2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Explainable and Efficient Sequential Correlation Network for 3D Single Person Concurrent Activity Detection

Yi Wei*[1] Wenbo Li*[2] Ming-Ching Chang[1] Hongxia Jin[2] Siwei Lyu[1]

***Abstract*— We present the *sequential correlation network* (SCN) to improve concurrent activity detection. SCN combines a recurrent neural network and a correlation model hierarchically to model the complex correlations and temporal dynamics of concurrent activities. SCN has several advantages that enable effective learning even from a small dataset for real-world deployment. Unlike the majority of approaches assuming that each subject performs one activity at a time, SCN is end-to-end trainable, *i.e.*, it can automatically learn the inclusive or exclusive relations of concurrent activities. SCN is lightweight in design using only a small set of learnable parameters to model the spatio-temporal correlations of activities. This also enhances the explainability of the learned parameters. Furthermore, the learning of SCN can benefit from the initialization using semantically meaningful priors. We evaluate the proposed method against the state-of-the-art method on two benchmark datasets with human skeletal data, SCN achieves comparable performance to the SOTA but with much faster inference speed and less memory usage.**

I. Introduction

Activity recognition is useful to robots, especially in scenarios involving interactions with humans. Detecting activities *in the wild* and *on device* is challenging due to three aspects, *i.e.*, the *physical*, *semantic*, and *efficiency* issues.

For the physical aspect, detecting activities in the spatio-temporal volume is usually formulated as a complicated three-step pipeline: the localization of the person and/or body parts, the segmentation of temporal intervals of activities (which may be of variable lengths), and the classification of activity types. Several challenging factors (*i.e.*, viewpoint variations, occlusions, scale and context variability) associated with such a complex pipeline further complicate the activity detection problem. For the semantic aspect, how to conceptually define or describe an activity is often less addressed, which leads to the category confusion problem [1]. Moreover, real-world activities can occur concurrently [2] or hierarchically [3], for a single or multiple subjects. For the efficiency, it remains an open problem how to model activities' complexities in a lightweight fashion.

The activity recognition techniques have progressed greatly for the past few years in resolving the physical issues. Early works focused on the classification of a 'trimmed' video with a single activity [4], and later on, the focuses were gradually shifted to 'untrimmed' videos where the temporal segmentation of activities is required [5]. The attentions of researchers are also migrating from a single person to multiple individuals [1], from a single view to multiple views [6], and from 2D images to 3D (RGB-D) sequences [7]. Albeit these advances, most existing activity recognition methods are still based on an assumption that is not very practical, *i.e.*, one subject (a person or a group) can only perform one activity at a time. In other words, they assume that different activity classes are mutually exclusive for one subject. The same situation also exists in the existing multi-label activity detection datasets, *e.g.*, MultiTHUMOS [8]. Because the motion of a subject is usually multi-purpose by nature, this assumption reduces the practicability of the activity detection algorithms. The main goal of this work is to remove this assumption to develop a more effective and efficient method toward real-world deployment.

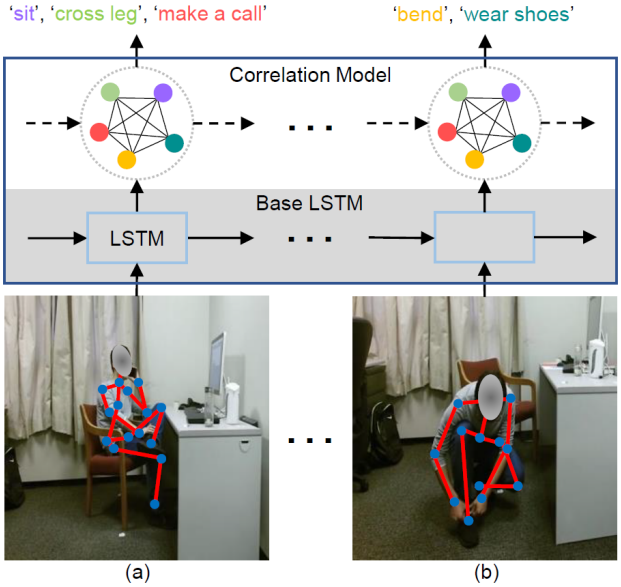

Fig. 1: We aim to robustly detect (*i.e.*, segment and classify) complex concurrent activities. Taking human pose skeletal sequences as input, the underlying activities are rich in context, fine-grained and concurrent in nature. The proposed SCN can recognize fine-grained concurrent activities via LSTM and a correlation model. It is lightweight in design, end-to-end trainable, and explainable.

