3D Single-Person Concurrent Activity Detection Using Stacked Relation Network

Yi Wei,[1]* Wenbo Li,[2]* Yanbo Fan,[3] Linghan Xu,[4] Ming-Ching Chang,[1] Siwei Lyu[1]
[1]University at Albany, State University of New York, USA
[2]Samsung Research America AI Center, USA
[3]Tencent AI Laboratory, China, [4]Tianjin University, China
{ywei2, mchang2, slyu}@albany.edu, wenbo.li1@samsung.com, fanyanbo0124@gmail.com, linghanxu@tju.edu.cn

Abstract

We aim to detect real-world concurrent activities performed by a single person from a streaming 3D skeleton sequence. Different from most existing works that deal with concurrent activities performed by multiple persons that are seldom correlated, we focus on concurrent activities that are spatio-temporally or causally correlated and performed by a single person. For the sake of generalization, we propose an approach based on a *decompositional design* to learn a dedicated feature representation for each activity class. To address the scalability issue, we further extend the class-level decompositional design to the postural-primitive level, such that each class-wise representation does not need to be extracted by independent backbones, but through a dedicated weighted aggregation of a shared pool of postural primitives. There are multiple interdependent instances deriving from each decomposition. Thus, we propose Stacked Relation Networks (SRN), with a specialized relation network for each decomposition, so as to enhance the expressiveness of instance-wise representations via the inter-instance relationship modeling. SRN achieves state-of-the-art performance on a public dataset and a newly collected dataset. The relation weights within SRN are interpretable among the activity contexts. The new dataset and code are available at https://github.com/weiyi1991/UA_Concurrent/

1 Introduction

Human activity detection is an important problem in computer vision with many practical applications such as video surveillance, human computer interaction, smart home, *etc*. There are several issues hindering the practical applicability of activity detection algorithms, including a large variety of scenes (*e.g*, background, viewpoint, distance and illumination), diversity in human-object interactions (*i.e*, similar motions may appear in different activities regarding different objects), and complexity of body movement (*i.e*, human ego motion is usually multi-purpose and temporally or causally correlated). Among these issues, the *variety of scenes* can be handled by disentangling human motions from the environment using 3D human skeletal key points (Aggarwal and Xia 2014). The *diversity in human-object interactions* is often addressed by integrating image appearance features with 3D skeletal data (Wang et al. 2012; Yu, Liu, and Yuan 2014). However, the last issue of *complexity of body movement* has not been well addressed. Most existing 3D skeleton based activity recognition methods (Gao et al. 2018; Hu, Cui, and Yu 2018; Ke et al. 2018; Li et al. 2018; Liu, Akhtar, and Mian 2017; Shi et al. 2018; Weng et al. 2019; Yan, Xiong, and Lin 2018; Zhang et al. 2018) are developed for *trimmed* activity sequences and limited by the assumption that one subject can only perform one activity at a time.

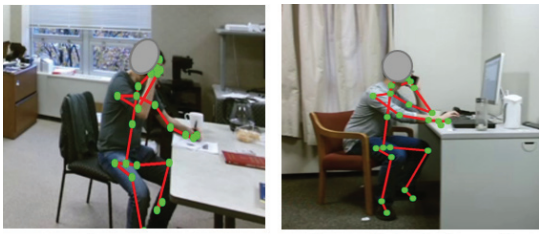

Figure 1: Samples of single-person concurrent activities.

This one-activity-at-a-time assumption is indeed problematic in the reality, as one person usually performs multiple activities concurrently. In this work, we aim to address the problem of detecting single-person *concurrent* activities (see Figure 1) from streaming 3D skeleton sequences. In our setting, representing a specific body motion with a unified fixed-size feature representation is not a good solution due to two generalization issues. *(i)* It is hard to make a unified fixed-size representation flexible enough to accommodate an indefinite number of concurrent activities. *(ii)* The combinations of concurrent activities at the test stage might be different from those at the training stage, for which the unified representation learned in training will become inadequate. To this end, we choose a *decompositional design* to represent each activity class with a dedicated representation. An intuitive implementation of such a decompositional design is to extract dedicated class-wise features using separate branches of the model. However, this approach suffers from the scalability issue, as the number of classes can increase arbitrarily in general. To address this scalability issue, we employ a shared backbone

*Authors contributed equally

for all classes at lower semantic-agnostic level (namely the postural-primitive level), and retain learning separate representation for each class at higher semantic-specific class level. The representations at postural-primitive level are decompositional by nature because they are extracted from multiple postural primitives. The two levels are connected by a weighted average pooling module which aggregates the postural primitive presentations dedicatedly for each class.

This decompositional design results in a number of *instances* for both class-wise and postural-primitive levels. For the class level, the decomposed instances refer to individual activities (*e.g*, *bend down* or *pick up trash*), and for the postural-primitive level, the decomposed instances refer to individual poses or movements (*e.g*, *the position of left hand*, or *the velocity of right arm*). Some activity classes or primitive states tend to co-occur or be mutually exclusive while others can be independent. It is thus beneficial to augment the feature representations with their contexts, so as to reduce the fuzziness. A recent work of *relation network* (Hu et al. 2018) is suitable for this purpose, in learning feature representations of interdependent instances through the *attentive modeling* of the inter-instance relationships.

In this paper, we propose a new model by extending the relation network to the spatio-temporal domain for 3D single-person concurrent activity detection problem. We propose the **Stacked Relation Networks (SRN)** which consists of two relation networks with each for a specific decomposition. The first relation network extracts feature representations for two types of postural primitives (from 3D skeletal joints and bones). The extracted primitive representations are pooled averagely according to the learned class-wise weights, so as to extract a raw representation for each activity class. We feed the raw class-wise representations into the second relation network, to augment the representation with inter-class relationships. This augmented class-wise features are finally fed into a set of class-wise LSTM classifiers to determine the occurrence of each activity class.

The contributions of this work are three-fold. *(i)* We propose a two-level decompositional approach for 3D single-person concurrent activity detection. This decompositional design makes our model general and scalable. *(ii)* The activity or primitive instances deriving from the decomposition and the inter-instance relationships can be appropriately modeled using two relation networks. We further extend the frontier of this thought to the new end-to-end trainable "stacked" relation networks, which can effectively extract expressive class-wise representations for activity detection. Relation weights produced from the relation networks are interpretable. *(iii)* The proposed stacked relation networks achieve the state-of-the-art performance on a public benchmark dataset and a newly collected dataset.

