DAAL: Deep Activation-based Attribute Learning for Action Recognition in Depth Videos

Chenyang Zhang[★], Yingli Tian[★1], Xiaojie Guo[☆], and Jingen Liu[✪]

[★]*Department of Electrical Engineering*
The City College of New York
160 Convent Ave. New York, NY 10031.
{*czhang10@citymail,ytian@ccny*}*.cuny.edu*
[☆]*School of Computer Software*
Tianjin University
Tianjin 300350, China
xj.max.guo@gmail.com
[✪]*SRI International*
jingen.liu@sri.com

Abstract

In this paper, we propose a joint semantic preserving action attribute learning framework for action recognition from depth videos, which is built on multi-stream deep neural networks. More specifically, this paper describes the idea to explore action attributes learned from deep activations. Multiple stream deep neural networks rather than conventional hand-crafted low-level features are employed to learn the deep activations. An undirected graph is utilized to model the complex semantics among action attributes and is integrated into our proposed joint action attribute learning algorithm. Experiments on several public datasets for action recognition demonstrate that 1) the deep activations achieve the state-of-the-art discriminative performance as feature vectors and 2) the attribute learner can produce generic attributes, and thus obtains decent performance on zero-shot action recognition.

Keywords: Attribute Learning, Action Recognition, Depth Camera

[1]Corresponding author.

1. Introduction

Over the past decade, many research efforts have been made towards recognizing human actions from RGB videos [1, 2, 3, 4]. Recently, with the increasing applications of RGBD cameras in surveillance and human-computer interaction, recognizing human actions using depth information becomes more attractive to the success of many intelligent systems. Compared with RGB cameras, RGBD cameras provide more geometric information such as human body size, shape and position, which is significantly important for action recognition. However, most of the existing depth-based action recognition methods concentrate on designing various low-level features from different information channels such as 1D skeleton joints [5, 6, 7], 2D Depth Motion Map (DMM) [8], and 3D spatial-temporal volume [9, 10, 11, 12, 13]. The feature extracted from each channel is usually well designed and tuned independently for a specific framework.

However, for human recognition processes, information received from multiple receptors would activate some corresponding specific neurons, which further enable high-level neuron activations to accomplish the abstract recognition task (*e.g.*, face recognition) [14]. Inspired by this observation, we propose a uniform deep feature learning architecture, which can automatically learn homogeneous features from heterogeneous channels. These learned features are named as *Deep Activations*, since each feature element acts like a neuron that can be activated to capture some properties of the learning. Leveraging this deep learning network, our system is able to treat each information channel identically. In other words, only the Deep Activations are visible to the high-level learners such as the attributes proposed shortly. As a result, deep learning models have been successfully applied to large-scale visual recognition tasks using multiple layers of convolution filters [15, 16, 17]. Compared with conventional hand-designed features, deep-learned features are advantageous not only because they need much less effort and domain knowledge to become more generic to different modalities, but also because of their potential to automatically learn an organized hierarchy of semantic features [18].

Although the deep learning network has been very successful in visual recognition, the deep features are usually treated as mid-level features [19], and function like signal filters, which affect the recognition performance and limit their applications. Therefore, inspired by [20], instead of directly mapping deep features onto action labels, a set of pre-defined action attributes serves as mid-level representations. These attributes can boost recognition and enable new applications such as zero-shot learning. As human bodies/joints are easier to track than open

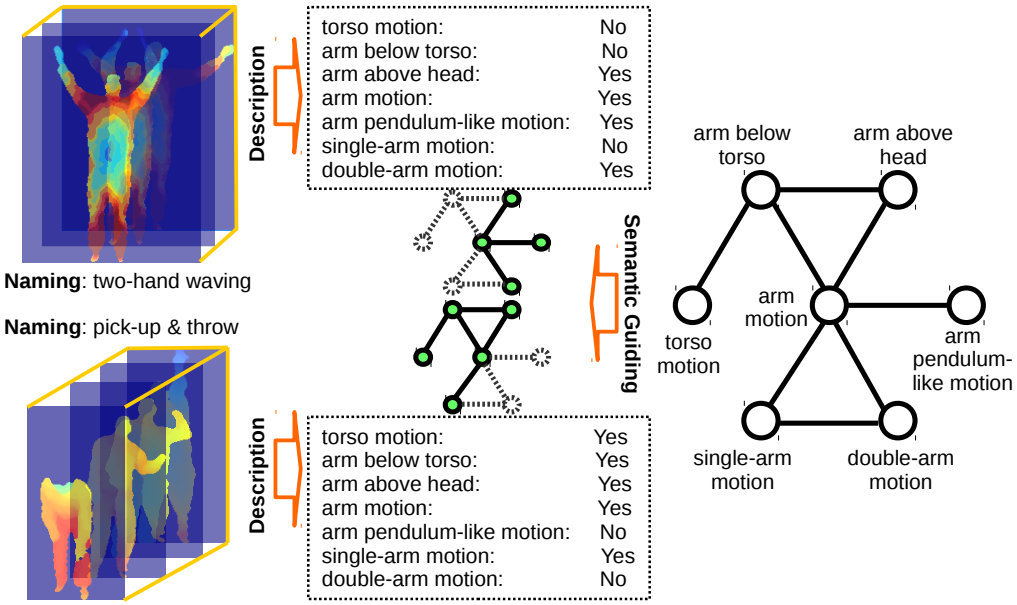

Figure 1: Illustration of the proposed joint action attribute learning algorithm. Instead of treating each action attribute independently, we apply a semantic graph to guide the joint action attribute learning algorithm to preserve the relationship among action attributes (*e.g.* "arm below torso" may share common information with "arm motion" and "torso motion".)

source videos in [20], we found that action attributes are more appropriate for describing actions in depth videos. To our knowledge, our work is the first attempt to leverage "attributes" to recognize actions from depth videos.

Including [20], most existing attribute learning approaches tend to learn the attribute detectors independently. As a result, some detectors may learn the properties that do not belong yet correlate to the attribute of interest. In other words, they do not "learn the right thing" [21]. For example, the attribute detector "arm motion" may learn patterns related to "torso up-down motion" in action "jogging" because of their co-occurrence. We believe the semantic/geometric relationships among the attributes can serve as constrains during the attribute learning, and eventually enable the detectors to learn the exact human motions and postures. Therefore, this paper introduces a joint attribute learning framework which leverages the relationships among attributes represented by a graph, as shown in Figure 1. The proposed algorithm utilizes a relationship affinity graph in the optimization of attribute detector learning processes. It tends to decorrelate attributes that are

semantically distant, while enhance correlation of neighboring attribute detectors.

In object attribute learning, Jayaraman *et al.* [21] propose to use "groups" to define the relationships among object attributes. However, since object attributes are much more fine-grained than action attributes (*e.g.*, "furry" and "brown"), they can be organized into "groups" such as "color", "shape" and "texture", but it is not helpful to coarsely group action attributes based on human bodies such as "arm", "head", "torso", *etc*. For instance (see Figure 1), the action attribute "arm below torso" is related to "arm above head" as both describe positions of upper limbs, but it is also related to "torso motion" because both are related to the body part "torso". Therefore, we argue that an undirected relation graph is better to capture the semantic/geometric relationships among action attributes compared to "groups". Actually, to some extent, the relation graph also groups attributes if they are close on the graph. But it captures more complex relationship beyond the "groups". Our experiments further verify that attribute detectors trained with the proposed graph perform much better than detectors trained with "groups".

In summary, as illustrated in Figure 2, our system takes the heterogeneous visual information received from the 1D, 2D and 3D channels as inputs, and then leverage the deep neural networks to automatically learn the homogeneous deep activations. Building on the deep activation, our system further jointly learns the attribute detectors by leveraging graph-based constraints. These attributes enable zero-shot learning and further boost the action recognition.