In this paper, we proposed a *sequential correlation network* (SCN) model to detect the *concurrent activities performed by a single person from a streaming sequence*. For instance, in Figure 1(a), a person is sitting with leg crossed and making a phone call, where our goal is to detect all three labels ('sit', 'cross leg', 'make phone call'). We study three correlations between any two concurrent activities: positive, negative and neutral. By "positive", it means that these two activities tend to co-occur. The "negative" and "neutral" correlations represent the mutual exclusion and independence,

* Authors contributed equally.
[1]University at Albany, State University of New York, USA
[2]Samsung Research America AI Center, USA

respectively. Since we are studying the concurrent activity detection from the streaming sequence, we further extend these three correlations from the spatial domain (within the same time interval) to the temporal (in order). We note that the correlation of activities (spatially, temporally, or causally) can be a good feature to leverage to improve event detection robustness. For example, in Figure 1(b), the 'wear shoes' activity should correlate strongly with 'bend'.

SCN consists of two major components as in Figure 1: *(i)* a base LSTM for generating raw per-frame detection results, and *(ii)* a graphical correlation model to refine the raw results by considering the correlations among activity classes. In practice, we propose two types of correlation models, to handle concurrent activities that are either spatially or spatio-temporally correlated. Figure 2 illustrates the pipeline, in which the dotted arrows indicate the model choice. The spatio-temporally correlation model is designed as a recurrent architecture. But unlike most recurrent networks in which the neurons within a recurrent layer only interact with themselves along the time axis, neurons in the correlation model can interact with each other (spatially and temporally). Such fully connectivity enables efficient message passing among neurons, and thus improve the capacity to model complex correlations among concurrent activities.

We design the correlation model to be lightweight in such a way that the correlation type between any two activity classes is explained by a learnable parameter. This characteristic enables the initialization of the correlation model with the statistical prior, which boosts the trainability of SCN. In addition, the lightweight design enables the fast inference speed and efficient memory usage. SCN achieves comparable performance to the state-of-the-art method in benchmark evaluations but with minimal resource consumption.

II. Related Work

Activity recognition is a fundamental problem in computer vision. Numerous advances have been achieved in activity recognition for the past decade, see [9] for a survey. We only review the most relevant methods herein.

Concurrent activity detection. The introduction of concurrency relaxes the constraint that one subject (a person or a group) can only perform one activity at a time, and there are only a few works addressing this more general problem. Wei *et al*. [2] designed hand-crafted features to represent activities and their correlations, manually segmented sequences using sliding windows, and classify activities for each interval using a local detector. Recently, Wei *et al*. [9] proposed SRN, a two-stage relation network, for this topic, and achieves the state-of-the-art performance. The correlations of activities at both the posture level and class level are modeled by Transformer-like [10] relation layers which brings heavy computational overhead. This makes us wonder if we can design a more efficient model but with similar performance. To this end, we propose SCN which relieves the burden of the neural network in modeling the correlations. For the posture level, we relieve the burden by using correlation-encoded hand-crafted features as input, which contradicts SRN [9] that takes raw input and use relation layers to model the correlations. For the class level, we propose a lightweight correlation model to capture the correlations at the logits level, which is much more efficient than the relation layers.

