \frac{\hat{v}_2}{\|\hat{v}_2\|} \,\middle|\, \frac{v_1 \times \hat{v}_2}{\|v_1 \times \hat{v}_2\|} \right], d_R = x_{t=0}^{\text{root}},$$

where $v_1 = x_{t=0}^{\text{spine}} - x_{t=0}^{\text{root}}$ is the vector perpendicular to the ground, $v_2 = x_{t=0}^{\text{hip left}} - x_{t=0}^{\text{hip right}}$ is the difference vector between left and right hips joints in the initial frame of each sequence and $\hat{v}_2 = \frac{v_2 - \text{Proj}_{v_1}(v_2)}{\|v_2 - \text{Proj}_{v_1}(v_2)\|}$. $\text{Proj}_{v_1}(v_2)$ and $v_1 \times \hat{v}_2$

denote the vector projection of v_2 onto v_1 and the cross product of v_1 and $\hat{v}_2$, respectively. $x_{t=0}^{\text{Root}}$ is the coordinate of the root joint in the initial frame [9] (see Fig. 3). Since actions can be of different lengths we down-sample each action sequence to be of at most of a fixed length T_{max} and pad with zeros if the sequence length is smaller than that.

Self-organization of hidden states clustering: A key property that we utilize in our system is the recent observation that propagation of input sequences through RNN self-organizes them into clusters within the hidden states of the network, i.e., clusters represent features in an embedding of the hidden states [4]. Such strategy is a promising unsupervised method for multi-dimensional sequence clustering such as body keypoints sequences [27]. As we show, self-organization is inherent to any RNN architecture and even holds for random RNN which are initialized with random weights and kept fixed, i.e., no training is performed. Indeed, when we input sequences of body keypoints of different actions into random RNN, the features in the hidden state space turn out to be effective filters. While such strategy is promising, the recognition accuracy outcome appears to be non-optimal (Table 1 P&C Rand). We therefore implement an encoder-decoder system, which we call PREDICT & CLUSTER (P&C), where the encoder propagates input sequences and passes the last hidden state to the decoder. The decoder is used to regenerate the encoder input sequences. Furthermore, we utilize the random network setup (which does not require training) to choose the optimal hyper-parameters for the network to be trained. We describe the components of P&C below.

Motion prediction: At the core of our unsupervised method is an encoder-decoder RNN (Seq2Seq). Such network models were shown to be effective in prediction of future evolution of multi-dimensional time-series of features including skeleton temporal data of various actions [15, 6]. In these applications the typical flow in the network is unidirectional. The *encoder* processes an initial sequence of

activity and passes the last state to the *decoder* which in turn, based on this state generates the evolution forward. We extend such network structure for our method (see system overview in Fig. 2).

We propose a *bi-directional* flow such that the network can capture better long-term dependencies in the action sequences. Specifically, the encoder is a multi-layered bi-directional Gated Recurrent Unit (GRU) which input is a whole sequence of body keypoints corresponding to an action. We denote the forward and backward directions hidden states of the last layer of encoder at time t as $\overrightarrow{E_t}$ and $\overleftarrow{E_t}$ respectively, and the final state of the encoder as their concatenation $E_T = \{\overrightarrow{E_T}, \overleftarrow{E_T}\}$. The decoder is a uni-directional GRU with hidden states at time t denoted as D_t. The final state of the encoder is fed into the decoder as its initial state, i.e., $D_0 = E_T$. In such a setup, the decoder generates a sequence based on E_T initialization. In a typical prediction task, the generated sequence will be compared with forward evolution of the same sequence (prediction loss). In our system, since our goal is to perform action recognition, the decoder is required to re-generate the whole input sequence (re-generation loss). Specifically, for the decoder outputs $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T\}$ the re-generation loss function is the error between X and $\hat{X}$. In particular, we use mean square error (MSE) $L = \frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2$ or mean absolute error (MAE) $L = \frac{1}{T}\sum_{t=1}^{T}|x_t - \hat{x}_t|$ as plausible losses.