2 Related work

Our method is most relevant to single-person activity recognition methods using 3D skeleton data. We review three categories of methods from the ones with more constrained settings to less-constrained ones: *(i)* activity *classification* on trimmed sequences, *(ii)* activity *detection* from untrimmed sequences, and *(iii) concurrent* activity detection. We also review relation network used in other tasks.

Activity classification on trimmed sequences. Under the assumption that each activity interval has been pre-segmented from the streaming sequences, the task is to classify a well-trimmed video interval into one of the predefined activity classes. Due to this simplified formulation, these methods mainly focus on the design and learning of discriminative features, *e.g*, anthropometry (Du, Wang, and Wang 2015), integration of spatial and temporal cues (Liu et al. 2017), sequential dependency range adjustments (Lee et al. 2017), and spatio-temporal graph formulations (Shi et al. 2019; Yan, Xiong, and Lin 2018), *etc*.

Activity detection from untrimmed sequences. Relaxing from the activity classification on well-trimmed sequences to activity detection from untrimmed sequence introduces new challenges. However, there is still one useful assumption to make, that there exists only one activity to be classified within a time interval. Thus, the task requires not only to recognize activities but also to localize them along the temporal dimension. Existing methods achieve activity localization using several sequential models, *e.g*, RNN (Donahue et al. 2015), hidden Markov models (Wu and Shao 2014), and dynamic time warping (Zhao et al. 2013). Due to the flexibility and adaptability in modeling the long-range dependencies of streaming sequences, RNN has become the model of choice for handling untrimmed data.

Concurrent activity detection. The introduction of activity concurrency removes the single-activity constraint, that at each time a subject can perform multiple activities. There exists very few works addressing this general problem. The most relevant one is (Wei et al. 2013), which uses hand-crafted features to represent activities and their temporal relationships. Furthermore, this method relies on the sliding window to segment sequences and uses local detectors to classify activities for each interval. Notably, the model in (Wei et al. 2013) shares some spirit with our decompositional approach, in that the weight vector is learned to parameterize the importance of body parts for each activity class. Major difference between our method and (Wei et al. 2013) is that our model is deep, end-to-end trainable. Also, we utilize inter-instance relationships to augment feature representations for each instance, while (Wei et al. 2013) assumes the independence of instances deriving from the decomposition.

Other related tasks. There are other tasks that are related to concurrent activity detection. The multi-label activity detection problem (Yeung et al. 2017) provides multiple labels for each frame, but usually these "concurrent" activities are performed by multiple persons, which lacks of *semantic correlations*, *i.e*, whether two activities are mutually exclusive or not. (Lillo, Soto, and Niebles 2014; Lillo, Niebles, and Soto 2016) define the concept of complex activity which is the composition of several activities. The activity recognition is performed at the level of complex activities, so for the sake of generalization at the test stage, it requires a huge number of pre-defined complex activities due to the combination blast. This restricts the practicability significantly. On the contrary to these tasks, our target

single-person concurrent activity detection problem involves more semantic correlations, and is more practical from the perspective of generalization.

Relation network. Based on the intuition that contextual information is helpful in localizing and classifying objects, Hu *et al*(Hu et al. 2018) proposed the relation network for object detection. The integration of contextual feature is achieved using relation modules, and the importance of each piece of contextual information is estimated using an *attention* mechanism. Note that the adaption of the relation network to our problem involves significant modifications and improvements regarding various factors, *e.g*, the input, noise tolerance, temporal relationship, and the connection for two heterogeneous relation networks.

3 Method

The proposed **Stacked Relation Networks (SRN)** is a two-level pipeline (Figure 2) consists of two relation networks — one for learning postural primitive representations (RN-PP), and the other for learning class-wise representations (RN-CLS). RN-PP extracts the postural primitive representations. The input is a 3D skeleton sequence obtained from the RGB-D sensor. Specifically, we embed two types of raw postural primitive representations (from 3D skeletal joints and bones) into a high-dimensional latent space, and use RN-PP to augment the embedded representations with co-occurrence relationships between a pair of such primitives. The augmented postural primitive representations will be averagely pooled according to the learned class-wise weights, and fed into the second level as raw class-wise representations. Next, RN-CLS will augment these raw representations with inter-class co-occurrence relationships, which will then be used by the class-wise LSTM binary classifiers for activity recognition.

In RN-PP and RN-CLS, the inter-instance relationships are partially estimated by an attention module, and the influences of the contextual representations on an instance representation are parameterized by the estimated relationships. The two relation networks are connected through a weighted average pooling layer. The two relation networks share similar architectures, and their parameters are shared spatio-temporally by all instances at the same level.

We will review the preliminaries of relation network in § 3.1. The architecture of RN-PP and RN-CLS will be introduced in § 3.2 and § 3.3, respectively. Finally, § 3.4 will describe the LSTM classification and the training of SRN.

3.1 Review of relation network

The relation network was first proposed for object detection (Hu et al. 2018), with a major improvement on instance-wise representations in terms of expressive capabilities, by leveraging the co-occurrence relationships among objects. The architecture of a relation network is composed of a fully-connected layer and a relation layer in alternations. A fully-connected layer maps an instance-wise representation to a new latent space via both linear and nonlinear transformations. A relation layer augments an instance-wise representation by linearly aggregating the representations of contextual instances (*e.g*, nearby objects). The relation layer is designed to be multi-head, such that the representation augmentation benefits from the advantage of ensemble learning in flexibility.

