Long term spatio-temporal modeling for action detection

Makarand Tapaswi* Vijay Kumar* Ivan Laptev
Inria Paris

Abstract

Modeling person interactions with their surroundings has proven to be effective for recognizing and localizing human actions in videos. While most recent works focus on learning short term interactions, in this work, we consider long-term person interactions and jointly localize actions of multiple actors over an entire video shot. We construct a graph with nodes that correspond to keyframe actor instances and connect them with two edge types. Spatial edges connect actors within a keyframe, and temporal edges connect multiple instances of the same actor over a video shot. We propose a Graph Neural Network that explicitly models spatial and temporal states for each person instance and learns to effectively combine information from both modalities to make predictions at the same time. We conduct experiments on the AVA *dataset and show that our graph-based model provides consistent improvements over several video descriptors, achieving state-of-the-art performance without any fine-tuning.*

1. Introduction

Understanding human actions involves analyzing how a group of people interact with each other over a period of time. People are often motivated by a goal, and their actions at any time may depend on their previous actions as well as actions of other people around them. For instance, a person entering their home usually follows this sequence: first *opens* the door, *enters*, and then *closes* it; actions such as *handshake* or *kiss* require two people; or in a conversation where someone is *talking*, other people are (hopefully) *listening*. Fig. 1 presents how a scene between two characters evolves over time. We see that a woman enters the kitchen, prepares a drink and walks along with a man before sharing a drink.

The range of human activities is fairly complex and is best presented as a hierarchy. In particular, *atomic actions* [11] such as simple body movements or object manipulation (*e.g.* pour water) are composed to create complex activities that are often described using the goal (*e.g.* bake a cake). We are interested in studying the interplay between people performing such *atomic actions* as they are a fundamental aspect of visual understanding. In particular, we are interested in analyzing actions from three categories: *person movement* such as stand or walk; *person-object* manipulation such as carry (an object) or text-using-a-cellphone; and *person-person* interactions such as hug or listen-to. We study atomic actions performed by characters in videos obtained from old films.

Figure 1. Multiple frames of a video shot depicting the evolution of actions (left to right, top to bottom) between two people. We highlight the people using detection boxes and present a temporally ordered set of salient atomic actions performed by each person at the sides. Note that the characters perform multiple actions simultaneously, *e.g. enter, walk* by the woman and *stand, talk to* by the man. The original video may be viewed at https://youtu.be/CrlfWnsS7ac?t=1394.

Modeling context has an important standing in visual recognition tasks for both humans and machines [35, 47]. For example, in scene understanding, evidence of a car can help disambiguate the presence of a sidewalk [37], or presence of a blue sky may indicate that the foreground object is an airplane or a bird [7]. Classically, context has been modeled with Conditional Random Fields (CRFs) [23], but in the past few years, we see a transition towards simultaneously learning feature representations [69]. A variety of neural architectures that operate on graph structures, Graph Neural Networks (GNNs) [19, 29, 59, 62] have been introduced. A typical GNN initializes latent state vectors for each of its nodes and performs several steps of message propagation before producing node-level or graph-level outputs.

*indicates equal contribution. Work done while MT and VK were at Inria.

In this paper, we propose to jointly infer the actions performed by several people at multiple time steps with graph neural networks. We explicitly model the interactions between people appearing in a single frame and the evolution of actions for each character within a video shot (consecutive frames from the same camera). A video shot is decomposed into a graph. Each node corresponds to a short temporal segment (typically one second) for each person, and edges between these nodes represent their spatial (within keyframe) and temporal (across keyframe) relationships. We propose a novel graph message passing mechanism to encode different contextual information through explicit separation of hidden states. Our model analyzes and makes predictions for *all* characters in the shot, typically ranging from 2 to over 10 seconds.

Compared to existing works that model short-term interactions with graph networks (*e.g*. [68, 2]), our approach has following benefits. Our method jointly predicts actions performed by all people in a video shot as opposed to predictions for single person/keyframe from an arbitrarily long video input (LFB [57]) or fixed 3-sec clips [8, 10, 49]. This allows us to better model interactions and leverage long term context as opposed to short term relations. Our model is a unified framework that properly integrates both spatial and temporal context into a single graph as opposed to separate networks in [68]. To the best of our knowledge, shot-level joint prediction of multi-person actions in a unified graph framework has not been attempted before.

In summary, we make the following key contributions: (i) we propose a novel method for jointly infering actions performed by several people using Graph Neural Networks that incorporates spatio-temporal context (Sec. 3); (ii) we show that modeling and learning dependencies between people at the shot level learns long range patterns not available at the frame level; and (iii) we present thorough evaluation and ablation studies demonstrating consistent improvements for different feature backbones, achieving a new state-of-the-art performance on the challenging AVA dataset (Sec. 4).