Hyper-parameter search: As in any deep learning system, hyper-parameters significantly impact network performance and require tuning for optimal regime. We utilize the self-organization feature of random initialized RNN to propagate the sequences through the network and use network performance prior to training as an optimization for hyper-parameter tuning. Specifically, we evaluate the capacity of the encoder by propagating the skeleton sequence through the encoder and evaluate the performance of recognition on the final encoder state. We observe that this efficient hyper-parameter search significantly reduces total training time when an optimal network amenable for training is being selected.

Training: With optimal hyper-parameter encoder being set, training is performed on the outputs of the decoder to predict (re-generate) the encoder's input action sequence. Training for prediction is typically performed according to one of the two approaches: (i) *unconditional* training in which zeros are being fed into the decoder at each time step or (ii) *conditional* in which an initial input is fed into the first time-step of the decoder and subsequent time-steps use the predicted output of the previous time step as their input [26]. Based on these training strategies, we propose two decoder configurations *(i) Fixed Weights decoder (FW)* or *(ii) Fixed States decoder (FS)* to weaken the decoder, i.e. to force it to perform the re-generation based upon the infor-
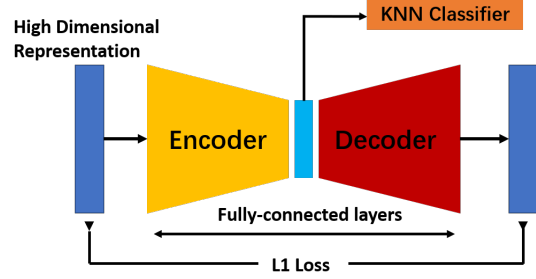

Figure 5: Feature-level autoencoder and KNN Classifier

mation provided by the hidden representation of the encoder and thus improve the encoder's clustering performance, see Fig. 4.

1.Fixed Weights decoder (**FW**): The input into the decoder is unconditional in this configuration. The decoder is not expected to learn a useful information for prediction and it exclusively relies on the state passed by the encoder. The weights of the decoder can thereby be assigned as random and the decoder is used as a recurrent propagator of the sequences. In training for the re-generation loss such configuration is expected to *force the encoder to learn the latent features* and represent them with the final state passed to the decoder. This intuitive method turns out to be computationally efficient since only the encoder is being trained and our results indicate favorable performance in conjunction with KNN action classification.

2.Fixed States decoder (**FS**): The external input into the decoder is conditional in this configuration (external input into each time-step is the output of the previous time-step), however the internal input, typically the hidden state from previous step, is replaced by the final state of the encoder E_T. Namely, in RNN cell

$$h_t = \sigma(W_x x_t + W_h g_t + b_h), g_t = h_{t-1} \to E_T,$$
$$y_t = \sigma(W_y h_t + b_y),$$
$$x_{t+1} = y_t,$$

with x_t the external input, y_t the output and h_t the hidden state at time-step t, h_{t-1} terms are replaced by E_T. In addition, we also add residual connection between external input and output, which has been shown useful in human motion prediction [15]. The final output and next input will be $\hat{y}_t = y_t + x_t$ and $\hat{x}_{t+1} = \hat{y}_t$, respectively. The configuration forces the network to rely on E_T, instead of the hidden state at previous time-step and eliminates vanishing of the gradient since during back-propagation at each time-step there is a defined gradient back to the final encoder state.

Feature level auto-encoder: After training the prediction network we extract the final encoder state E_T as the feature vector associated with each action sequence. Since the feature vector is high-dimensional, we use a feature-level

auto-encoder that learns the core low dimensional components of the high-dimensional feature so it can be utilized for classification (Fig. 5). Specifically, we implement the auto-encoder, denoted as f to be of an encoder-decoder architecture with parameters θ such that

$$\hat{E}_T = f_\theta(E_T) \approx E_T.$$

The encoder and the decoder are multi-layer FC networks with non-linear $\tanh$ activation and we implement the following loss $l_{aec} = \left| E_T - \hat{E}_T \right|$.