Our Contributions: 1) we propose a uniform framework to learn homogeneous deep activations from the heterogeneous information sources. It is superior to most previous work on recognizing human actions from depth videos, which heavily relies on hand-designed low-level features. 2) Our system jointly learns attribute detectors by incorporating the attribute relation graph as constraints, which de-correlates some attributes, and as a result enables the detectors to "learn the right thing". The relation graph pre-defines the semantic/geometric relationships among action attributes, which is superior to "groups" based constraints for action recognition. In this paper, we are focusing on how to regularize attribute learning parameters via pre-defined graph. To the best of our knowledge, this paper is the first to leverage deep learning features to jointly learn action attribute detectors constrained on the relation graph to de-correlate attributes for action recognition from depth videos. The proposed algorithm are evaluated on three benchmarked datasets, and experimental results demonstrate the effectiveness of the proposed framework by achieving the state-of-the-art performances on both attribute detection and zero-shot action recognition.

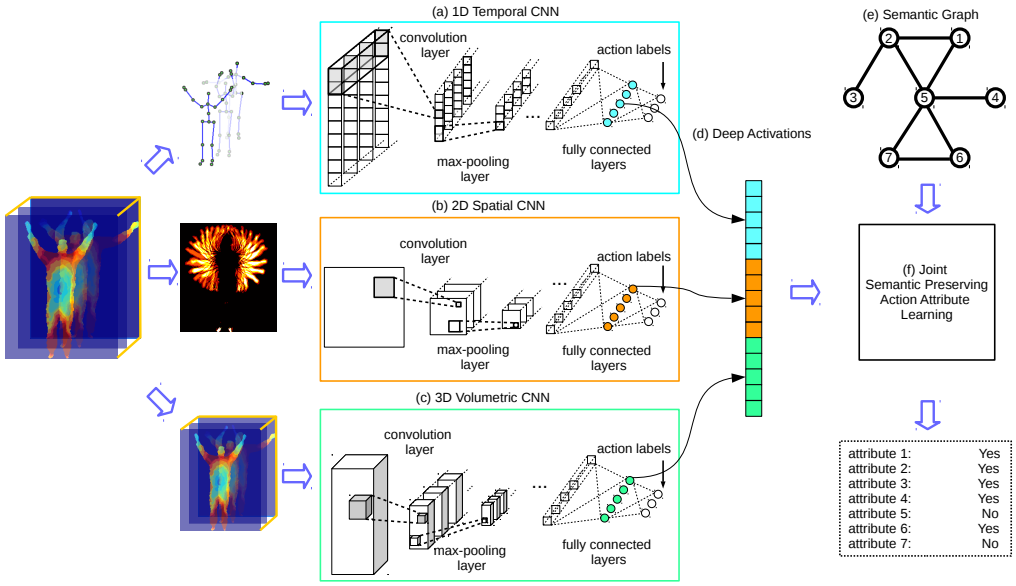

Figure 2: **Overview of the proposed deep activation-based action attribute learning modelling**. (a), (b) and (c) Multiple Convolutional Neural Networks are trained on different dimensional representations of the given depth videos such as 1D skeleton joint coordinates, 2D depth motion maps and 3D video volumes. The CNNs are trained in a supervised manner where action labels are used. (d) The second-last layer neuron activations from multiple CNNs are collected as **Deep Activations**. (e) and (f) Semantic preserving joint attribute learning algorithm is proposed by leveraging the prior knowledge of relations among attributes.

2. Related Work

Action Recognition from Depth Sequences: Action recognition from depth videos can be roughly categorized into two groups, depth map-based methods [22, 11, 10], and joint based methods [5, 7, 6, 23] which are based on a skeleton joint estimator (*e.g.* [24]). In depth map-based methods, Li *et al.* [22] sample the silhouette points from three 2D planes that are orthogonal projections from a 3D point cloud, and then use a bag of 3D points together with action graph [25] to infer class labels. Some researchers have developed 3D or 4D local patterns such as 4D-normals [11], occupancy patterns [9, 10], and cuboid features [26]. As an instance for joint-based methods, Xia *et al.* [5] directly use 3D joint locations to recognize actions by a Hidden Markov Model on posture words.

Deep Learning for Action Recognition: Deep Convolutional Neural Network (CNN) has been applied in video classification [27] and action recognition

[28]. In [28], learned spatio-temporal features from video sequences using independent subspace analysis achieve the state-of-the-art performances on several benchmark datasets. Combined with the exploration by Zeiler *et al.* [18], the deep-learned features demonstrate desirable properties such as increasing invariance and class discrimination with ascending layers. Inspired by the above progress, our proposed framework leverages the advantageous properties of deep-learned features (*i.e.* deep activations) and mines high-level semantic concepts (action attributes). In recent years, sequential input networks (RNN) have also played an essential role in action recognition. In [29], human-tracking problem and group recognition problem are jointly tackled by a two-staged framework composed of two LSTM modules: one for person-level action recognition and the other for group level dynamics. In [30], LSTMs are applied on different body joint groups to perform skeleton-based action recognition. In addition to learning representation, deep networks are also used in learning view point transfer function [31] for action recognition from novel viewpoints. Besides, deep autodecoders are also employed in feature fusion [32]. In [33], multi-stream deep network architecture is also employed to learn action descriptors from trajectory-based raw features.

Attribute Learning: Attributes serve as high-level semantic features for visual recognition tasks [34, 35]. As studied by Liu *et al.* [20], action attributes are useful for zero-shot action recognition. Deep learning based attribute classification [36] shows promising results in human attribute classification. The relationships among attributes are often ignored in attribute learning, which may result in learning the correlated yet wrong properties. In [21], the authors propose to decorrelate the attributes by grouping object attributes into disjoint groups to eliminate the ambiguity. However, simply grouping attributes is inadequate to model the complexness of action attributes. Therefore, an undirected graph is integrated into the joint attribute learning framework to preserve the complex action semantics.

3. Architecture of Learning Deep Activations

This section elaborates the architecture of each deep CNN in our multi-stream deep neural network framework. An overview of three types of CNN architectures is illustrated in Figure 3. Note that the numbers of dimensions in this figure are trained on the MSR Action3D dataset [22]. For different datasets, these numbers

may vary [2].

1D Representation: In the 1D-Temporal-CNN model, the input is a 1D sequence where the dimension is the frame number of the depth video. Each element in the sequence represents the skeleton joints in the corresponding depth frame. Each coordinate of a skeleton joint is compared with 1) its two counterparts in the previous and initial frames and 2) the anchor joint in the current frame and the coordinate offsets are used for representation. Thus the dimension of each joint is 6 and for each skeleton is 120 (20 joints). An abstract feature extraction layer is composed by one temporal convolution layer and one max-pooling layer. Three abstract layers and an additional 3-layer multilayer perceptron (MLP) are added. The deep activation layer here denotes the second layer in the MLP, which is composed by abstract features learned from input and supervised by its action label.

2D Representation: To capture the spatial energy distribution of an action, Depth Motion Map (DMM) [8] is employed for each depth sequence as the 2D representation. The input of 2D-spatial-CNN is a 128×128 depth motion map that characterizes the spatial movement during the whole action. Then 4 abstraction layers are employed before the MLP.

3D Representation: In many deep learning-based action recognition algorithms [28, 27], the spatial-temporal video volumes, *per se*, can also be a representation. In our work, the depth spatial-temporal 3D volume itself is used as the 3D representation. The input of 3D-Volumetric-CNN is a $128 \times 128 \times T$ ($T = 39$ in Figure 3) tensor which is the normalized video volume itself. The filters are also 3D-tensors which are applied on the spatial-temporal subvolumes of the depth video to extract features. More implementation details for 1D, 2D and 3D representations can be found in the appendices. In this work, each CNN is trained individually in a supervised manner. By collecting deep activations learned from multiple representations, the deep activations are desired to be discriminative from different aspects. Another benefit of using multiple representations is that it can alleviate the demand of a vast amount of training data for deep CNNs [36].