We describe the details regarding augmenting the representation of the n^{th} instance with the representations from other instances. Let $\mathbf{f}^n$ denote the representation of the n^{th} instance, and let $\mathbf{f}^m$ denote the representation of any other instance with contextual co-occurrence relationship with the n^{th} instance. Assume that there are N^r parallel relation modules within a relation layer, and each of them outputs a relation feature $\mathbf{f}_R(n) \in \mathbb{R}^{d_f}$, which is computed as a weighted sum of the representations of other instances:

$$\mathbf{f}_R(n) = \sum_m \bar{\omega}^{mn} \cdot (\mathbf{W}_V \cdot \mathbf{f}^m), \tag{1}$$

where $\mathbf{W}_V$ is a weight matrix of a linear transformation shared among all instances. $\bar{\omega}^{mn}$ represents the *attentively estimated* relation weight between the n^{th} and m^{th} instances. $\bar{\omega}^{mn}$ is computed as a normalized attention weight: $\bar{\omega}^{mn} = \frac{\omega^{mn}}{\sum_k \omega^{kn}}$, and ω^{mn} is computed as the scaled dot-product similarity:

$$\omega^{mn} = \frac{dot(\mathbf{W}_K \cdot \mathbf{f}^m, \mathbf{W}_Q \cdot \mathbf{f}^n)}{\sqrt{d_k}}, \tag{2}$$

where $\mathbf{W}_K$ and $\mathbf{W}_Q$ are linear transformation matrices that are shared among all instances. d_k is the dimension of the linearly transformed $\mathbf{f}^m$.

Representation $\mathbf{f}^n$ is augmented by concatenating all relation features:

$$\mathbf{f}^n := \mathbf{f}^n + \text{Concat}\left[\mathbf{f}_R^1(n), ..., \mathbf{f}_R^{N_r}(n)\right]. \tag{3}$$

3.2 Learning postural primitive representations

Input. As in Figure 2, the input at time step t to the SRN pipeline is a set of 3D body skeletal points. The first step to obtain postural representation is to extract raw *joint* representations (*i.e*, positions and offsets) and *bone* representations (*i.e*, positions, directions and offsets) from these skeletal points. Since it is hard to directly measure the relation weights between these two heterogeneous raw representations, they are first embedded into a common latent space as the 'emb' steps in Figure 2 (a), and then the embedded representations are fed to RN-PP for augmentation.

Spatial relationships of the postural primitive representations are learned in the spatial relation layer (S-relation) in RN-PP via a feature augmentation process following steps in Eqs. (1, 2, 3) and pipeline depicted in Figure 2 (a). Since RN-PP is applied at every time step, the learnable parameters of S-relation, *i.e*, $\mathbf{W}_V$ in (1), and $\mathbf{W}_K$ and $\mathbf{W}_Q$ in (2), are shared across the temporal domain for the purpose of generalization. We note that the raw input skeletal data are with much lower dimensionality when compared to the augmented postural representations. However, raw skeletal data are also much noisier due to the unreliable sensor or body pose estimation, resulting in unreliable relation weights learned in (2). To address this issue, for each pair of instances, we incorporate a *learnable* weight $\hat{\omega}^{mn}$ (which is optimized during back-propagation) that is also temporally shared. The weight $\hat{\omega}^{mn}$ serves as the complementary

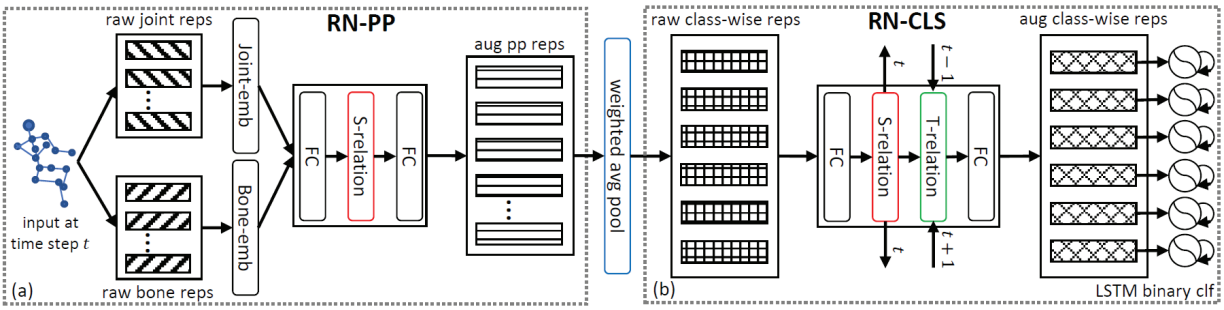

Figure 2: Pipeline overview: (a) In the first level, postural primitive representations are extracted via RN-PP. (b) In the second level, class-wise representations are extracted via RN-CLS, and afterwards a set of LSTM binary activity classifications are performed based on these representations. S-relation and T-relation represent the spatial and temporal relation layers, respectively.

base relation weight, which works hand-in-hand with the attentively estimated weight $\bar{\omega}^{mn}$ (see Figure 6 for visualizations). The postural primitive relation feature $\mathbf{f}_{p,R}(n)$ for the n^{th} instance can then be computed as the following:

$$\mathbf{f}_{p,R}(n) = \sum_m (\bar{\omega}^{mn} + \hat{\omega}^{mn}) \cdot (\mathbf{W}_V \cdot \mathbf{f}_p^m), \tag{4}$$

where $\mathbf{f}_p^m$ denotes the representation of the m^{th} postural primitive at time step t; subscript p indicates the postural-primitive level. Weight $\hat{\omega}^{mn}$ is initialized as 0. This design can effectively strengthen the robustness and flexibility of the relation module without harming performance. Given the relation features from all parallel relation modules, RN-PP outputs the augmented representation obtained from (3).

3.3 Learning class-wise representations

Input. As shown in Figure 2, the first level of SRN produces augmented representations for all N_p postural primitives $\mathbf{F}_p = (\mathbf{f}_p^1, \cdots, \mathbf{f}_p^{N_p})$, which will then be converted into a form of *raw class-wise representations* that can be taken as input to RN-CLS for activity detection in the second level. Specifically, $\mathbf{F}_p$ is averagely pooled according to a learned weight matrix $\mathbf{\Lambda} = (\Lambda^1, \cdots, \Lambda^{N_c})$, where N_c represents the number of activity classes; and $\Lambda^n = (\lambda_n^1, \cdots, \lambda_n^{N_p})$ represents the set of weights for the n^{th} class. Formally, the raw representation for the n^{th} class is computed as $\mathbf{f}_c^n = \frac{1}{N_p} \sum_m \lambda_n^m \cdot \mathbf{f}_p^m$, where subscripts p and c are used to index postural-primitive level and class level representations, respectively.

Temporal relationship. The SRN architecture can incorporate arbitrary number of relation layers in the pipeline. In Figure 2, RN-PP contains a S-relation layer, while RN-CLS contains a S-relation layer followed by a temporal relation layer (T-relation). Both RN-PP and RN-CLS share the same principled architecture. The intuition behind the T-relation layer is to augment the activity instance-wise representation with temporal contexts, so as to enhance the robustness of the representations. The T-relation layer also enforces temporal smoothness in the representation.