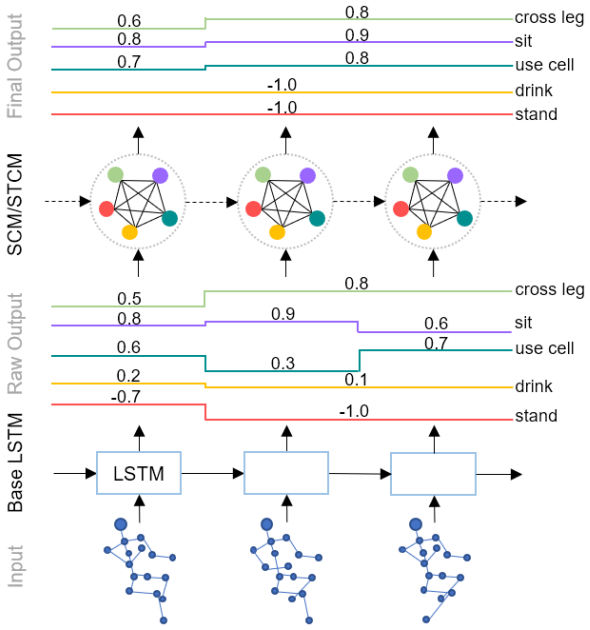

Fig. 2: Overall pipeline of SCN for concurrent activity detection, taking a human skeleton sequence as input. The noisy, unreliable multi-labeling of activities generated from the base LSTM can be refined and corrected in the proposed SCM or STCM models via message passing. Color represents individual activity class.

Graph neural network. The correlation model of SCN resembles a complete graph in which each node corresponds to an activity class, and each edge corresponds to the correlation between two activity classes. It is designed as a message passing network which belongs to the category of *graph neural network* (GNN). GNN refers to a wide spectrum of models, *e.g*., graph convolutional neural networks [11], non-local neural network [12], Transformer [10], structural RNN (SRNN) [13], message passing neural network [14], *etc*. . See [15] for a survey. Most GNNs aim to learn expressive node-wise features based on the correlations captured by the semantically-vague edges. Only a few methods model the semantically meaningful edges, *e.g*., SRNN. For the sake of explainability, the edges in our correlation model are also semantically meaningful. The differences between SRNN and our model are two-fold: *(i)* SRNN models the correlations among human, object and scene which comprise an activity, while our correlation model captures the inter-activity correlations which are higher-level; *(ii)* due to the adherence to the RNN properties, SRNN only models the temporal correlations between the corresponding neurons along the time axis, while our correlation model captures the fully correlations via a complete graph both spatially and temporally.

III. Method

Our approach is based on a hypothesis that the unstable modeling of concurrent activities by RNN can be improved

by explicitly modeling the activity correlations using a message-passing refining layer. The intuition is that errors from the state-of-the-art RNN/LSTM activity detections can be further corrected or recovered. Specifically, we propose SCN, which consists of two main modules (Figure 2): *(i)* a base LSTM taking a skeleton sequence as input and producing raw classification scores, *(ii)* a correlation model, which is implemented as a message passing layer that corrects and refines the LSTM's predictions, by leveraging the learned inter-activity correlations. We design two types of correlation models: *(i)* the *spatial correlation model* (SCM) for activities with high simultaneous correlations, and *(ii)* the *spatio-temporal correlation model* (STCM) for activities with strong spatio-temporal correlations.

The correlation model is generic that can work on top of any concurrent activity detectors. The correlation between any pair of activities can be represented by a single normalized scalar value (1 for positive, 0 for neutral, and -1 for negative correlations), which is essentially a first-order representation of correlations. The parameters of both SCM and STCM are *lightweighted*, such that (1) The SCN model is explainable by the learned parameters of SCM/STCM, via checking their semantic meanings for the activity classes, *e.g.*, 'wear shoes' and 'bend' in Figure 1 should have positive correlation weight. (2) Both SCM and STCM only induce a small amount of parameters overhead for training, thus SCN is end-to-end trainable on a small sample set, despite that concurrent activity detection is supposed to require a large training set. (3) The parameters of both SCM and STCM can be easily initialized with the statistical priors.

We review the formulation of the spatial message passing network of [16] in § III-A, which we adapt in § III-B for concurrent activity detection. We then explain in § III-C how we endow recurrence to the formulation, such that both spatial and temporal dynamic correlations can be modeled. § III-D describes how we train our model with weights initialization from the statistical prior knowledge.

A. Spatial message passing network

A MRF-like spatial network model was originally designed to model the correlations of body parts' positions in [16] for human pose estimation, which we termed the *spatial message passing network* (SMPN). Specifically, the location distribution of a part a is formulated as a marginal likelihood:

$$\bar{p}_a = \frac{1}{Z} \prod_{v \in V} (p_{a|v} \cdot p_v + b_{v \to a}), \tag{1}$$

where V represents the set of all body parts. $p_{a|v}$ represents the pairwise conditional distribution of the location of one body part to another. $b_{v \to a}$ is a bias term used to describe the background probability for the message from part v to a. Z is the partition function. When $v = a$, the message is passed from a body part to itself, where $(p_{a|v} \cdot p_v + b_{v \to a})$ represents the unary potential; otherwise, it represents the binary message.