We also apply drop-out layers after each of the convolution layers in all CNN models to avoid feature co-adaptation. The idea of drop-out is proposed by Hinton *et al.* [37], to randomly zero a fraction of the neuron units during training processes. The drop-out layers can effectively avoid the overfitting caused by

[2]The actual numbers of dimensions shown in the figure may vary in different datasets. Here the numbers are of our models trained on MSR Action3D dataset [22]

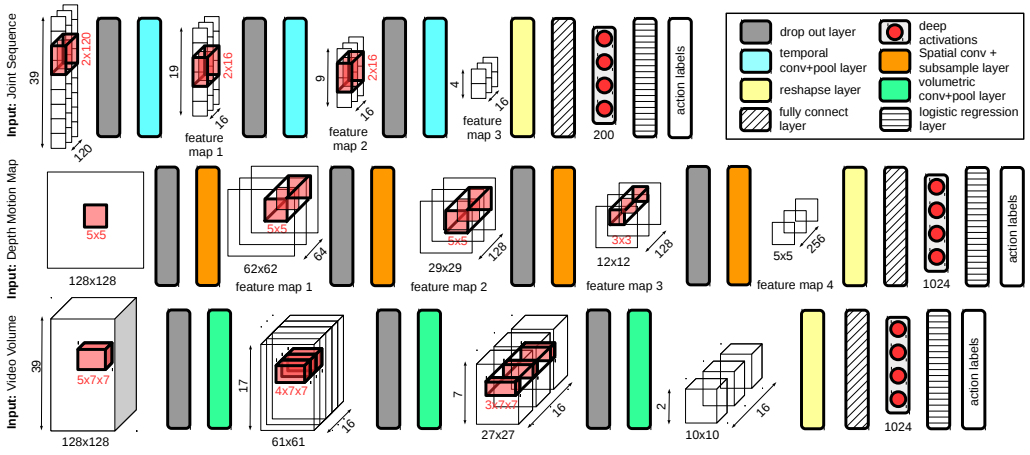

Figure 3: Overview of architectures for each of deep CNNs employed in proposed algorithm. Top row is for 1D-Temporal-CNN and the middle and bottom rows are for 2D-Spatial-CNN and 3D-Volumetric-CNN. Legend for layers is shown in the top-right corner. Convolution filters of each layer are shown as red cubes or rectangles. Dimensions of feature maps, deep activations and filters are shown accordingly.

complex co-adaptations, where feature detectors are only helpful with a certain internal context. CNNs with random drop-out layers show improvements on speech and object recognition benchmarks, and better generalization without using very large training data. All the CNN layers in our network are trained starting from random initial values in a supervised manner by a classification loss against action labels. Different with the framework in [36], which directly learns CNNs on attribute labels, the CNNs of our framework are trained to learn action discriminative deep activations without the involvement of action attributes. The main reasons are two-fold. On one hand, training CNNs directly on action labels can ensure the learned activations are action discriminative. On the other hand, semantic relations between attributes are difficult to be directly embedded into a CNN. More favored structure must be designed to learn action attributes. The deep activations are the activations in the middle layer of MLP in each CNN. We collect all deep activations together as the final output of the multi-stream deep CNNs for each depth video sequence. For instance, as illustrated in Figure 3, the final output of the tri-stream model is a $200 + 1024 + 1024 = 2248$ dimensional activation vector.

4. Semantic Preserving Multi-task Action Attribute Learning

This section firstly discusses the characteristics of relations among action attributes and the similarity/difference with object attributes. Then the formulation of the joint semantic preserving action attribute learning problem together with an efficient solution will be introduced.

4.1. Semantic Relations among Action Attributes

Attribute learning is a popular topic in object recognition and face recognition [35, 34, 38]. While modeling co-occurrence between attributes is helpful in object recognition, attribute learning with de-correlating attribute pairs can prevent excessive biasing the likelihood function on the training set [21]. In action recognition, the benefits of using action attributes have also been initially explored in recent years [20]. However, as most of previous methods in object attribute recognition, action attributes are often learned independently without considering the relations among action attributes. In this paper, we resolve this problem by embedding the relationships among action attributes into a joint multi-task attribute learning formulation.

As object attributes are often fine-grained and have simple semantic relations, simple grouping is often enough to capture the essential information. However, action attributes have more complicated semantic relationships than object attributes, thus need a more suitable structure. Human action attributes often involve one or more body parts, therefore a natural connection would be built on the body parts that the attributes involve. For example, as illustrated in Figure 1, the attribute "arm below torso" is related to body parts "torso" and "arm", so it is related to attributes "torso motion" and "arm motion". In addition, since "arm below torso" is an attribute describing "*the position of upper limbs*", it is related to other attributes of the same topic, such as "hand above head". In this work, the relationships among attributes are represented by an undirected graph. An example of such graph is provided in Figure 1 and more detailed semantic graphs can be found in the appendices.

4.2. Joint Attribute Learning

As suggested in [20], we pre-define a number of attributes as well as their correspondences between each action class. The protocol to label these attributes is based on motions and relative positions of body parts. Therefore from the ground-truth action class labels, we infer the attribute labels for each training sample by

its action label. 1 is used to indicate that the attribute is "active" and -1 otherwise. In the following, the formulation the joint attribute learning problem as a multi-task learning problem is proposed.

Formulation: Suppose there is a set of training samples $X \in \mathbb{R}^{M\times N}$ and corresponding attribute labels $Y \in \{-1, 1\}^{K\times N}$, where each column $X_{i\in[1,N]}$ of X is a learned deep activation and each column Y_i in Y is X_i's attribute label. M is the deep activation dimension, N and K are the numbers of training samples and number of defined attributes, respectively. The objective is to learn a matrix $W \in \mathbb{R}^{M\times K}$. Each column $W_{k\in[1,K]}$ in W is the parameter of the corresponding attribute predictor where $W_k^T X_i = Y_i^k$.

Therefore, learning the optimal W is to minimize the following problem:

$$W^* = \underset{W}{\operatorname{argmin}}\ \mathcal{L}(X, Y, W) + \mathcal{O}(W), \tag{1}$$

where $\mathcal{L}(X, Y, W)$ is the empirical loss function of predicting attribute labels. In this work, we use $\|W^T X - Y\|_F^2$ as our loss. And $\mathcal{O}(W)$ is a regularization term on W to pursue some prior structures such as sparsity.

What are the desired properties of W? Since deep activation vector is discriminative on action labels and each one has the potential to describe a semantic concept, so an attribute should have sparse response to the deep activation vector. As suggested by [21] and [39], the group sparsity enforced by $l_{2,1}$ norm plays an important role in feature selection. Secondly, to preserve the semantic relationships among attributes, the attributes that are semantically close should share features while distant ones should compete for features. We advocate this property by using the *graph Laplacian* of a predefined attribute graph.

By putting all the concerns aforementioned together, the problem of semantic preserving joint attribute learning can be formulated in the following shape:

$$W^* = \underset{W}{\operatorname{argmin}}\ \|W^T X - Y\|_F^2 + \lambda\|W\|_{2,1} + \beta tr(WLW^T), \tag{2}$$

where λ and β are the weights for the row-sparsity and semantic preserving regularizers, respectively. The first term is the empirical loss for predicting attribute labels. The second term introduces row-sparsity to the learned weight matrix, which avoids overfitting and includes feature-selection. The third term models the relationships among attribute weight vectors based on the graph.