We provide justification on why T-relation layer is not used in RN-PP. We observed frequent fluctuations in the postural primitive features, which are noisy but might contain useful local patterns and discriminative information. Directly smoothing out of these signals can reduce representation expressibility. In contrast, learning high-level class-wise representations in RN-CLS is more global, so temporal continuity can be assumed in general.

We denote the temporal relation feature for the n^{th} class output by the T-relation layer at time t as $\mathbf{f}_{c,R}^n(t)$, which can be computed from the spatially augmented representation for the n^{th} class as:

$$\mathbf{f}_{c,R}^n(t) = \frac{1}{2}\left[\overrightarrow{\mathbf{W}}_V \cdot \mathbf{f}_c^n(t-1) + \overleftarrow{\mathbf{W}}_V \cdot \mathbf{f}_c^n(t+1)\right], \tag{5}$$

where $\mathbf{f}_c^n(t-1)$ and $\mathbf{f}_c^n(t+1)$ are spatially augmented representation for the n^{th} class at time step $(t-1)$ and $(t+1)$, respectively. $\overrightarrow{\mathbf{W}}_V$ and $\overleftarrow{\mathbf{W}}_V$ are two temporally shared matrices of linear transformations, which play the similar role as the *encoding matrices* in bidirectional RNN (Schuster and Paliwal 1997) for the previous and next hidden states.

There are two major distinctions between the computation of the spatial relation features in (4) and that of the temporal features in (5). First, to reduce model complexity and improve running efficiency, we do not explicitly model the across-time inter-instance relations in an exhaustive message-passing fashion, but implicitly impose such relationships through the spatially augmented features, *i.e*, $\mathbf{f}_c^n(t-1)$ and $\mathbf{f}_c^n(t+1)$ in (5). Second, unlike the adaptive relation weight used in the S-relation layer in (4), we employ fixed relation weight $\frac{1}{2}$ for the representations at the previous and next time steps. This is based on an assumption that consecutive information should be constantly helpful. Given $\mathbf{f}_{c,R}^n(t)$, we augment the representation for the n^{th} class by $\mathbf{f}_c^n(t) := \mathbf{f}_c^n(t) + \mathbf{f}_{c,R}^n(t)$.

3.4 Classification and SRN training

Given the spatio-temporally augmented class-wise representations $\mathbf{F}_c(t) = \left(\mathbf{f}_c^1(t), \cdots, \mathbf{f}_c^{N_c}(t)\right)$ from RN-CLS, each class-wise representation is fed to the corresponding dedicated LSTM binary classifier to predict the logits $\mathbf{Z}_t = (\mathbf{z}_t^1, \cdots, \mathbf{z}_t^{N_c})$, where $\mathbf{z}_t^n \in [-1, 1]$ represents the occurring likelihood of the n^{th} class. SRN is trained by minimizing the mean squared error between $\mathbf{Z}$ and the ground-truth logits $\tilde{\mathbf{Z}}$. Specifically, a ground-truth logit $\tilde{\mathbf{z}}_t^n = 1$ indicates that the n^{th} activity class occurs at time t, and $\mathbf{z}_t^n = -1$ indicates

the absence of the n^{th} class at time t. The objective function is given as $L = \sum_t^T \sum_n^{N_c} (\mathbf{z}_t^n - \tilde{\mathbf{z}}_t^n)^2$, which is minimized using back-propagation with ADAM optimizer and learning rate 0.001, betas $(0.9, 0.999)$.

4 Experiments

To evaluate the effectiveness of SRN, we conduct experiments on the UCLA concurrent activity dataset (Wei et al. 2013), which provides 3D human skeleton sequences with concurrent activity annotations. We further collect a larger 3D concurrent activity detection dataset, to provide both the skeletal sequences and the corresponding RGB-D videos of a wider range of concurrent activities.

Implementation details. The input of our proposed Stacked Relation Network consists two parts, *i.e*, joint postural primitive representations and bone primitive representations as shown in Figure 5 (b). The raw representation for each **joint** postural primitive consists of two components: ***(i)*** normalized joint positions and ***(ii)*** their offset between frames. To unify the format for UCLA dataset (with 20 joints per subject acquired using Kinect V1) (Wei et al. 2013) and the newly collected dataset (with 25 joints per subject acquired using Kinect V2), we keep the 20 joints tracked by Kinect V1, and ignore additional joints provided by Kinect V2. Dimension of the raw features for each joint postural primitive is 6. The raw representation for each **bone** postural primitive consists of three components: ***(i)*** the center position of the two endpoints of each bone, ***(ii)*** the unit orientation vector of the bone, and ***(iii)*** the offset of center position between frames. We take all bone postural primitives except the two foot bones which are more noisy in general. Dimension of the raw feature for each bone postural primitive is 9. After all, the dimension of input features is $20 \times 6 + 17 \times 9 = 273$.

We use $N_r = 4$ parallel relation modules within an S-relation layer. Dimension of the relation feature d_f is 64, and d_k in (2) is set as 8. We set the hidden dimension for each LSTM binary classifier as 128. Our method is implemented with PyTorch. All experiments are conducted on a machine with an NVIDIA GTX1080Ti GPU with 11GB onboard memory.

4.1 Dataset and experiment setup

UCLA concurrent activity dataset (Wei et al. 2013), to our knowledge, is the only publicly available dataset for 3D single-person concurrent activity detection. It contains 12 indoor activity classes and 61 sequences, with at most 3 concurrent activities in each frame. We follow the same experimental setting as in (Wei et al. 2013) to use sequences with even indices for training and the remaining for test.