The above formulation in [16] is implemented as a message passing network. The marginal likelihood in (1) is reformulated as an energy function by abandoning the evaluation of Z for efficiency:

$$\bar{e}_a = \exp\left(\sum_{v \in V} \left[\log\left(s(e_{a|v}) \cdot r(e_v) + s(b_{v \to a}) \right) \right] \right), \tag{2}$$

where $s(\cdot)$ and $r(\cdot)$ are SoftPlus and ReLU functions, respectively. The inclusion of the SoftPlus and ReLU maintains a strictly greater than zero output, which prevents numerical issues for the values leading into the log stage.

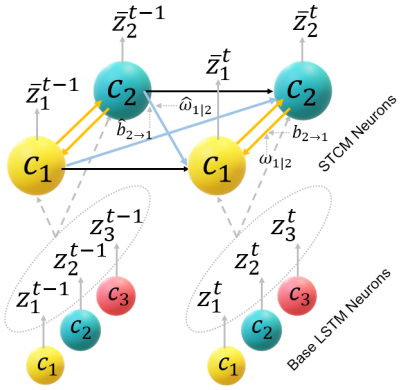

Fig. 3: STCM. Unique colors represent neurons for each activity classes. The orange neural connections represent the spatial correlation connections, and the black and blue connections represent the temporal correlation connections. Note that blue and orange connections does not exist in the RNN models.

B. SCM for concurrent activity detection

We exploit the idea of SMPN in § III-A to model inter-activity correlations within a specific temporal interval. This model is termed as *spatial correlation model* (SCM). The adaptation of SMPN to concurrent activity correlation modeling involves several issues, and we present how to resolve them in the following. *(i)* First, the pairwise condition $s(e_{a|v})$ in (2) is restricted to be greater than or equal to zero, so only the positive ($s(e_{a|v}) > 0$) and neutral correlations ($s(e_{a|v}) = 0$) are considered in this formulation. The ignorance of negative correlations in (2) is because of its applied pose estimation task. In pose estimation, a target person is supposed to have all joints, so there should not exist exclusive (or negative) correlations among different joints. However, concerning our concurrent activity detection task, there obviously tend to exist negative correlations among the activity classes of interest, *e.g.*, sit *vs.* stand. As such, the pairwise condition in our formulation is allowed to be negative. *(ii)* A negative pairwise condition might lead to the numerical issues in the log stage. Thus we replace the log and exp with $tanh$ in our formulation to keep the model parameters' range consistent. We also bound the pairwise condition by the range $[-1, 1]$ for better explainability. *(iii)* The pairwise condition $p_{a|a}$ for the unary potential $(p_{a|a} \cdot p_a + b_{a \to a})$ should always indicate the positive correlation. Thus, we disentangle the computation of unary potential from that of pairwise one, and force its "pairwise condition" $p_{a|a}$ to be 1 and bias $b_{a|a}$ to be 0.

As a result, we reformulate the marginal likelihood in (1), take the three aforementioned modifications into account, and derive the following potential function:

$$\bar{z}_a = \tanh\left(z_a + \alpha \sum_{c\in C,\, c\neq a} \left[\psi(\omega_{a|c}) \cdot z_c + b_{c\to a}\right]\right), \quad (3)$$

where C is the set of all classes. z_a is the disentangled unary potential, and we use z_a to denote the logit for activity class a which is the output of the base LSTM in SCN. $\bar{z}_a$ represents the refined logit for class a, which is output by SCM. $\omega_{a|c}$ is the pairwise condition, and $\psi(\cdot)$ is a clipping function that restricts the range of $\omega_{a|c}$ to be $[-1, 1]$. We employ α as a learnable parameter to control the overall influence of the messages passed from the other classes.