K-nearest neighbors classifier: For evaluation of our method on action recognition task we use a K-nearest neighbors (KNN) classifier on the middle layer of the auto-encoder feature vector. Specifically, we apply the KNN classifier (with $k = 1$) on the features of the trained network on all sequences in the training set to assign classes. We then use cosine similarity as the distance metric to perform recognition, i.e., place each tested sequence in a class. Notably, KNN classifier does not require to learn extra weights for action placement.

4. Experimental Results and Datasets

Implementation details: To train the network, all body keypoints sequences are pre-processed according to the view-invariant transformation and down-sampled to have at most 50 frames (Fig. 3). The coordinates are also normalized to the range of $[-1, 1]$. Using the hyper-parameter search, employing random RNN propagation discussed above, we set the following architecture: *Encoder*: 3-Layer Bi-GRU with $N = 1024$ units in each layer. *Decoder*: 1-Layer Uni-GRU with $N = 2048$ units such that it is compatible with the dimensions of the encoder final state E_T. All GRUs are initialized with a random uniform distribution. *Feature-level auto-encoder:* 6 FC Layers with the following dimensions: input feature vector(dim= 2048) $\rightarrow$ FC(1024) $\rightarrow$ FC(512) $\rightarrow$ FC(256)$\rightarrow$ FC(512)$\rightarrow$FC(1024) $\rightarrow$FC(2048). All FCs use $\tanh$ activation except the last layer which is linear. The middle layer of the auto-encoder outputs a vector feature of 256 elements which is used as the final feature. We use Adam optimizer and learning rate starting from 10^{-4} and 0.95 decay rate at every 1000 iterations. The gradients are clipped if the norm is greater than 25 to avoid gradient explosion. It takes 0.7sec per training iteration and 0.1sec to forward propagate with a batch size of 64 on one Nvidia Titan X GPU. Please see additional details of architectural choices in the supplementary material. **Datasets**: We use three different datasets for training, evaluation and comparison of our P&C system with related approaches. The three data-sets include various number of classes, types of actions, body keypoints captured from different views and on different subjects. In these datasets, the body keypoints are captured by depth cameras and also include additional data, e.g., videos (RGB) and depth (+D). Various types of action recognition approaches have been applied to these datasets, e.g., supervised skeleton approaches and unsupervised RGB+D approaches. We list these types of approaches and their performance on the tests in the datasets in Table 1. Notably, as far as we know, our work is the **first fully unsupervised skeleton based approach applied on these extensive action recognition tests**.

The datasets that we have applied our P&C system to are *(i) NW-UCLA*, *(ii) UWA3D*, and *(iii) NTU RGB+D*. The datasets include 3D body keypoints of $10, 30, 60$ action classes respectively. We briefly describe them below. **North-Western UCLA (NW-UCLA)** dataset [30] is captured by Kinect v1 and contains 1494 videos of 10 actions. These actions are performed by 10 subjects repeated 1 to 6 times. There are three views of each action and for each subject 20 joints are being recorded. We follow [13] and [30] to use the first two views (V1,V2) for training and last views (V3) to test cross view action recognition. **UWA3D Multiview Activity II (UWA3D)** dataset [20] contains 30 human actions performed 4 times by 10 subjects. 15 joints are being recorded and each action is observed from four views: frontal, left and right sides, and top. The dataset is challenging due to many views and the resulting self-occlusions from considering only parts of them. In addition, there is high similarity among actions, e.g., the two actions "drinking" and "phone answering" have many keypoints being near identical & idle and in the dynamic keypoints there are subtle differences. **NTU RGB+D** dataset [21] is a large scale dataset for 3D human activity analysis. This dataset consists of 56880 video samples, captured from 40 different human subjects, using Microsoft Kinect v2. NTU RGB+D(60) contains 60 action classes. We use the 3D skeleton data for our experiments such that each time-sample contains 25 joints. We test our P&C method on both cross-view and cross-subject protocols.

5. Evaluation and Comparison

Evaluation: In all experiments, we use the K-nearest neighbors classifier with $k = 1$ to compute the action recognition accuracy and evaluate the performance of our P&C method. We test different variants of P&C architectures (combinations of components described in Section 3) and report a subset of these in the paper: *baseline* random initialized encoder with no training (**P&C-Rand**), full system with FS decoder and feature-level auto-encoder (**P&C-FS-AEC**) and full system with FW decoder and feature-level auto-encoder (**P&C-FW-AEC**). We report the rest of the combinations and their results in the Supplementary material.