A new concurrent activity dataset. A major shortcoming of the UCLA dataset is that the number of activity classes and amount of sequences are relatively small. To this end, we collected a larger and more challenging 3D concurrent activity detection dataset. This dataset contains 201 sequences of indoor, real-world common daily life activities with high co-occurring likelihoods. We annotated 35 indoor activity classes: *drink, eat, read book, write on paper, write on blackboard, look at monitor, use remote controller, play/read phone, make a call, pick up phone, turn on monitor, type on keyboard, fetch water, pour water, throw trash, pick up trash, tear up paper, wear jacket, take off jacket, wear shoes, take off shoes, wear on glasses, take off glasses, touch hair, clapping, wave hand, nod head, shake head, wipe face, bend down, walk, sit, stand up, jump, cross leg*. The RGB-D videos are captured in resolution 512×424 at 15 FPS. We provide three types of data sources: RGB videos, Depth videos and human skeleton sequences. Figure 3 shows two sample frames with skeletal joints overlay on the RGB-D videos. The dataset is captured at two in-door locations with multiple volunteers. The number of concurrent activities at each frame varies from 1 to 5. This dataset includes three times more activity classes and amount of sequences over the UCLA dataset. It contains subtle activities that are not easily perceivable, *e.g*, *nod head*, *shake head*. Many activities are highly correlated, *e.g*, *pick up phone, read/play phone, make a call*, *etc*. We use two thirds of the sequences for training and the remaining for test.

Figure 3: Sample frames of the new RGB-D concurrent activity dataset. The skeletons are drawn on the color images. The data is captured with varieties of human subjects, camera views and activity classes. Total number of concurrent activities and combinations may vary on different videos.

4.2 Evaluation criteria

Average precision (AP). We evaluate the *per-activity-interval* performance in terms of average precision following (Wei et al. 2013). A detected activity interval is considered to be correct if the overlap between it and a ground-truth interval is greater than or equal to 60%. We set the threshold 0.8 to obtain intervals in our experiments. We use both the *mean class-wise AP (MAP)* and *overall AP (OAP)* over all classes as metrics. The AP score is sensitive to false positives, which sets a strict requirement in the localization accuracy of the compared methods.

Error rate (ER). In addition to the above per-interval evaluation, we also evaluate the *per-frame* activity detection performance in terms of *error rate*. We use a binary vector $G_{i,j}$

Figure 4: Wrong prediction area (WPA). The plot shows a clip of ground-truth annotation and activity prediction result. Blue and red indicate false positives and false negatives, respectively. WPA is defined as the sum of false positive and false negative areas.

(1 for occurrence and 0 for absence) to denote the ground-truth annotation of the i^{th} sequence for the j^{th} activity class. We use $S_{i,j}$ to denote the corresponding prediction scores re-scaled to the range of $[0, 1]$. As illustrated in Figure 4, we define the *wrong prediction area* (WPA) as the sum of the false positive and false negative areas:

$$\text{WPA}_{i,j} = \sum_t \underbrace{(\mathbf{1} - G^t_{i,j}) \cdot S^t_{i,j}}_{\text{false positive}} + \underbrace{G^t_{i,j} \cdot (\mathbf{1} - S^t_{i,j})}_{\text{false negative}}. \quad (6)$$

ER for all sequences over all classes is computed by $\text{ER} = \sum_i \sum_j \frac{\text{WPA}_{ij}}{T_i}$, where T_i is the length of the i^{th} sequence.

4.3 Results and analysis

Compared methods. We compare SRN with three SVM-based methods (namely ALE, MIP and COA) described in (Wei et al. 2013), two LSTM-based baselines named LSTM3 and LSTM4, respectively and one state-of-the-art model on activity classification task - Adaptive Graph Convolutional Network (AGCN) (Shi et al. 2019). LSTM3 is a 3-layer LSTM with 128 hidden units for each layer, followed by a linear classifier. Compared to LSTM3, LSTM4 has larger capacity, *i.e*, 4 layers with 256 hidden units for each layer. Both LSTM3 and LSTM4 take as input a 310 dimensional vector with four types of features (joint positions, angles, offsets, pairwise joint distances) as in (Veeriah, Zhuang, and Qi 2015). While AGCN takes the same input features of SRN. To adapt AGCN to our problem setting, we changed the output of network from single label per video to multiple label per frame. The amount of trainable parameters for LSTM3, LSTM4, AGCN and SRN are 0.5M, 2.2M, 6.9M and 2.8M, respectively.

Analysis. Table 1 and Table 2 show the performance evaluation of our method and comparison to others on the two datasets. SRN achieves the best performance on all three metrics (OAP, MAP and ER) on both datasets. Table 1 shows that SRN outperforms all baselines in 5 out of 12 activity classes on the UCLA dataset in class-wise AP scores. Note that ALE, MIP and COA achieves moderate OAP, but yield poor MAP with large standard deviations. This is because these methods overfit on a few frequently occurring classes, *e.g*, *sit, stand* and *drink*, while underfit the other classes. In contrast, SRN outperforms all baselines significantly in MAP and the MAP standard deviation. It indicates that SRN is more robust on the imbalance datasets. Compared to LSTM3 and LSTM4 based on the unified fixed-size representation, SRN achieves more balanced performance. A reasonable explanation is that the decompositional design and relationship modeling of SRN can effectively increase the network representative capacity and flexibility with only moderate parameter increment. Our SRN also outperforms AGCN on both three metrics with the same input features, which demonstrates the class-wise representation learning of SRN is more effective than the unified feature. In our newly collected dataset, although OAP and ER are not as high as on UCLA dataset due to the data variances and imperceptible class differences, the performance of SRN also outperforms other methods in a certain margin. We also want to mention that SRN achieves such good performance with only a small amount parameters (on a par with LSTM4).

Table 1: Comparisons on the UCLA dataset. ↑ (↓) means the higher (lower) the better. See § 4.2 for criteria descriptions.

Activity	ALE	MIP	COA	LSTM3	LSTM4	AGCN	SRN
make a call	0.85	0.93	**0.97**	0.71	0.69	0.82	**0.97**
sit	**0.99**	0.98	0.98	0.73	0.79	0.87	0.96
stand	**0.99**	0.98	0.98	0.91	0.90	0.91	0.94
drink	0.91	0.92	**0.96**	0.95	0.92	0.90	0.90
type on keyboard	0.92	0.91	0.93	**0.99**	0.93	0.98	0.97
turn on monitor	0.55	0.42	0.43	0.86	0.84	0.88	**1.00**
fetch water	0.58	0.59	0.60	**0.99**	0.95	0.91	0.95
pour water	0.71	0.58	0.71	0.85	0.81	0.92	**0.95**
press button	0.66	0.22	0.33	0.92	0.79	0.86	**0.99**
pick up trash	0.39	0.40	0.55	0.72	0.80	0.66	**0.82**
throw trash	0.21	0.29	0.59	**0.91**	0.77	0.76	0.84
bend down	0.47	0.58	0.67	0.85	**0.89**	0.87	0.86
MAP ↑ ± std ↓	0.69 0.25	0.65 0.28	0.73 0.23	0.87 0.10	0.84 0.08	0.86 0.08	**0.93** **0.06**
OAP ↑	0.84	0.86	0.88	0.86	0.86	0.88	**0.91**
ER ↓	n/a	n/a	n/a	0.29	0.30	0.31	**0.22**

Table 2: Comparisons on the new dataset. ↑ (↓) means the higher (lower) the better. See § 4.2 for criteria descriptions.