C. Spatio-temporal correlation model

Although SCM can model the correlation of activities for each time step, it does not directly model the temporal dynamics of activities. To this end, we introduce the *spatio-temporal correlation model* (STCM) by adding the recurrent connections into SCM to learn the temporal inter-activity correlations. Refer to Figure 3. In the existing RNN structures, recurrent connections within a recurrent layer only appear in-between the corresponding neurons along the time axis (black arrows). In contrast, the connections in STCM appear in-between all neurons within the same layer, such that messages can be passed along the time axis without barriers. Consequently, the temporal message passing connections (black and blue arrows in Figure 3) coupled with spatial ones (orange arrows in Figure 3) are able to encode fully spatio-temporal correlations among different activity classes. As such, we extend (3) to include the temporal potentials:

$$\begin{aligned}\bar{z}_a^t = \tanh\big(z_a^t + \alpha \sum_{c\in C, c\neq a} \left[\psi(\omega_{a|c}) \cdot z_c^t + b_{c\to a}\right] \\ + \beta \sum_{c\in C} \left[\psi(\hat{\omega}_{a|c}) \cdot \bar{z}_c^{t-1} + \hat{b}_{c\to a}\right]\big),\end{aligned} \quad (4)$$

where $\hat{\omega}_{a|c}$ and $\hat{b}_{c\to a}$ represent the parameters brought in by the temporal message passing connections and t indicates the time step. β is a learnable parameter (similar to α) that controls the overall influence of the messages passed from the previous activity classes. Note that we treat $\hat{\omega}_{a|a}$ as a pairwise condition instead of a unary one, because the previous activities and the current ones are treated separately. This endows STCM with more flexibility to detect the ending of activity intervals.

D. Training and initialization with priors

Given the predicted logits $\bar{\mathbf{Z}} = (\bar{Z}_1, \ldots, \bar{Z}_t, \ldots, \bar{Z}_T)$ for each time step, where $\bar{Z}_t = (\bar{z}_1^t, \ldots, \bar{z}_c^t, \ldots, \bar{z}_{|C|}^t)$ and $\bar{z}_c^t \in [-1, 1]$, we train SCN by minimizing the mean squared error between $\bar{\mathbf{Z}}$ and the ground-truth logits $\mathbf{Z}$. Specifically, a ground-truth logit $z_c^t = 1$ indicates that activity class c occurs at time t, and $z_c^t = -1$ indicates the absence of class c at time t. The objective function is written as:

$$L = \sum_t^T \sum_c^{|C|} (\bar{z}_c^t - z_c^t). \quad (5)$$

(5) can be minimized using back-propagation with ADAM optimizer with learning rate 0.005 and betas $(0.9, 0.999)$.

The work of [17] proposed a methodology where domain knowledge can be integrated into RNN via a low-dimensional abstract layer, to enhance the consciousness of RNN. The design of our correlation model, can be regarded as an instance of such a theory for activity detection. Because of this, the learned correlation parameters in SCN are more semantically meaningful, and they can be effectively initialized with the Bayesian statistics in a similar way as indicated in [17].

Recall in § III-B, the correlation model parameters are semantically meaningful (*i.e.*, 1 for the positive, 0 for neutral, and -1 for negative correlations). Such parameters can be initialized with the prior knowledge based on the *coefficient of colligation* [18], which represents the correlation ρ of two activities by the following formulation:

$$\rho = \frac{\mu_{11}\mu_{00} - \mu_{10}\mu_{01}}{\mu_{11}\mu_{00} + \mu_{10}\mu_{01}}, \quad (6)$$

where μ_{11} / μ_{00} represent the number of occurrences that both activity class A and B get 1 (occur) / -1 (not-occur). μ_{10} represents the times that activity class A occur and B not-occur; the case is reversed for μ_{01}. Obviously, when ρ approaches 1 (or -1), it indicates that A and B are positively (or negatively) correlated. When $\rho \to 0$, there does not exist a specific pattern regarding the co-occurrence of A and B, thus their correlation is neutral.

IV. Experiments

A. Dataset and experiment setup

We conduct experiments on the UCLA concurrent activity dataset [2] and UA concurrent activity dataset [9], which provide 3D skeleton data. The UCLA dataset contains 12 indoor activity classes, and 61 sequences in total. The UA dataset is much larger dataset which provides 35 indoor activity classes and 201 sequences.

Feature extraction is performed as in [19] from the skeletal joints. Four types of features (positions, angles, offsets, pairwise joint distances) are concatenated together to form a 310 dimension feature vector for each frame.