2. Related Work

We survey related works in action recognition, methods that perform spatio-temporal action localization on the Atomic Visual Actions (AVA) dataset, and those that incorporate context. We also review works that employ GNNs, especially in the context of action recognition.

Action recognition A very popular and broad field in computer vision is classifying and localizing human actions. The goal of action classification is to predict the action category of a short video clip typically of a few seconds (*e.g*. Kinetics [4], UCF-101 [42], HMDB-51 [21]). On the other hand, action localization is more challenging and aims at localizing each action instance either spatially [11], temporally [15] or both [56]. Early action classification methods employed handcrafted video descriptors [24, 51]. However, in recent years, deep convolutional neural networks (CNN) that learn spatio-temporal patterns either with two streams for appearance and motion [8, 40, 52], or with an end-to-end 3D CNN [4, 48, 60, 8] are showing improvements in performance.

Previous methods on action detection estimate temporal boundaries directly [66], refine temporal proposals [61], or produce tracklet/frame level predictions with a post-processing module [41]. Spatial localization is achieved by detecting actors with stand-alone person detectors [8, 49] or with region proposal network (RPN) modules [9, 10, 11] in an end-to-end CNN architecture.

Methods on AVA Existing approaches on spatio-temporal action localization such as [9, 10, 11, 44, 49, 64] extend state-of-the-art video classification networks (*e.g*. I3D, often pre-trained on Kinetics [4]), for spatial localization. The typical pipeline consists of a CNN backbone that generates spatio-temporal feature maps, an actor proposal network to detect and generate actor tubes, a region-of-interest (ROI) pooling layer to extract actor features from the shared feature maps, and a final 3D tail network to classify these tubelet features.

Gu *et al*. [11] use I3D head (up to `Mixed_4e`) to make predictions without an I3D tail, while features are pooled and passed on to another CNN block in [9]. Improved tubelet linking algorithms are proposed in [26, 28]. Recently, long-range patterns in videos are exploited by creating a feature bank [57], or appearance and motion cues are modeled effectively with a two stream architecture that operates at slow and fast frame rates [8]. Our work is complementary to these as we focus on joint modeling of spatio-temporal relations between multiple people using features from pre-trained models.

Modeling context Group activity recognition in videos has gained interest in recent years [58]. Previous work in this direction jointly estimates the correlation matrix with multi-label training for individual persons [18] or uses generative models [70] to reconstruct the scene and estimate group activities. A probabilistic soft logic is also used by [32] to express actions performed by a group, where Hinge-loss Markov Random Fields are used to reason over them.

Another interesting direction in action recognition models interactions between people and objects appearing in the scene [10, 44, 49, 55, 17, 34, 16]. In the absence of object level annotations, Actor Centric Relation Network [44] considers each cell in the feature map as a potential object. The joint representation of people and objects is then used

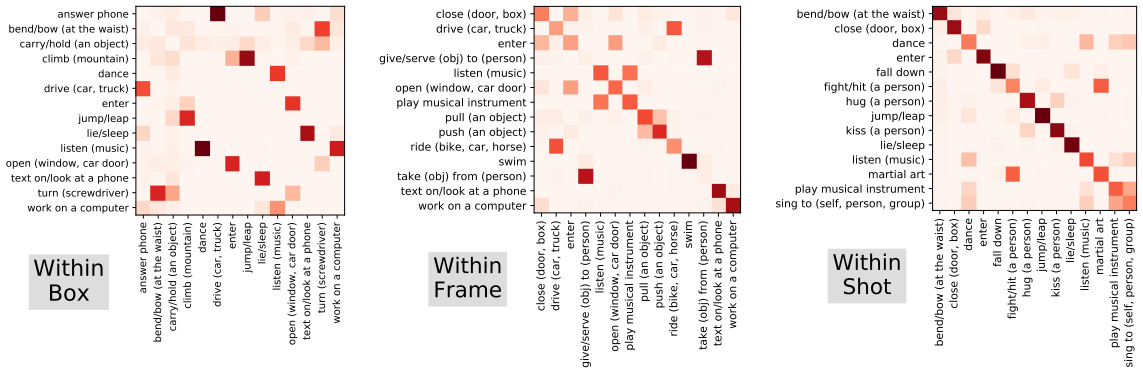

Figure 2. Normalized label correlation heat maps between atomic actions from the AVA dataset. Darker colors indicate higher correlation. From left-to-right: (a) Within **Box**: explores correlations in the multi-label setting; (b) Within **Frame**: explores actions that are performed simultaneously; (c) Within **Shot**: presents longer temporal correlations between actions by the same character. Please see the supplementary material for all actions. Best viewed on screen.

to learn interactions. More sophisticated attention modules are employed to learn such relations by Girdhar *et al*. [10] or Ulutan *et al*. [49]. Treating box proposals as queries and shared feature maps as a memory bank, a (self-)attention scheme is employed to update proposal features based on the context information containing other people and objects in the scene. Biswas *et al*. [2] combine a recurrent network on scene features with a graph neural network to model inter-person relations, however, predict actions at only one timestep. Our work is related to these approaches, but we operate on long-term spatio-temporal contexts.