Optimization: To efficiently and effectively solve the problem in Eq. (2), two auxiliary variables are introduced to make the problem separable, which give

the following program:

$$\begin{aligned} &\min_{W,P,Q} \|P^T X - Y\|_F^2 + \lambda \|Q\|_{2,1} + \beta tr(WLW^T) \\ &s.t. \quad W = P, \quad W = Q. \end{aligned} \tag{3}$$

The program in Eq. (3) can be solved in an unconstrained form by the dual ascent method. To bring robustness to the dual ascent method, we use Augmented Lagrangian methods (ALM) to generate the augmented Lagrangian for Eq. (3):

$$\begin{aligned} \mathcal{L}_\rho(X, Y, W, P, Q) = &\|P^T X - Y\|_F^2 + \lambda \|Q\|_{2,1} \\ &+ \beta tr(WLW^T) + \langle Z_1, P - W \rangle \\ &+ \frac{\rho}{2} \|P - W\|_F^2 + \langle Z_2, Q - W \rangle \\ &+ \frac{\rho}{2} \|Q - W\|_F^2, \end{aligned} \tag{4}$$

where Z_1 and Z_2 are Lagrangian multipliers associated with the two constraints in Eq. (3), and ρ is a positive penalty. Since the program in Eq. (4) is separable, we can apply the *alternating direction method of multipliers* (ADMM) [40] strategy. The solutions of the sub-problems based on ADMM are shown as follows:

W sub-problem: With unrelated terms discarded, this sub-problem becomes a classic least squares problem and the optimal $W^{(t+1)}$ can be calculated easily by:

$$\begin{aligned} W^{(t+1)} &= \operatorname*{argmin}_W \mathcal{L}_\rho(W, P^{(t)}, Q^{(t)}) \\ &= \operatorname*{argmin}_W \beta tr(WLW^T) + \frac{\rho}{2} \|P^{(t)} - W + u_1^{(t)}\|_F^2 \\ &+ \frac{\rho}{2} \|Q^{(t)} - W + u_2^{(t)}\|_F^2 \\ &= \rho(P^{(t)} + Q^{(t)} + u_1^{(t)} + u_2^{(t)})(2\beta L + 2\rho I)^{-1}, \end{aligned} \tag{5}$$

where $u_1^{(t)} = (1/\rho) Z_1^{(t)}$ and $u_2^{(t)} = (1/\rho) Z_2^{(t)}$ are scaled dual variables which make the representation more compact by combining linear and quadratic terms. Note that the matrix inverse $(2\beta L + 2\rho I)^{-1}$ only needs to be computed once.

P sub-problem: Similar to the W sub-problem, the P sub-problem is also a

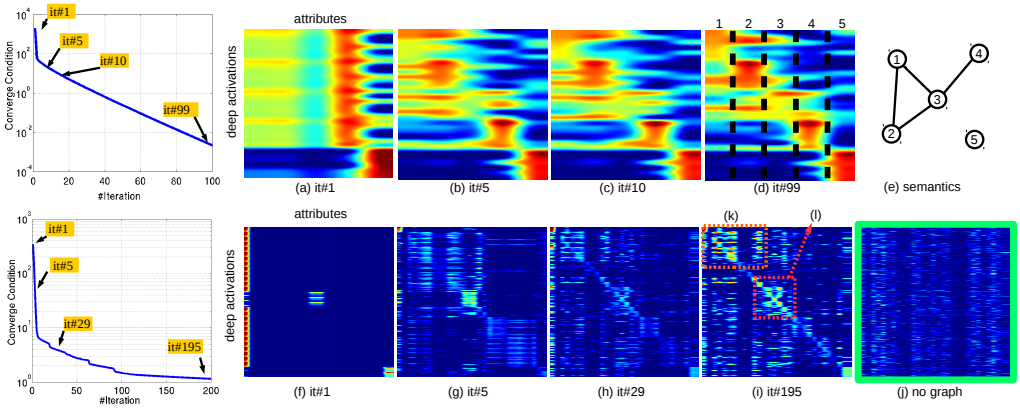

Figure 4: Illustration of the effect of Algorithm 1 on a synthetic dataset with 5 attributes (the top row) and MSR Action dataset with 30 attributes (the bottom row). (a-d) and (f-i) are learned weights for sampled iterations. Columns correspond to attributes and rows correspond to features or deep activations. Warmer colors indicate higher absolute values in weight matrix, the more the attribute relies on the feature. (e) The underlying semantic graph of the synthetic dataset. (j) The result generated without graph involved for comparison. (k) and (l) show two examples of graph-guided effects, please see text for details.

classic least squares problem:

$$\begin{aligned} P^{(t+1)} &= \underset{P}{\operatorname{argmin}}\ \mathcal{L}_\rho(X, Y, W^{(t+1)}, P, Q^{(t)}) \\ &= \underset{P}{\operatorname{argmin}}\ \|P^T X - Y\|_F^2 \\ &+ \frac{\rho}{2}\|P - W^{(t+1)} + u_1^{(t)}\|_F^2 \\ &= (2X^T X + \rho I)^{-1}[2X^T Y + \rho(W^{(t+1)} - u_1^{(t)})]. \end{aligned} \tag{6}$$

Please note that the terms $(2X^T X + \rho I)^{-1}$ and $2X^T Y$ also need to be computed only once.

Q sub-problem: The closed form solution of $Q^{(t+1)}$ can be obtained by:

$$\begin{aligned} Q^{(t+1)} &= \underset{Q}{\operatorname{argmin}}\ \mathcal{L}_\rho(W^{(t+1)}, P^{(t+1)}, Q) \\ &= \underset{Q}{\operatorname{argmin}}\ \lambda\|Q\|_{2,1} + \frac{\rho}{2}\|Q - W^{(t+1)} + u_2^{(t)}\|_F^2 \\ &= \mathcal{S}_{\frac{\lambda}{\rho}}^{2,1}(W^{(t+1)} - u_2^t), \end{aligned} \tag{7}$$

where $\mathcal{S}^{2,1}_{\epsilon>0}(\cdot)$ represents the shrinkage operator [41].

In addition, the two scaled dual variables u_1 and u_2 need to be updated using corresponding residuals:

$$\begin{aligned} u_1^{(t+1)} &= u_1^{(t)} + P^{(t+1)} - W^{(t+1)} \\ u_2^{(t+1)} &= u_2^{(t)} + Q^{(t+1)} - W^{(t+1)}. \end{aligned} \tag{8}$$

For clarity, we summarize the optimization procedure of the deep activation-based attribute learning algorithm (DAAL) in Algorithm 1. The algorithm terminates when $(\|P^{(t)} - W^{(t)}\|_F + \|Q^{(t)} - W^{(t)}\|_F) \leq \delta(\|P^{(0)} - W^{(0)}\|_F + \|Q^{(0)} - W^{(0)}\|_F)$ where $\delta = 10^{-5}$, or when the predefined maximal number of iterations is reached.

Algorithm 1: DAAL

Input: Deep Activation Matrix X, Attribute Ground-truth Y

Initialization: Randomly initialize $W^{(0)}$,$P^{(0)}$,$Q^{(0)}$, Set $u_1^{(0)}$ and $u_2^{(0)}$ to be zero matrices. $\rho = 1.5$,$t = 0$

while *not converge* **do**

update $W^{(t+1)}$ via Eq. (5)

update $P^{(t+1)}$ via Eq. (6)

update $Q^{(t+1)}$ via Eq. (7)

update $u_1^{(t+1)}, u_2^{(t+1)}$ via Eq. (8)

$t = t + 1$

end

Output: Optimal solution W^* = $W^{(t)}$

5. Experimental Results

Effectiveness of the Algorithm: To better understand our joint attribute action attribute learning process, a simulation is conducted on five attributes with 1000 features. The semantic relationships among these attributes are shown in Figure 4 (e). One can consider the attributes to be {1: *"arm-upward motion"*, 2: *"arm-downward motion"*, 3: *"arm-motion"*, 4: *"arm below torso"*, 5: *"leg motion"*}. Learned weights for sample iterations are shown in Figure 4 (a) to (d), from which we can observe that the semantic relationship among attributes are more obvious with more iterations, note that warmer colors indicate higher absolute weights and each column corresponds to the weight vector for an attribute.