Fig. 6 shows the optimization of the regeneration loss (red) and the resulting accuracy (blue) during training for
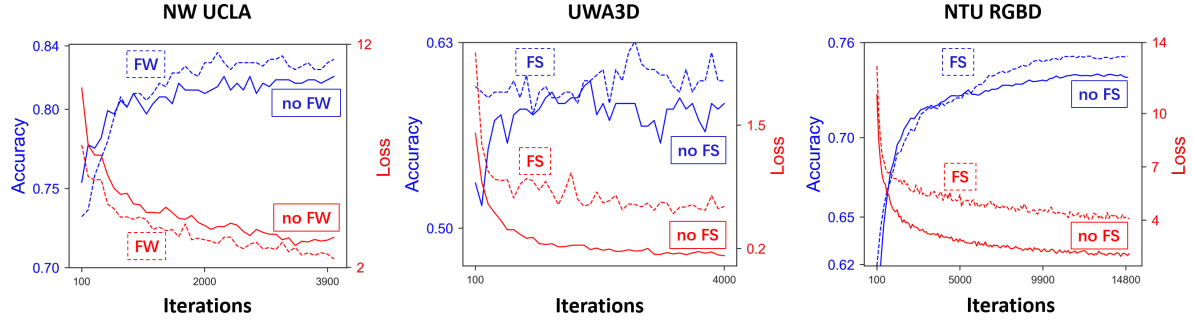
Figure 6: Training curves (accuracy:blue; loss:red) for three of the datasets, left to right: NW-UCLA (FW v.s. no FW), UWA3D (FS v.s. no FS), NTU-RGB+D Cross-View (FS v.s. no FS).

Method	NW-UCLA (%)
Supervised Skeleton	
HOPC[20]	74.2
Actionlet Ens [29]	76.0
HBRNN-L[3]	78.5
VA-RNN-Aug[35]	**90.7**
AGC-LSTM[25]	**93.3**
Unsupervised RGB+D	
Luo et al.[14]	**50.7**
Li et al.[10]	**62.5**
Unsupervised Skeleton	
P&C Rand (**Our**)	72.0
LongT GAN [36]	74.3
P&C FS-AEC (**Our**)	**83.8**
P&C FW-AEC (**Our**)	**84.9**

Method	UWA3D	
	V3 (%)	V4 (%)
Supervised Skeleton		
HOJ3D[32]	15.3	28.2
2-layer P-LSTM[31]	27.6	24.3
IndRNN (6 layers)[31]	30.7	47.2
IndRNN (4 layers)[31]	34.3	54.8
ST-GCN[31]	36.4	26.2
Actionlet Ens[29]	45.0	40.4
LARP[28]	49.4	42.8
HOPC[20]	**52.7**	**51.8**
VA-RNN-Aug[35]	**70.9**	**73.2**
Unsupervised Skeleton		
P&C Rand (**Our**)	48.5	51.5
LongT GAN [36]	53.4	59.9
P&C FS-AEC (**Our**)	**59.5**	**63.1**
P&C FW-AEC (**Our**)	**59.9**	**63.1**

Method	NTU RGB-D 60	
	C-View (%)	C-Subject (%)
Supervised Skeleton		
HOPC[20]	52.8	50.1
HBRNN[3]	64.0	59.1
2L P-LSTM[21]	70.3	62.9
ST-LSTM[12]	**77.7**	**69.2**
VA-RNN-Aug[35]	**87.6**	**79.4**
Unsupervised RGB+D		
Shuffle & learn[16]	40.9	46.2
Luo et al.[14]	**53.2**	**61.4**
Li et al.[10]	**63.9**	**68.1**
Unsupervised Skeleton		
LongT GAN [36]	48.1	39.1
P&C Rand (**Our**)	56.4	39.6
P&C FS-AEC (**Our**)	**76.3**	**50.6**
P&C FW-AEC (**Our**)	**76.1**	**50.7**