Implementation details. We implemented the proposed SCNs in Pytorch. The base LSTM takes the 310 dimension feature sequence as input. Initial weight for the learnable parameter α and β in (4) is set to 0.05. We followed the training and testing procedure as in [2], where sequences with even indices are used for training and the remaining are used for test. For the newly collected dataset, we take two thirds of sequences are training data and the remaining for test. All experiments are conducted on a machine with an NVIDIA TITAN-X GPU with 12GB on-board memory.

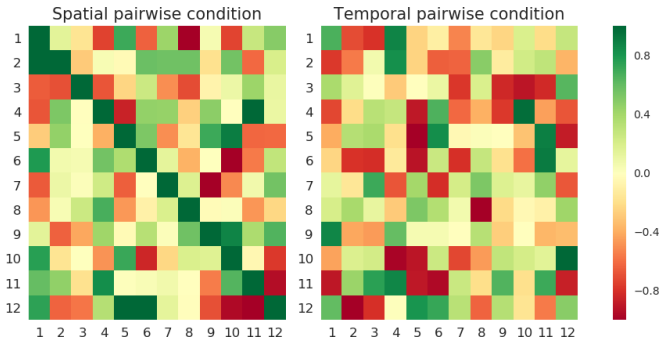

Fig. 4: Visualization of the learned pairwise condition parameters: spatial related parameters ω and the temporal ones $\hat{\omega}$ for each pair of classes extracted from the STCM model. We denote activity classes with number axis: *drink* (1), *make a call* (2), *turn on monitor* (3), *type on keyboard* (4), *fetch water* (5), *pour water* (6), *press button* (7), *pick trash* (8), *throw trash* (9), *bend* (10), *sit* (11), *stand* (12).

TABLE I: Comparisons on the UCLA dataset. ↑ (↓) means the higher (lower) the better. See § IV-B for criteria descriptions.

Activity	ALE	MIP	COA	LSTM3	LSTM4	AGCN	SRN	SCM	STCM
make a call	0.85	0.93	**0.97**	0.71	0.69	0.82	**0.97**	0.78	0.79
sit	**0.99**	0.98	0.98	0.78	0.79	0.87	0.96	0.82	0.80
stand	**0.99**	0.98	0.98	0.83	0.90	0.91	0.94	0.93	0.91
drink	0.91	0.92	**0.96**	0.95	0.92	0.90	0.90	0.94	**0.96**
type on keyboard	0.92	0.91	0.93	**0.99**	0.93	0.98	0.97	**0.99**	**0.99**
turn on monitor	0.55	0.42	0.43	0.86	0.84	0.88	**1.00**	0.92	0.90
fetch water	0.58	0.59	0.60	**0.99**	0.95	0.91	0.95	0.91	0.98
pour water	0.71	0.58	0.71	0.85	0.81	0.92	**0.95**	0.92	0.88
press button	0.66	0.22	0.33	0.92	0.79	0.86	**0.99**	0.83	0.96
pick up trash	0.39	0.40	0.55	0.72	0.80	0.66	0.82	0.73	**0.91**
throw trash	0.21	0.29	0.59	**0.91**	0.77	0.76	0.84	0.90	0.89
bend down	0.47	0.58	0.67	0.85	**0.89**	0.87	0.86	0.85	0.82
MAP ↑	0.69	0.65	0.73	0.87	0.84	0.86	**0.93**	0.88	0.90
± std ↓	0.25	0.28	0.23	0.10	0.08	0.08	**0.06**	0.08	0.07
OAP ↑	0.84	0.86	0.88	0.86	0.86	0.88	**0.91**	0.88	0.90
ER ↓	n/a	n/a	n/a	0.29	0.30	0.31	**0.22**	0.25	0.23

B. Evaluation criteria

Average precision (AP) is proposed in [2] to measure the localization accuracy for activity intervals. 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 use both the *mean class-wise AP (MAP)* and *overall AP (OAP)* over all classes as metrics.

Error rate (ER). In addition to the per-interval evaluation, we also evaluate the *per-frame* activity detection performance in terms of *Error Rate* which is proposed in [9]. It computes the false positive and false negative area among all test sequences which measures the frame-level detection accuracy.