Table 1: Attribute detection scores (mean average precision) and zero-shot action recognition rates on three benchmark datasets, higher is better.

Tasks	Attribute detection scores (MAP)						Zero-shot learning (%)		
Datasets	MRA		UTA		MRP		MRA	UTA	MRP
Methods	Seen	Unseen	Seen	Unseen	Seen	Unseen	l2o	l2o	l2o
no-regularize	0.4057	0.4913	0.4880	0.3620	0.5305	0.5254	50.27	53.40	55.89
lasso	0.8283	0.5105	0.9473	0.4293	0.9894	0.6414	67.82	80.94	93.28
all-sharing [39]	0.4291	0.4794	0.6085	0.3989	0.5590	0.5809	49.48	73.07	81.82
group-lasso [21]	0.9356	0.5236	0.9051	**0.4329**	0.9985	0.6405	70.81	81.53	93.27
proposed	**0.9667**	**0.5356**	**0.9687**	0.4304	**0.9994**	**0.6426**	**72.03**	**81.89**	**94.69**

In (d), attribute "1","2" and "3" share many features, "3" and "4" share some features and "5" barely shares features with other attributes.

In addition, a similar experiment is conducted on a real dataset, MSR Action Dataset [22]. 284 samples are used with 2248 deep activations and the activation-attribute map is visualized for sampled iterations in Figure 4 (f) to (i). In (k), we show the learned pattern showing that *"arm in-front-of torso"* and *"arm above head"* tend to share features with arm-related motions while (l) *"arm below torso"* tends to share features with torso related motions. For comparison, the weights learned on the same set of features without graph involved are illustrated in (j).

5.1. Experiment Setup and Datasets

Datasets: There are three datasets for depth based action recognition used in the experiments, including the MSR Action 3D dataset [22] (**MRA**), the UTA Action 3D dataset [26] (**UTA**) and the MSR Action Pairs dataset [11] (**MRP**). The MRA dataset contains 20 gaming actions, such as "two arms waving" and "golf swing". Each action is performed by 10 different subjects and the subjects perform each action 2 to 3 times in the same location. The UTA dataset contains 10 actions which cover movements of hands, arms, legs and upper torso. Each action is performed by 10 different persons. The MRP dataset contains 6 pairs of actions that each pair of actions has opposite temporal orders, such as "push chairs" and "pulling chairs". Different from the MRA dataset, UTA and MRP allow the subjects moving around while performing actions. We define 30 action attributes for MRA, 19 for UTA and 16 for MRP, where they share some common attributes such as "arm-motion", *etc*.

Deep Activations: For the MRA and MRP datasets, since the skeleton joint locations are available, we apply all three streams of deep CNNs as illustrated in Figure 3. For the UTA dataset, only the DMM-based 2D CNN and the video-volume-based 3D CNN streams are applied because the skeleton joints are not available. For the MRP and UTA datasets, since the temporal order plays an

important role and the 2D representations is temporal order invariant, multiple CNNs are trained for 2D representations following the idea of temporal pyramid. For thorough lists of action attributes, their relationships, and CNNs used in each dataset, please refer to the appendices.

Baselines: For attribute detection and zero-shot learning, the proposed method is compared to four related baselines. All empirical loss functions are same as in Eq. (2) for uniformity. The four baselines include (1) "non-regularize": is single-task learning using least-squares loss without any regularization term. (2)"lasso" is l_1-regularized. (3)"all-sharing" is a multi-task learning method with $l_{2,1}$-regularized. (4) "group-lasso" is using the same regularize terms as in [21]. We set the default parameter values of λ and β for each baseline (if existed) to 1.

5.2. Attribute Detection and Zero-shot Action Recognition

This section shows the evaluation of the proposed joint attribute learning method on all three datasets with two tasks: 1) attribute detection and 2) zero-shot action recognition using only the learned attributes. For the first task, we employ two splitting ways for training and testing sets: 1) "Seen": this is the same as the "cross-subject" splitting protocols as in [9], [26] and [11], where half of the subjects are used for training and the remaining half for testing. All action classes appear during training. 2) "Unseen": the protocol introduced in [20] as "leave-two-out" scheme, where all combinations of action classes are considered. Since some combinations may contain attributes that do not appear in the training set, we leave these combinations out and keep the rest. For the MRA dataset, there are total of 104 combinations which fulfill this condition. For the UTA dataset, there are 20 such combinations and for the MRP dataset there are 64 combinations. Since the training on the 3D volumetric deep CNNs is time-consuming, we only train CNNs for each combination using 2D spatial and 1D spatial models for "Unseen" tasks, if available.

Table 1 shows the action attribute detection results in terms of mean average precisions (MAP) and zero-shot action recognition in terms of recognition rates. Our method outperforms other baselines in most tests. The poor results obtained by "no-regularize" indicate that the training process is easy to overfit. We observe that "group-lasso" performs stably during all tests while "all-sharing" and "lasso" do not always perform well, which suggest that solely pursuing sparsity may result in biased attribute estimations. Preserving the semantics in attributes is beneficial for attribute detection. In most cases, our method significantly outperforms "group lasso" which is proposed in [21]. This is because our method is more suitable in modeling relationships among action attributes. The right panel of Table 1 lists

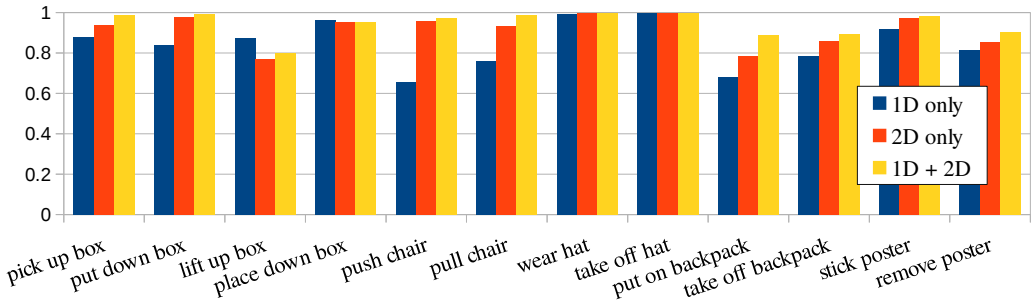

Figure 5: The average accuracies of zero-shot action recognition test on the MRP dataset using deep activations based on 1D, 2D and both representations.

the zero-shot learning action recognition results. Our method generalizes well in all datasets when dealing with zero-shot action learning, which demonstrates that our method learns better and more discriminative attribute vectors. By comparing attribute learning results and zero-shot learning results, we notice that higher MAP scores in attribute detection may not necessarily lead to better classification results in zero-shot action recognition, especially when they are very close.

Figure 5 shows the class-wise average accuracies comparison of using deep activations learned from 1D, 2D, and 1D+2D CNNs. We observe that deep activations learned from multi-stream CNNs perform better than single streams. It is interesting to observe that 2D model performs better than 1D model except for action pair "lift up box" and "place down box", since this pair of actions involves drastic motion "bent", which is easy for joint-based models.

5.3. DAAL Boosting Action Recognition

In this section, we compare the action recognition accuracies with some state-of-the-art methods. The results for three datasets are shown in Tables 2, 3, and 4, respectively. In all experiments, the same protocols used in [9], [26] and [11] are followed, where half of the subjects are used for training and the other half of subjects for testing. We evaluate deep activations, learned action attributes and their combination.