Method	OAP ↑	MAP ↑ ± std ↓	ER ↓
LSTM3	0.56	0.44 ± **0.30**	1.24
LSTM4	0.57	0.43 ± 0.31	1.31
AGCN	0.55	0.44 ± 0.34	1.30
SRN	**0.61**	**0.48** ± 0.31	**1.20**

4.4 Ablation study

We conducted detailed ablation study to evaluate four aspects of SRN: *(i)* necessity of relation layers at each level, *(ii)* necessity of the T-relation layer, *(iii)* design of the relation weight, and *(iv)* decompositional design for the activity representation. We create several variants of our model by disabling/adding/swapping layers or exchanges component modules, and evaluate these variants on the UCLA dataset. All variants take the same input. Evaluation results are shown in Table 3, where (a) represents the standard SRN as in Figure 2, and the rest (b) to (i) are variants to compare. Specifically, the standard SRN in (a) contains one RN-PP and one RN-CLS with T-relation layer, where the relation weight is the sum of the attentively estimated weight $\bar{\omega}^{mn}$ and the learnable base weight $\hat{\omega}^{mn}$ in (4).

Relationship modeling at each level. To evaluate the effectiveness of the proposed relation modules, we keep the basic architecture settings intact (*i.e*, FC layers, weighted average pooling) but consider in Table 3 (b) a stacked network without relation layers in both RN-PP and RN-CLS,

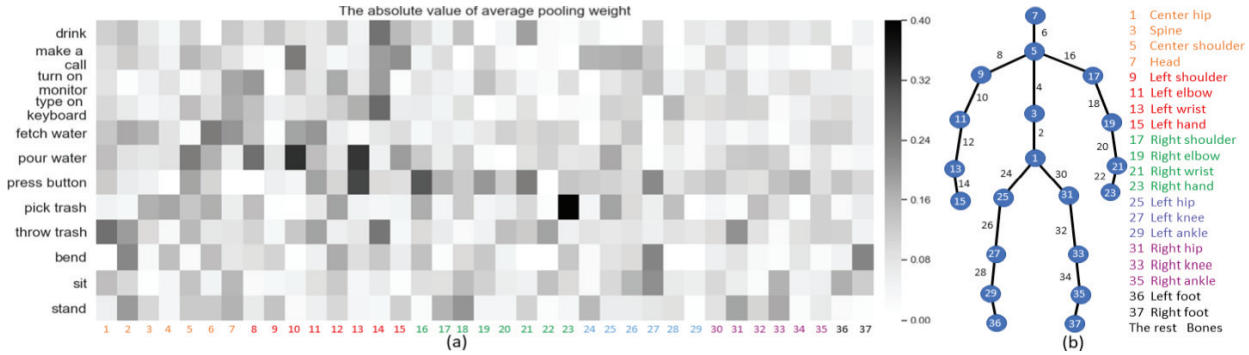

Figure 5: (a) Visualization of the absolute value of average pooling weight $\mathbf{\Lambda}$ (defined in § 3.3) on UCLA dataset. (b) Legend of skeletal joints and bones in color groups, which corresponds to the indices of the horizontal axis in (a).

Table 3: Ablation study on the UCLA dataset.

Method	OAP ↑	MAP ↑ ± std	ER ↓
(a) SRN (see Figure 2)	**0.91**	**0.93 ± 0.06**	**0.22**
(b) SRN w/o all relation layers	0.85	0.81 ± 0.14	0.27
(c) SRN w/o relation layers in RN-CLS	0.87	0.89 ± 0.10	0.23
(d) SRN w/o relation layers in RN-PP	0.85	0.82 ± 0.14	0.27
(e) SRN w/o T-relation in RN-CLS	0.88	0.87 ± 0.11	0.26
(f) SRN w/ T-relation in RN-PP	0.88	0.87 ± 0.10	0.24
(g) SRN w/o $\bar{\omega}^{mn}$ in (4)	0.88	0.88 ± 0.08	0.23
(h) SRN w/o $\hat{\omega}^{mn}$ in (4)	0.87	0.83 ± 0.10	0.26
(i) SRN w/ unified activity representation	0.85	0.86 ± 0.11	0.40

in Table 3 (c) a relation network without RN-CLS, and in Table 3 (d) a relation network without RN-PP. Observe that (c) has 2% improvements in OAP, 8% in MAP and 5% in ER than (b), while (d) has no improvement. This suggests that RN-PP can better capture information from raw input features. The postural primitive module in (d) only uses FC layers to learn representations, thus the expressibility there is insufficient to learn good enough class-wise representations. The comparison between (a) and (c) demonstrates that it is beneficial to model the inter-class relationships for learning class-wise representations.

Temporal relation. We investigate the capability of temporal relation module (T-relation in Figure 2). In Table 3 (e), no temporal relation modules are applied to neither RN-PP nor RN-CLS. In (f), T-relation is applied to both RN-PP and RN-CLS. Observe that Table 3 (a) (w/ T-relation only in RN-CLS) performs better than (e) on all metrics. Since class-level T-relation incorporates temporal context in feature learning, it makes activity predictions more immune against inconsistent spatial features. In Table 3 (f), adding T-relation to postural primitive learning causes negative effects in reducing the discriminability of the representation, which results in inferior class-wise representations.

Relation weights. We investigate two aspects of the relation weights defined in (4). In Table 3 (g), we use only the learnable relation weight $\hat{\omega}^{mn}$ to compute relation features. In (h), we use only the input-dependent attentive relation weight $\bar{\omega}^{mn}$. The results show that either $\hat{\omega}^{mn}$ or $\bar{\omega}^{mn}$ alone cannot match the use of them combined. The attentive relation weight $\bar{\omega}^{mn}$ is good at estimating instance relationships with diverse motion features. However, $\hat{\omega}^{mn}$ is input-independent and thus good at modeling global relationship among instances, which can provide some resistance to the noise incurred by the attentively estimated weights.