C. Results and analysis

Compared methods. We compare the proposed SCNs – SCM and STCM with three SVM-based methods (namely ALE, MIP and COA) described in [2], two LSTM-based baselines named LSTM3 and LSTM4, respectively and one graph CNN model - Adaptive Graph Convolutional Network (AGCN) [20] and one state-of-the-art method on concurrent activity detection namely SRN [9]. LSTM3 is a 3-layer LSTM with 128 hidden units for each layer, followed by a linear classifier. Compared to LSTM3, LSTM4 has a larger parameter capacity, i.e., 4 layers with 256 hidden units for each layer. Both LSTM3 and LSTM4 take a sequence of 310 dimensional feature vector as input, the input features are described in IV-A. SCM and STCM are applying our two variants of correlation model - SCM, STCM on top of the base LSTM respectively. In this experiment, we adopt the LSTM3 as our base LSTM as its amount of parameters is not too large to cause overfitting.

TABLE II: Comparisons on the UA dataset. ↑ (↓) means the higher (lower) the better. See § IV-B 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
SCM	0.59	0.47±0.36	**1.18**
STCM	0.60	0.48±0.33	1.21

Results analysis. As shown in Table I, the two variants of SCN work well on most of the activities with high mean AP and low variation. In comparison, ALE, MIP and COA overfit on a few classes, *e.g.*, *sit, stand, drink*, which are frequently appeared in the dataset. This indicates that our methods are more robust than the other methods in handling the biased dataset. In addition, SCM and STCM outperform LSTM3 on high correlated activity classes. We argue that SCM and STCM can recover/correct some missing/wrong detections with the help of spatio-temporal inter-activity correlations. For example, STCM has a great improvement on detecting *pick up trash*, since *pick up trash* usually occurs simultaneously with or after *bend down*. STCM learns the intrinsic spatio-temporal inter-activity correlations, increase the confidence of *pick up trash* when *bend down* happens. The performances of SCM and STCM are comparable to the state-of-the-art method SRN. But SCM and STCM have less parameters which are efficient to train.

We also compares LSTM baselines, AGCN, SRN and our proposed SCM, STCM on the UA dataset in terms of OAP, MAP, ER. The SCM and STCM adopt LSTM4 as their base LSTM to fit for the increasing training data. From table II, we observe that both SCM and STCM outperform LSTM4 on OAP, MAP and ER, which is consistent with the results on the UCLA dataset. The experimental results confirm the effectiveness of our correlation model.

The SCM and STCM are two variants of the proposed correlation model regarding spatial correlations and temporal correlations. It is not guaranteed that STCM should always outperform SCM, and the selection is application dependent. If concurrent activities contain high temporal correlations, STCM should benefit from the formulation by leveraging temporal correlations.

Efficiency. The design of SCM and STCM is very lightweight. The amount of trainable parameters for LSTM3, LSTM4, AGCN, SRN and SCM/STCM are 0.5M, 2.2M, 6.9M, 2.8M and 0.5M, respectively. Given N activity classes, the parameter overhead of STCM is only $2N^2$, which is trivial compared to that of base LSTM3 (0.5M). Such lightweight design will prevent our model from severe overfitting on small dataset. Training SCM or STCM on UCLA dataset only takes 2 GPU hours with comparison to 10 GPU

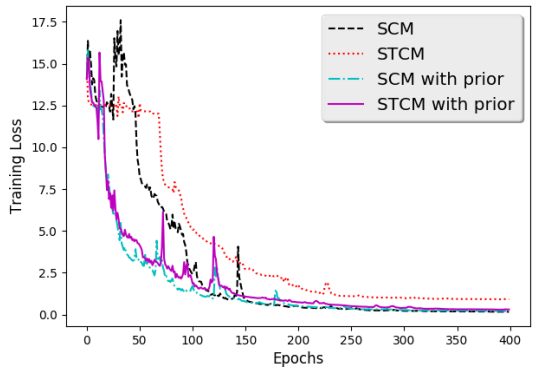

Fig. 5: Comparisons of loss convergence regarding the effect of initialization. Both SCM and STCM converge fast without performance degradation, when the prior initialization is applied.

training hours of SRN. The small parameter size enables model to be deployed on on-board devices. We also compare the inference time of SCM/STCM and SRN. SCM/STCM achieves 1200 FPS on a Intel Xeon X5570 @2.93GHz CPU, which is 12 times faster than SRN (100 FPS). The fast inference speed makes real-time activity recognition pipeline available for on-board devices considering skeleton extraction and feature pre-processing. In addition, the activity pairs with high correlations are sparse according to observations in Figure 4, thus less correlated activities will not harm the original predictions.