From Tables 2, 3, and 4, we can observe that the deep activation vectors are very discriminative, which demonstrate the effectiveness of our multi-stream deep architectures. Compared to previous features, the learned attributes are very compact (only 16$\sim$30 dimensions) and discriminative in action labels. By combining the learned activations and attributes together, our proposed framework achieves

Methods	Accuracy
Bag of 3D points [22]	74.70%
HOJ3D [5]	79.00%
STOP [13]	84.80%
ROP [10]	86.50%
Actionlet [9]	88.20%
HON4D [11]	88.89%
DSTIP [26]	89.30%
Pose Set [23]	90.00%
SNV [12]	91.64%
Moving Pose [6]	91.70%
deep act. (Ours)	**92.30%**
attr. (Ours)	**87.18%**
deep act. + attr. (Ours)	**93.40%**

Table 2: Comparison of action recognition rate on MSR Action 3D with other methods using the protocol in [9].

Methods	Accuracy
Posture Word[26]	79.57%
DSTIP [26]	85.80%
deep act. (Ours)	**86.87%**
attr. (Ours)	**78.79%**
deep act. + attr. (Ours)	**87.88%**

Table 3: Comparison of action recognition rate on UTA Action 3D dataset with other methods using the protocol in [26].

the best performances on all three datasets, because the attributes transfer knowledge from other classes to further complete the information for action classification.

5.4. Evidence of Learning the Right Things

In this section, we conduct an experiment to show what the attribute learner learned from deep activations. For visualization purpose, only 2D representations are used in this experiment. Given a 2D representation, we use a patch with random values to occlude it and use pre-trained deep CNN to generate the deep activation vector. Then the learned attribute classifier is employed to generate an attribute vector. By comparing the generated attribute vector with the ground-truth attribute vector, we propagate the deviation to the locations where the patch is. By

Methods	Accuracy
Skeleton + LOP [9]	63.33%
[9] + Pyramid	82.22%
HON4D[11]	97.67%
SNV [12]	98.89%
deep act. (Ours)	**98.89%**
attr. (Ours)	**87.22%**
deep act. + attr. (Ours)	**99.44%**

Table 4: Comparison of action recognition rate on MSR Action Pairs dataset with other methods using the protocol in [11].

densely sampling multi-scale occlusion patches, we can accumulate an error map, which implies the responsible region of every attribute. Some results are shown in Figure 6. By comparing results generated by "group lasso" (top) and ours, our method locates more accurately for regions responsible for a specific action attribute than "group lasso". For example, in action "hand clapping", the attribute detector for "arm-motion" in "group-lasso" concentrates on the lower-body and ours covers more on the hand area. This experiment further demonstrates that our proposed method is more suitable for feature selection in action attribute learning and it can locate the right part for a specific attribute.

6. Conclusion

In this paper, a novel joint action attribute learning algorithm for depth videos is discussed. A multi-stream deep neural networks based attribute learning framework is proposed. To model complex semantics in action attributes, a pre-defined undirected graph is integrated in the formulation of attribute learning. Extensive experiments demonstrate that the proposed method is effective in learning action attributes for depth videos. Experiment results based on our method outperform existing state-of-the-art methods in action attribute detection, zero-shot action recognition and conventional action recognition. Our future work will focus on exploring the dynamic semantic organization methods for attributes and novel attribute discovery from deep activations.

ACKNOWLEDGMENTS

This work was supported in part by NSF Grants EFRI-1137172 and IIS-1400802.

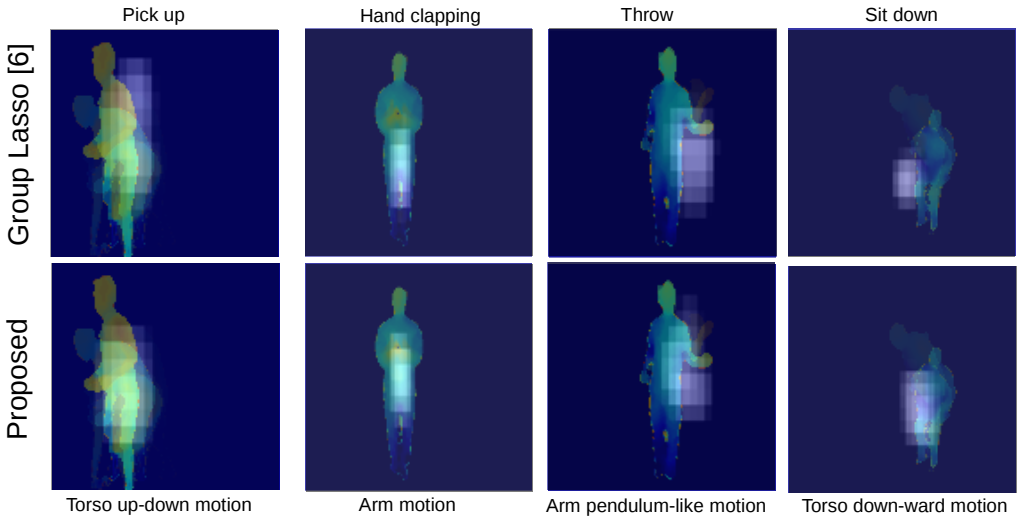

Figure 6: Sample results showing the responsible regions for attributes from UTA dataset. The top row shows the results generated by [21], the bottom row shows ours.

References

[1] J. Liu and M. Shah, "Learning human actions via information maximization," in *CVPR*, pp. 1–8, IEEE, 2008.

[2] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *CVPR*, pp. 1996–2003, IEEE, 2009.

[3] K. K. Reddy, J. Liu, and M. Shah, "Incremental action recognition using feature-tree," in *ICCV*, pp. 1010–1017, IEEE, 2009.

[4] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, pp. 1–8, IEEE, 2008.

[5] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *CVPR Workshops*, pp. 20–27, IEEE, 2012.

[6] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection," in *ICCV*, pp. 2752–2759, IEEE, 2013.

[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *CVPR*, pp. 588–595, IEEE, 2014.

[8] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *ACM Multimedia*, pp. 1057–1060, ACM, 2012.

[9] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *CVPR*, pp. 1290–1297, IEEE, 2012.

[10] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *ECCV*, pp. 872–885, Springer, 2012.

[11] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *CVPR*, pp. 716–723, IEEE, 2013.

[12] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences," in *CVPR*, 2014.

[13] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. Campos, "Stop: Space-time occupancy mail patterns for 3d action recognition from depth map sequences," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 252–259, Springer, 2012.

[14] S. Ohayon, W. A. Freiwald, and D. Y. Tsao, "What makes a cell face selective? the importance of contrast," *Neuron*, vol. 74, no. 3, pp. 567–581, 2012.

[15] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, 1995.

[16] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[17] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV 2014*, pp. 818–833, Springer, 2014.

[19] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[20] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *CVPR*, pp. 3337–3344, IEEE, 2011.

[21] D. Jayaraman, F. Sha, and K. Grauman, "Decorrelating semantic visual attributes by resisting the urge to share," in *CVPR*, IEEE, 2014.

[22] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *CVPR Workshops*, pp. 9–14, IEEE, 2010.

[23] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *CVPR*, pp. 915–922, IEEE, 2013.

[24] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[25] W. Li, Z. Zhang, and Z. Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1499–1510, 2008.

[26] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *CVPR*, pp. 2834–2841, IEEE, 2013.

[27] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[28] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR*, pp. 3361–3368, IEEE, 2011.

[29] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1971–1980, 2016.

[30] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118, 2015.

[31] H. Rahmani, A. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[32] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in RGB+ D videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[33] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream cnn," *IEEE Transactions on Multimedia*, 2017.

[34] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, pp. 951–958, IEEE, 2009.

[35] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, pp. 1778–1785, IEEE, 2009.

[36] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," *arXiv preprint arXiv:1311.5591*, 2013.

[37] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[38] F. Song, X. Tan, and S. Chen, "Exploiting relationship between attributes for improved face verification," *CVIU*, vol. 122, pp. 143–154, 2014.