Table 1: Comparison of action recognition performance of our P&C system with state-of-the-art approaches of *Supervised Skeleton* (blue) and *Unsupervised RGB+D* (purple); *Unsupervised Skeleton* (red)) types.
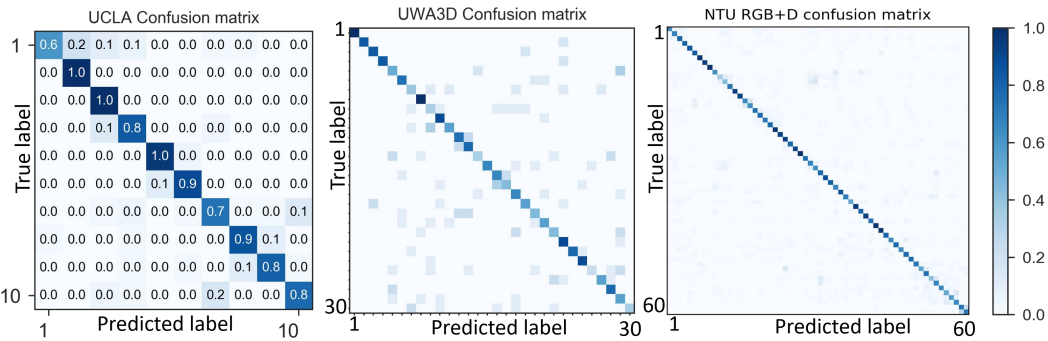

Figure 7: Confusion matrices for testing P&C performance on the three datasets (from left to right): NW-UCLA (10 actions); UWA3D V4 (30 actions); NTU-RGBD Cross-View (60 actions).
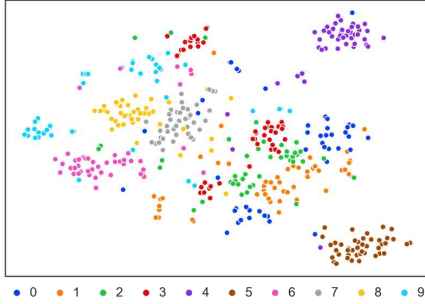
Figure 8: t-SNE visualization of learned features on NW-UCLA dataset.

each dataset. We include plots of additional P&C configurations in the Supplementary material. The initial accuracy appears to be substantial and this is attributed to the hyper-parameter search being performed on random initialized networks prior to training that we describe in Section 3. Indeed, we find that using appropriate initialization, the encoder, without any training, effectively directs similar action sequences to similar final states. Training enhances this performance further in both P&C FW and P&C FS configurations. Over multiple training iterations both P&C FW and P&C FS achieve higher accuracy than no-FW and no-FS in all datasets. While the convergence of the loss curve indicates improvement of the accuracy, the value of the loss does not necessarily indicate a better accuracy as can be observed from loss and accuracy curves of training on UWA3D and NTU-RGB+D (Fig. 6 middle, right).

We show the confusion matrices for the three considered datasets in Fig. 7. In NW-UCLA (with least classes) we show the elements of the 10x10 matrix. Our method achieves high-accuracy ($> 83\%$) on average and there are three actions (pick up with two hands, drop trash, sit down) for which it recognizes them with nearly 100% accuracy. We also show in Fig. 8 a t-SNE visualization of the learned features for NW-UCLA test. Even in this 2D embedding it is clearly evident that the features for each class are well separated. As more action classes are considered, the recognition becomes a more difficult task and also depends on the amount of training data. For example, while NTU RGB+D has more classes than UWA3D, the recognition accuracy on NTU RGB+D is smoother and results with better performance since it has 40 times more data than UWA3D. Our results show that our method is compatible with varying data sizes and number of classes.

Comparison: We compare the performance of our P&C method with prior related supervised and unsupervised methods applied to (left-to-right): NW-UCLA, UWA3D, NTU RGB+D datasets, see Table 1. In particular, we compare action recognition accuracy with approaches based on supervised skeleton data (blue), unsupervised RGB+D data

(purple) and unsupervised skeleton data (red). For comparison with unsupervised skeleton methods, we implement and reproduce the LongTerm GAN model (LongT GAN) as introduced in [36] and list its performance.