Decompositional design in class-wise representations. We compare the performance of the proposed decompositional design (with class-wise representation) and an unified approach (without decomposition of representation). We create a variant model based on the following modifications. First, we only maintain a unified set of learnable weights for all classes at the weighted average pooling layer, instead of maintaining a unique set of weights per class. As such, the pooling layer will output a raw unified representation. Second, we remove all relation layers within RN-CLS. Third, we replace the class-wise LSTM binary classifiers with a unified multi-class LSTM classifier. Performance drops significantly after the removal of the decompositional design in Table 3 (i) when compared with original SRN in (a). Such drop should be caused by the loss of representation capacity and poor inter-class relationship modeling.

4.5 Interpretability

Average pooling weight. As described in § 3.3 and Figure 2, raw class-wise representations are computed from averagely pooling of the postural primitive representations according to the class-wise weights $\mathbf{\Lambda}$. Therefore, the *absolute value* of $\mathbf{\Lambda}$ can reflect the influences of each postural primitive representation to each activity class. As such, we visualize the absolute value of average pooling weights in Figure 5. Entries of each row represent the learned pooling weights of all postural primitive instances for a specific activity class. The human skeletal joints and bones are visualized into five groups of colors in Figure 5 (b). Observe in Figure 5 (a) that the upper-body groups, such as left and right arms (red and green), tend to have more influence on upper-limb activities, *e.g*, *pour water*, *type on keyboard*, *etc*. The lower-body groups (blue and purple) have more influence on lower-body or full-body activities, *e.g*, *bend* and *sit*. These observations are consistent with our common knowledge. These result shows the interpretability of the learned weights, which also supports the rationality of our decompositional design.

Relation weights. The two types of spatial relation weights (namely, the attentively estimated weight $\bar{\omega}^{mn}$ and learnable weight $\hat{\omega}^{mn}$) in (4) play an important role at modeling co-occurrence relationship between activity pairs. We visualize

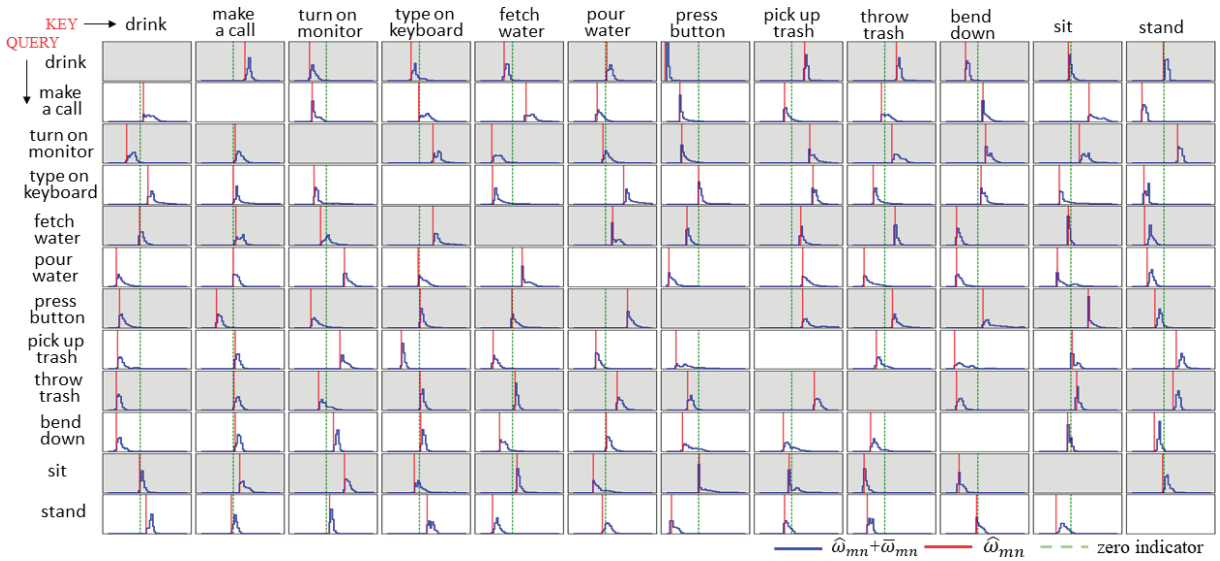

Figure 6: Distributions of the spatial relation weight at the class level on UCLA dataset. Blue curves show distributions of $\bar{\omega}^{mn} + \hat{\omega}^{mn}$ for each activity pair. Red vertical line indicates the learnable relation weight $\hat{\omega}^{mn}$. Green dotted line indicates the zero relation weight. Please zoom in for details.

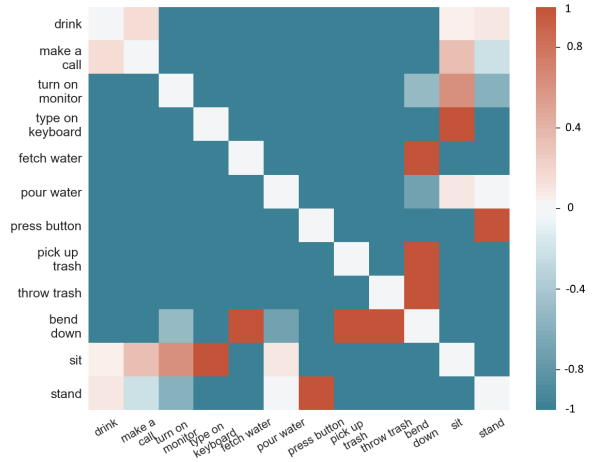

Figure 7: Statistical co-occurrence likelihood of the class pairs. Positive value (~ 1) means that the two classes tend to co-occur; negative value (~ -1) means they are statistically exclusive. Please zoom in for details.

the distributions of these relation weights in Figure 6 to investigate their interpretability. In Figure 6, each row corresponds to a query class, so we can study the influences of a specific contextual class against each query class by checking Figure 6 column-by-column. We visualize $\hat{\omega}^{mn}$ using a red line, as the value is fixed for a class pair. Note that $\bar{\omega}^{mn}$ is a normalized weight, thus the magnitude is less meaningful. We visualize $\bar{\omega}^{mn} + \hat{\omega}^{mn}$ using a blue curve to show the distribution more clearly. In general, the farther a distribution deviates from zero, the stronger the co-occurring relationship between the current class pair is. However, since $\bar{\omega}^{mn} \in [0, 1]$, the sign of $\hat{\omega}^{mn}$ is meaningful in that a positive (negative) $\hat{\omega}^{mn}$ provides the lower (upper) bound for the relationship strength.