Initialization with prior. We evaluate the effectiveness of the proposed initialization method for pairwise condition weight on UCLA dataset. The base LSTM of SCM and SCTM adopts the same architecture of LSTM3.

Figure 5 shows the training loss degradation regarding to iterations with prior initialization, *w.r.t.* random initialization. Four models (SCM and STCM with random initialization, and SCM and STCM initialized with priors) were trained for comparison, with identical parameter settings except for the pairwise condition initialization. The results show that SCM and STCM initialized with the priors converge much faster than the random initialization case. Which demonstrate that the computed statistical priors are closer to the optimal solution and can thus accelerate the convergence speed.

Explainability. Remind that in § III we argue that the pairwise condition parameters can represent the spatial/temporal correlation between activity pairs. We visualize the learned pairwise condition parameters of STCM which is trained on UCLA dataset in Figure 4. The pairwise condition weights are assigned with different colors according to its value.

In the spatial pairwise condition matrix ω, the following class pairs (*stand, fetch water*), (*stand, pour water*), (*sit, type on keyboard*) have large positive values, which indicates positive correlations. On the contrary, the weight of the activity class pair (*sit, stand*) is close to -1, which is obvious that the two activities cannot co-occur. In the temporal pairwise condition matrix $\hat{\omega}$, *fetch water* before *pour water* is strongly correlated but *fetch water* after *pour water* is mutually exclusive, which is consistent with our prior knowledge that these two activities always happen in order. The pairwise conditions may reflect the bias of the dataset, *e.g.*, *make a call* and *drink* are strongly correlated in spatial domain since the activities always co-occur.

V. Conclusion

In this work, we present SCN that can reliably detect multiple concurrent activities from the streaming video. This novel model captures the spatio-temporal inter-activity correlations with lightweight learnable parameters. The model incorporates the advantage of RNN and graphical model by building a correlation model on top of the RNN, thus making the framework end-to-end trainable. The lightweight correlation parameters are semantically explainable. Experimental results demonstrate that SCN achieves comparable performance to the state-of-the-art methods on several datasets but with less model size and much faster inference time.

References

[1] W. Li, L. Wen, M. C. Chuah, and S. Lyu, "Category-blind human action recognition: A practical recognition system," in *ICCV*, 2015, pp. 4444–4452.
[2] P. Wei, N. Zheng, Y. Zhao, and S. Zhu, "Concurrent action detection with structural prediction," in *ICCV*, 2013, pp. 3136–3143.
[3] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *ECCV*, 2012, pp. 215–230.
[4] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *CVPR*, 2012, pp. 1290–1297.
[5] G. Yu, Z. Liu, and J. Yuan, "Discriminative orderlet mining for real-time recognition of human-object interaction," in *ACCV*, 2014, pp. 50–65.
[6] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *CVPR*, 2014, pp. 596–603.
[7] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *CVPRW*, 2010, pp. 9–14.
[8] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *IJCV*, vol. 126, no. 2-4, pp. 375–389, 2018.
[9] Y. Wei, W. Li, Y. Fan, L. Xu, M.-C. Chang, and S. Lyu, "3d single-person concurrent activity detection using stacked relation network," in *AAAI*, 2020.
[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 6000–6010.
[11] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *ICML*, 2016, pp. 2014–2023.
[12] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803.
[13] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *CVPR*, 2016.
[14] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017, pp. 1263–1272.
[15] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.
[16] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NIPS*, 2014, pp. 1799–1807.
[17] Y. Bengio, "The consciousness prior," *CoRR*, vol. abs/1709.08568, 2017. [Online]. Available: http://arxiv.org/abs/1709.08568
[18] G. U. Yule, "On the methods of measuring association between two attributes," *Journal of the Royal Statistical Society*, vol. 75, no. 6, pp. 579–652, 1912.
[19] V. Veeriah, N. Zhuang, and G. Qi, "Differential recurrent neural networks for action recognition," in *ICCV*, 2015, pp. 4041–4049.
[20] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019.