For NW-UCLA, P&C outperforms previous unsupervised methods (both RGB+D and skeleton based). Our method even outperforms the first three supervised methods listed in Table 1-left. UWA3D is considered a challenging test for many deep learning approaches since the number of sequences is small, while it includes a large number of classes (30). Indeed, action recognition performance of many supervised skeleton approaches is low ($< 50\%$). For such datasets, it appears that the unsupervised approach could be more favorable, i.e., even P&C Rand reaches performance of $\approx 50\%$. LongT GAN achieves slightly higher performance than P&C Rand, however, not as high as P&C FS/FW-AEC which perform with $\approx 60\%$. Only a single supervised skeleton method, VA-RNN-Aug, is able to perform better than our unsupervised approach, see Table 1-middle. On the large scale NTU-RGB+D dataset, our method performs extremely well on the cross-view test. It outperforms prior unsupervised methods (both RGB+D and skeleton based) and on-par with ST-LSTM (second best supervised skeleton method), see Table 1-right. On the cross-subject test we obtain performance that is higher (including P&C Rand) than the prior unsupervised skeleton approach, however, our accuracy does not outperform unsupervised RGB+D approaches. We believe that the reason stems from skeleton based approaches not performing well in general on cross-subject tests since additional aspects such as subjects parameters, e.g., skeleton geometry and invariant normalization from subject to subject, need to be taken into account.

In summary, for all three datasets, we used a single architecture and it was able to outperform the prior unsupervised skeleton method, LongT-GAN[36], most supervised skeleton methods and unsupervised RGB+D methods on cross view tests and some supervised skeleton and unsupervised RGB+D on large scale cross subject test.

6. Conclusion

We presented a novel unsupervised model for human skeleton-based action recognition. Our system reaches enhanced performance compared to prior approaches due to novel training strategies which weaken the decoder and strengthen the training of the encoder. As a result, the network learns more separable representations. Experimental results demonstrate that our unsupervised model can effectively learn distinctive action features on three benchmark datasets and outperform prior unsupervised methods.

References

[1] Amor Ben Tanfous, Hassen Drira, and Boulbaba Ben Amor. Coding kendall's shape trajectories for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2840–2849, 2018. 3

[2] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 579–583. IEEE, 2015. 3

[3] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015. 2, 7

[4] Matthew Farrell, Stefano Recanatesi, Guillaume Lajoie, and Eric Shea-Brown. Recurrent neural networks learn robust representations by dynamically balancing compression and expansion. 2019. 4

[5] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 3

[6] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018. 3, 4

[7] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016. 3

[8] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018. 3

[9] Inwoong Lee, Doyoung Kim, Seoungyoon Kang, and Sanghoon Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1012–1020, 2017. 4

[10] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In *Advances in Neural Information Processing Systems*, pages 1254–1264, 2018. 3, 7

[11] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019. 3

[12] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016. 7

[13] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 3, 6

[14] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE Conference*

[15] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 3, 4, 5

[16] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 7

[17] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007. 3

[18] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60. IEEE, 2013. 3

[19] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014. 3

[20] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *European conference on computer vision*, pages 742–757. Springer, 2014. 6, 7

[21] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 2, 6, 7

[22] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 3

[23] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019. 3

[24] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011. 4

[25] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019. 7

[26] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 2, 3, 5

[27] Kun Su and Eli Shlizerman. Clustering and recognition of spatiotemporal features through interpretable embedding of sequence to sequence recurrent neural networks. 4

[28] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014. 2, 7

[29] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):914–927, 2013. 7

[30] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014. 6

[31] Lei Wang, Du Q Huynh, and Piotr Koniusz. A comparative review of recent kinect-based action recognition algorithms. *arXiv preprint arXiv:1906.09955*, 2019. 7

[32] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27. IEEE, 2012. 7

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[34] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2752–2759, 2013. 3

[35] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2, 7

[36] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3, 7, 8