Further insights can be obtained by comparing the relation weight distributions in Figure 6 with the statistical co-occurrence likelihoods of activities computed by coefficient of colligation (Yule 1912) in Figure 7. The co-occurrence likelihood ρ is computed as:

$$\rho = \frac{\mu_{11}\mu_{00} - \mu_{10}\mu_{01}}{\mu_{11}\mu_{00} + \mu_{10}\mu_{01}}, \tag{7}$$

where μ_{11} / μ_{00} denotes the number of occurrences where both activity class A and B get 1 (occur) / 0 (not-occur). μ_{10} denotes the counts where activity class A occurs and B does not; and *vice versa* for μ_{01}. When ρ approaches 1, A and B always co-occur. When $\rho \to -1$, A and B seldom co-occur. When $\rho \to 0$, the co-occurrence between A and B is neutral. The co-occurrence likelihood of activities can reflect the co-occurrence of two activities in three aspects, *e.g.*, *co-occurring*, *mutually exclusive*, or *neutral*. Many entries in Figure 6 match those in Figure 7 perfectly (in that the cases of $\bar{\omega}^{mn}$ and $\hat{\omega}^{mn}$ away from 0 match large absolute values in Figure 7 and vice versa). Such well-matching cases include the influences of *drink* as a key on all query classes except *fetch water*, and the influences of *press button* on all query classes except *type on keyboard* and *sit*. For the entries in Figure 6 not matching Figure 7, our model automatically learns and adopts a conservative attitude to these relationships, such that even if these contexts are not helpful for augmenting the query representation, they at least do no harm. Such a conservation strategy is reflected in two cases: *(i)* Inaccurate weights in Figure 6 are pushed to zero, *e.g*, the influences of *make a call* as a key on queries *turn on monitor, type on keyboard* and *bend down*; *(ii)* Inaccurate weights in Figure 6 are limited by a low upper bound by negative $\hat{\omega}^{mn}$ and the relationship strengths are further weakened by $\bar{\omega}^{mn}$. Examples include the influences of *stand* as key on query *pour water*, and the influences of *fetch water* as a key on query *bend down*.

5 Conclusion

In this work, we describe an end-to-end network to detect single-person concurrent activities from untrimmed 3D skeletal sequence. The core of our approach is based on an decompositional design, where the attention-guided relation network can be applied to model low-level features from body part locations/movements, and adaptively learn relations between features that enables the learning of a dedicated feature representation for each activity class. Scalability of this approach is further addressed in the proposed Stacked Relation Network (SRN). SRN achieves the state-of-the-art performance on a public dataset and a newly collected dataset. The learned relation weights of SRN are lightweight and interpretable among the activity contexts.

References

Aggarwal, J. K., and Xia, L. 2014. Human activity recognition from 3D data: A review. *PRL*.

Donahue, J.; Hendricks, L. A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Darrell, T.; and Saenko, K. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Du, Y.; Wang, W.; and Wang, L. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*.

Gao, X.; Hu, W.; Tang, J.; Pan, P.; Liu, J.; and Guo, Z. 2018. Generalized graph convolutional networks for skeleton-based action recognition. *CoRR* abs/1811.12013.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *CVPR*.

Hu, G.; Cui, B.; and Yu, S. 2018. Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention. *CoRR* abs/1811.04237.

Ke, Q.; Bennamoun, M.; An, S.; Sohel, F. A.; and Boussaïd, F. 2018. Learning clip representations for skeleton-based 3D action recognition. *TIP*.

Lee, I.; Kim, D.; Kang, S.; and Lee, S. 2017. Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks. In *ICCV*.

Li, C.; Cui, Z.; Zheng, W.; Xu, C.; and Yang, J. 2018. Spatio-temporal graph convolution for skeleton based action recognition. In *AAAI*.

Lillo, I.; Niebles, J. C.; and Soto, A. 2016. A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets. In *CVPR*.

Lillo, I.; Soto, A.; and Niebles, J. C. 2014. Discriminative hierarchical modeling of spatio-temporally composable human activities. In *CVPR*.

Liu, J.; Akhtar, N.; and Mian, A. 2017. Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. *CoRR* abs/1711.05941.

Liu, J.; Wang, G.; Hu, P.; Duan, L.; and Kot, A. C. 2017. Global context-aware attention LSTM networks for 3D action recognition. In *CVPR*.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *TSP*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2018. Non-local graph convolutional networks for skeleton-based action recognition. *CoRR* abs/1805.07694.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*.

Veeriah, V.; Zhuang, N.; and Qi, G. 2015. Differential recurrent neural networks for action recognition. In *ICCV*.

Wang, J.; Liu, Z.; Wu, Y.; and Yuan, J. 2012. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*.

Wei, P.; Zheng, N.; Zhao, Y.; and Zhu, S. 2013. Concurrent action detection with structural prediction. In *ICCV*.

Weng, S.; Li, W.; Zhang, Y.; and Lyu, S. 2019. Dual-stream CNN for structured time series classification. In *ICASSP*.

Wu, D., and Shao, L. 2014. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *CVPR*.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Yeung, S.; Russakovsky, O.; Jin, N.; Andriluka, M.; Mori, G.; and Fei-Fei, L. 2017. Every moment counts: Dense detailed labeling of actions in complex videos. *IJCV*.

Yu, G.; Liu, Z.; and Yuan, J. 2014. Discriminative orderlet mining for real-time recognition of human-object interaction. In *ACCV*.

Yule, G. U. 1912. On the methods of measuring association between two attributes. *Journal of the Royal Statistical Society*.

Zhang, X.; Xu, C.; Tian, X.; and Tao, D. 2018. Graph edge convolutional neural networks for skeleton based action recognition. *CoRR* abs/1805.06184.

Zhao, X.; Li, X.; Pang, C.; Zhu, X.; and Sheng, Q. Z. 2013. Online human gesture recognition from motion data streams. In *ACM MM*.