Graph neural networks The basic idea in graph networks is to iteratively update the hidden representation of each node based on the aggregated information propagated from its neighborhood. Many variants of graphs networks exist in the literature that differ primarily on how the information is propagated between nodes. Node states are updated using convolutional [20] or recurrent feed-forward [29] operations on its neighborhood. Attention mechanism is also used to weight incoming messages [50].

Graph networks have been successfully applied for many computer vision tasks including image captioning [65], visual question answering [46], situation recognition [27, 43], and action recognition [38, 39, 54, 63, 68]. To recognize human actions using skeletons, spatio-temporal graphs learn the relationships between human joints in space and time [38, 39, 63]. Recurrent networks have also been applied on a graph for action forecasting by modeling spatio-temporal relations [45]. Graph (or tree) structures have been used to: discover a hierarchy of body structures for understanding actions [33]; model object-object interactions by encoding their spatial extent [14], or model human-object interactions through factors [16].

Perhaps most similar to our work, Zhang *et al*. [68] also use a graph to model spatio-temporal relations between people. Their model (a) predicts actions in a single keyframe; (b) involves a Graph Convolution Network that computes actor representations inside a 3 second window; and (c) features a separate attention-based model for person-person and person-object interactions. In contrast, we present a unified framework that (a) jointly infers actions of all actors in a video shot, (b) updates GNN states using a gated recurrent model (GRU), (c) uses dense temporal connections, and (d) effectively combines spatial and temporal information through a modified message passing mechanism.

3. Modeling Spatio-Temporal Context

We begin this section by motivating our idea based on correlations in the AVA dataset. Sec. 3.1 introduces notation and discusses how we link person detections across time. Our graph-based approach to model spatio-temporal context is presented in Sec. 3.2.

Motivation We are interested in modeling correlations between actions performed by multiple people in a video shot. To justify our goal further, we analyze correlations and dependencies among different action categories in the AVA dataset [11]. Within the multi-label setting (Fig. 2 left), people perform multiple actions simultaneously. For example, *driving* and *answering* the phone, or *lieing/sleeping* and *texting* have high correlations. Similarly, different people often perform the same (*swim*) or complementary (*give* and *take*) actions within a frame (Fig. 2 center). When analyzing how actions of different people change over the long-term, we notice a strong diagonal indicating that most atomic actions last longer than one second (Fig. 2 right), sometimes for entire shot duration. We observe correlations between certain action sequences such as *enter* and *close (door)* or *sing*, *dance*, and *play musical instrument*.

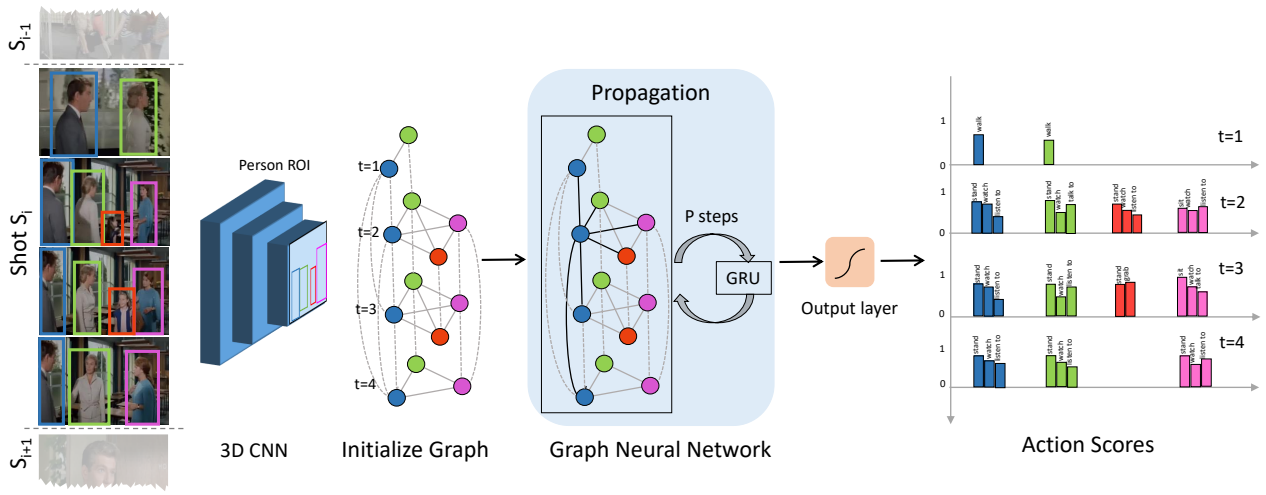

Figure 3. Overview of our proposed framework for action detection. We construct a graph for a video shot with person detections as its nodes, initialized with some visual (3D-CNN) representation, and two types of edges. Spatial edges are created between every person in a frame to capture their interactions. Relations between actions over time are learned using fully connected edges across each track. The GNN propagates information and updates the hidden representations of the nodes iteratively. An output layer uses final representations to predict actions performed by all people in the shot. Fully connected temporal edges are shown only for few nodes to avoid clutter.