[39] A. Evgeniou and M. Pontil, "Multi-task feature learning," *NIPS*, vol. 19, p. 41, 2007.

[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[41] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *ICML*, pp. 663–670, 2010.

Appendices

A.1 Depth Video Representations and CNN Configurations

This section provides additional details for Sections 3 and 4 in the paper.

1D Representation: Although "1D representation" actually uses a 2D matrix of dimension $120 \times T$ for storage, we name it "1D representation" because we treat each 120×1 vector as an element in the time sequence and apply 1D temporal convolution on that sequence in CNN. For a given skeleton joint sequence $\{K_1, K_2, .., K_T\}$, where T is the temporal dimension of the sequence, each skeleton comprises of 20 skeleton joints, such as "head", "chest", "left shoulder", *etc.* For the i^{th} joint in the t^{th} skeleton, there are two coordinates in the image coordinate system: $K^x_{t,i}$ and $K^y_{t,i}$. Each joint is mapped to a 6-dimensional space as following:

$$\begin{aligned} K^{\textbf{motion}}_{t,i} &= (K^x_{t,i} - K^x_{t-1,i}, K^y_{t,i} - K^y_{t-1,i}), \\ K^{\textbf{offset}}_{t,i} &= (K^x_{t,i} - K^x_{1,i}, K^y_{t,i} - K^y_{1,i}), \\ K^{\textbf{structure}}_{t,i} &= (K^x_{t,i} - K^x_{t,\textbf{chest}}, K^y_{t,i} - K^y_{t,\textbf{chest}}), \end{aligned} \tag{9}$$

where $K^{\textbf{motion}}_{t,i}$, $K^{\textbf{offset}}_{t,i}$ and $K^{\textbf{structure}}_{t,i}$ model the motion (translation of a specific joint for two consecutive skeletons), offset (translation compared with its counterpart in the initial skeleton), and structure (translation compared to the anchor joint, *i.e.,* chest, in current skeleton) information, respectively.

Since the lengths of videos for different datasets are different, we use $T = 39$ for the MSR Action 3D dataset [22] and $T = 20$ for the MSR Action Pairs dataset [11]. Since the UTA Action 3D dataset [26] has no skeleton joint information, we do not use 1D representation for this dataset.

2D Representation: As described in Section 3 in the paper, we employ Depth Motion Maps (DMM) [8] for 2D representations. The DMM is computed for each depth sequence and then normalized to the resolution at 128×128. Since in

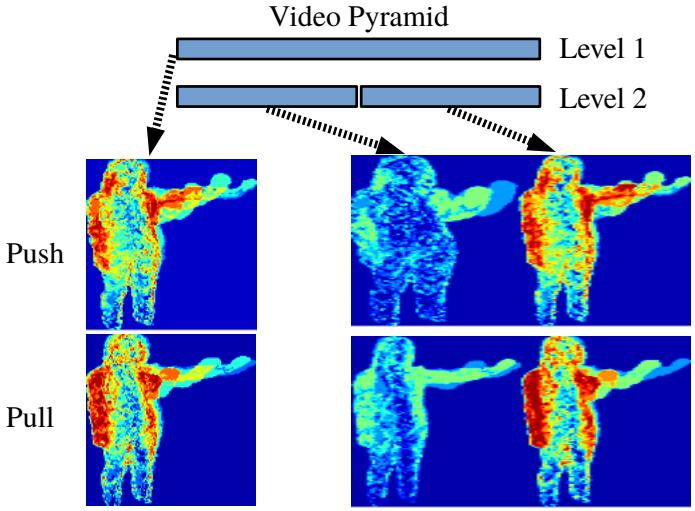

Figure 7: Illustration of the temporal pyramid idea applied to 2D representations. Two actions are listed as examples: "pull" and "push", which are temporally opposite to each other. There is almost no difference in the level 1 representation. In level 2 representation, the temporal orders of the arm motion directions are manifested.

the UTA Action 3D dataset and the MSR Action Pairs dataset, some action pairs are composed of two opposite actions in temporal orders (*e.g.*, "pull, push", "sit-down, stand-up",) we employ the idea of temporal pyramid to generate additional 2D representations as illustrated in Figure 7.

In Figure 7, we show an example of the action pair "pull" and "push" from the UTA Action 3D dataset, performed by the same subject. Level 1 representations (where DMMs are computed from whole videos) show almost no difference between these actions. While level 2 representations (where the first DMM is computed from the first half of the video and the second from the rest half) show the temporal order as "reaching out the arm" in action "push" and "drawing back the arm" in action "pull".

In our work, if temporal pyramid is applied, the temporal grids are sampled evenly without overlapping. Each level generates a separate 2D representation which is rescaled to 128×128. For example, if the depth videos from a dataset use three levels of 2D representations as well as 1D and 3D representations, total of five representations are generated and each of them is fed into a CNN individually.

3D Representation: As described in Section 3 in the paper, the 3D representations are the depth video volumes which resolutions are normalized to

$128 \times 128 \times T$. The value of T in each dataset is consistent with the value in the 1D representations for MSR Action 3D dataset and MSR Action Pairs dataset. For UTA Action 3D dataset, we set $T = 20$.

Leaning parameters For all three representations, we used the same set pf hyper-parameters, *i.e.*, learning rate = $1e^{-4}$; training stops at iteration 100K or training accuracy increases for 5K steps, whichever is earliest; dropout is applied as show in Figure 2.

In Table 5, we list the detailed information of CNN applied in each dataset for each task and the final dimensions of deep activation vectors. Temporal pyramid organizations are only applied for 2D representations. If a particular task uses temporal pyramid, the levels are also listed. "Level 1" indicates the whole video sequence, "level 2" indicates the video is segmented to two even and non-overlapping segments, and so on.

Table 5: Details of CNN configurations applied in each dataset and for each task. Blue ticks indicate the model is employed in corresponding task and red crosses indicate otherwise. If temporal pyramid is employed, levels of pyramid are shown in the next column. The dimensions of final deep activation vectors are shown in the last column.

Dataset	Task	1D	2D	3D	Pyramid	Level(s)	Deep Act. Dimension
MRA	Seen	✔	✔	✔	✘	N/A	2248
	Unseen	✔	✔	✘	✘	N/A	1224
UTA	Seen	✘	✔	✔	✔	1,2	3072
	Unseen	✘	✔	✘	✔	1,2	2048
MRP	Seen	✔	✔	✔	✔	1,2,3,4	5320
	Unseen	✔	✔	✘	✔	1,2,3,4	4296

A.2 Action Attribute Definitions and Relationships

Tables 6, 7 and 8 list the action attributes defined in our paper for the MSR Action 3D, the UTA Action 3D, and the MSR Action Pairs datasets respectively. In each table, columns represent actions and rows represent action attributes. Active attributes are labeled with blue ticks for each action. The relationship graphs for the attributes are shown in Figures 8, 9 and 10.

Please note that when we implement the method of "group-lasso" for comparison, the grouping of action attributes is obtained in such a manner that attributes in each green rectangle are grouped together. All other attributes are considered as separate groups. For example, in Figure 8, "single arm motion" and "double arm motion" are considered as one group and "arm pendulum-like motion" is

considered as another groups.

Table 6: Attribute defined for MSR Action 3D Dataset. Each column is an action and each row is an attribute. There are in total 30 attributes defined for 20 actions. Blue ticks indicate presences of attributes in actions.

	high arm wave	horizontal arm wave	hammer	hand catch	forward punch	high-throw	draw x	draw tick	draw circle	hand clap	two hand wave	side boxing	bend	forward kick	side kick	jogging	tennis swing	tennis serve	golf swing	pick-up and throw
arm pendulum-like motion	✔	✔									✔								✔	
arm-motion	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔				✔	✔	✔	✔	✔
arm-forward motion			✔		✔	✔	✔			✔							✔	✔		✔
arm horizontal motion	✔	✔								✔	✔	✔							✔	
arm verticle motion			✔	✔				✔									✔	✔		✔
raise arm	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔						✔		✔
put down arm	✔	✔		✔	✔	✔	✔	✔	✔	✔	✔	✔					✔	✔		✔
draw-x							✔													
draw-tick								✔												
draw-o									✔											
arm in-front-of torso		✔	✔		✔		✔	✔	✔	✔							✔	✔		
hand above head	✔		✔	✔		✔					✔						✔	✔		✔
hand below torso													✔					✔	✔	✔
torso up-down motion													✔			✔				✔
torso twist												✔					✔	✔	✔	
bent													✔						✔	✔
single-arm motion	✔	✔	✔	✔	✔	✔	✔	✔	✔			✔					✔			✔
double-arm motion										✔	✔		✔					✔	✔	
leg pendulum-like motion														✔	✔					
leg side motion															✔					
leg forward motion														✔						
leg motion												✔		✔	✔	✔				
repeated	✔	✔									✔					✔				
symmetric-t				✔	✔					✔		✔	✔	✔	✔					
symmetric-y										✔	✔					✔				
side-motion				✔						✔	✔	✔			✔				✔	
forward-motion					✔	✔	✔	✔	✔	✔				✔			✔	✔		✔
full-body motion												✔	✔			✔	✔	✔	✔	✔
one-phase motion	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔		✔	
coordinative motion																		✔		✔

Table 7: Attribute defined for UTA Action 3D Dataset. Each column is an action and each row is an attribute. There are in total 19 attributes defined for 10 actions.

	walk	sitDown	standUp	pickUp	carry	throw	push	pull	waveHands	clapHands
arm pendulum-like motion	✔					✔			✔	
arm-motion	✔			✔		✔	✔	✔	✔	✔
arm horizontal motion						✔	✔	✔		✔
arm vertical motion				✔					✔	
hand inward motion				✔				✔		✔
hand outward motion				✔		✔	✔			✔
arm in-front-of torso										✔
hand above head									✔	
hand below torso				✔						
torso up-down motion		✔	✔	✔						
torso upward motion			✔	✔						
torso downward motion		✔		✔						
body transition	✔				✔					
bent				✔						
single-arm motion						✔	✔	✔		
double-arm motion	✔								✔	✔
leg motion	✔	✔	✔	✔	✔					
full-body motion	✔	✔	✔	✔	✔					
with object		✔	✔	✔	✔	✔				

Table 8: Attribute defined for MSR Action Pairs Dataset. Each column is an action and each row is an attribute. There are in total 16 attributes defined for 12 (6 pairs of) actions.

	pick up box	put down box	lift up box	place down box	push chair	pull chair	wear hat	take off hat	put on backpack	take off	stick poster	remove poster
reach out arm	✔	✔	✔	✔					✔	✔	✔	✔
draw back arm	✔	✔	✔						✔			
hold object	✔	✔	✔	✔			✔	✔	✔	✔	✔	✔
hand below torso			✔	✔					✔	✔		
hand above head							✔	✔				
torso motion			✔	✔	✔	✔			✔	✔	✔	✔
bent			✔	✔					✔	✔		
leg motion					✔	✔						
body translate					✔	✔						
horizontal motion	✔	✔			✔	✔						
vertical motion			✔	✔			✔	✔	✔	✔	✔	✔
withdraw object	✔		✔			✔		✔	✔			✔
deposit object		✔		✔	✔		✔			✔	✔	
object-size small							✔	✔				
object-size medium	✔	✔	✔	✔					✔	✔	✔	✔
object-size large					✔	✔						

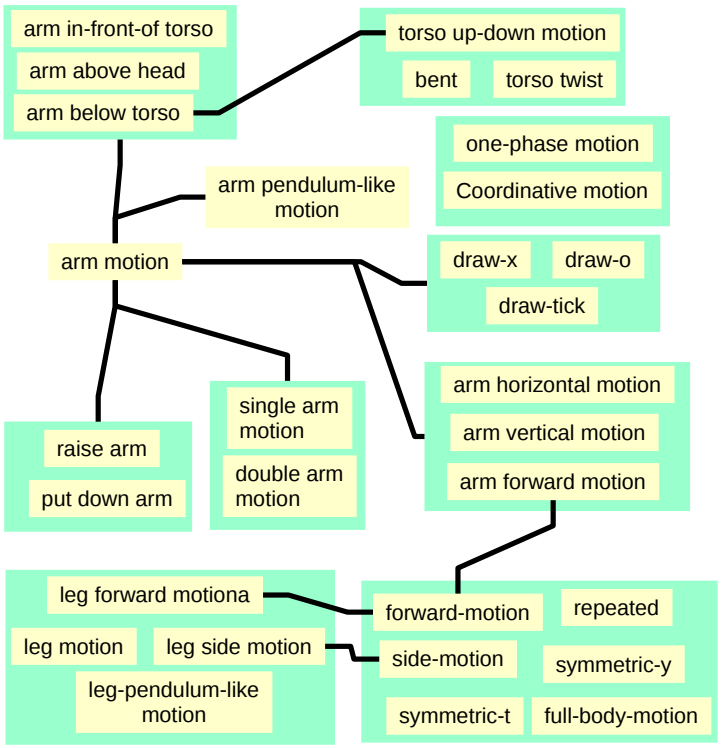

Figure 8: Semantic relationships among action attributes for the MSR Action 3D Dataset. Every attribute in the same rectangle is connected to each other. For clarity, we only show connections across different rectangles with solid lines. For example, action attribute "arm-motion" is connected to each attribute in the bottom rectangle consisting of "single arm motion" and "double arm motion"; "leg forward motion" is connected to "forward motion".

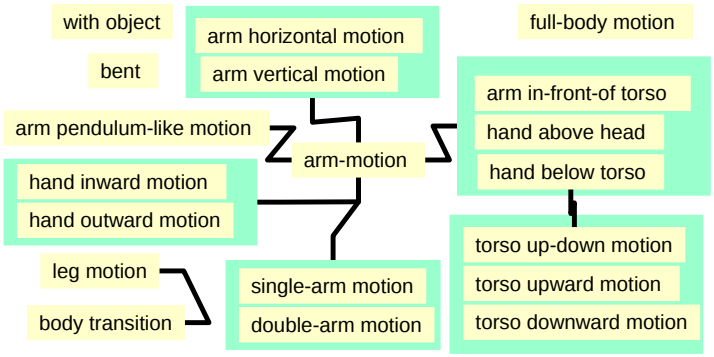

Figure 9: Semantic relationships among action attributes for the UTA Action 3D Dataset.

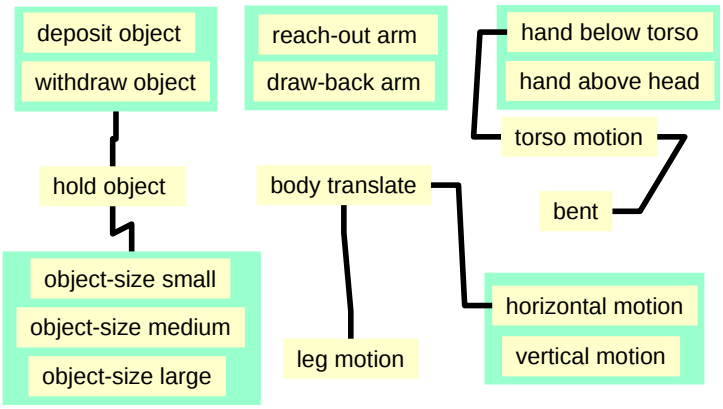

Figure 10: Semantic relationships among action attributes for the MSR Action Pairs Dataset.