Preliminaries Each video is divided into shots – contiguous frames captured from the same viewpoint. We denote person bounding boxes at a particular time t within a shot as $\mathcal{B}_t = \{b_t^j\}_{j=1}^J$, where J is the total number of boxes at time t. A feature encoding method $\Omega : \mathcal{B}_t \rightarrow \mathcal{X}_t$ (*e.g.* an I3D network [4]) is used to represent these boxes. Each person at any time b_j^t is performing multiple actions denoted as $\mathcal{Y} \subseteq \{y_1, \dots y_L\}, y_l \in \{0, 1\}$ where L is the number of action categories. In contrast to most previous work [8, 9, 28, 57], our emphasis is on developing a model $\Phi : \mathcal{X}_t \rightarrow \mathcal{Y}$, independent of various feature backbones Ω.

3.1. Linking people across time

Extending context to time requires linking person bounding boxes from the same character within a shot. Given person detections at every keyframe, we use a simple appearance based tracker to link them over time (spatial overlap may be used if detections are available at every frame). We perform tracking by associating high confidence person boxes across two consecutive keyframes successively. To learn appearance features, we train a deep Siamese CNN using the triplet loss [13] that ensures that images of the same person have a lower Euclidean distance as compared to images of different people. While training, we select a batch of person detections randomly, and compute the loss over hard positive and negative pairs for each anchor [13].

During inference, we first compute distances between all pairs of boxes in two successive keyframes and merge them in an online fashion. We solve the assignment problem using the Hungarian algorithm [22] (ensuring 1:1 mappings) and retain matches whose distance is lower than a threshold (chosen on validation set). Finally, the associated boxes are merged with active tracks while the unassociated boxes either create a new track or end an existing track.

3.2. Modeling context as a graph

Given person detections $\mathcal{B}$ at several keyframes in a shot and temporal links across these boxes, we construct an undirected graph $G = (\mathcal{N}, \mathcal{E})$. Each person detection is considered as a node, and we create two types of edges between them: (i) *spatial* edges are created between boxes (nodes) that appear in the same keyframe; and (ii) *temporal* edges connect boxes that correspond to the same person over time using the tracking described earlier. Note that these edges create fully-connected graphs in space and time respectively. Fig. 3 presents an overview of our method.

Graph Neural Network Each box (node) is associated with a localized video representation $\mathbf{x}_k \in \mathbb{R}^d$ (*e.g.* I3D), where $k = \{1, \dots, |\mathcal{N}|\}$, and $|\mathcal{N}|$ is the number of boxes in the shot or nodes in the graph.

We initialize the hidden states of each node in $\mathbb{R}^D$ as

$$\mathbf{h}_k^0 = \mathrm{ReLU}(W_{\text{init}}\mathbf{x}_k)\,, \tag{1}$$

where W_{init} embeds local region-pooled representation features $\mathbf{x}_k$ into the node state vector. Note that we expect the correlations between multiple labels associated with each box to be captured by this feature vector. Different from classical GNNs, we split the hidden state $\mathbf{h}_k^0$ into two parts: $\mathbf{h}_{k,C_s}^0 \in \mathbb{R}^{D/2}$ is reserved to represent information from the spatial context (C_s), while $\mathbf{h}_{k,C_t}^0 \in \mathbb{R}^{D/2}$ from temporal context (C_t).

Message propagation In the next step of the GNN, nodes create and send messages to their neighbors via a message propagation scheme for multiple iterations. Typically, each node of a GNN collects messages from all its neighbors. We modify this process to maintain separate spatial and temporal messages. In particular, the spatial and temporal messages accumulated by a node k at the p^{th} propagation step are

$$\mathbf{m}^p_{k,C_s} = \frac{1}{|N_S(k)|} \sum_{i \in N_S(k)} W_S \cdot \mathbf{h}^{p-1}_i \,, \tag{2}$$

$$\mathbf{m}^p_{k,C_t} = \frac{1}{|N_T(k)|} \sum_{j \in N_T(k)} W_T \cdot \mathbf{h}^{p-1}_j \,. \tag{3}$$

Here $N_S(k)$ and $N_T(k)$ is the set of spatial and temporal neighbors of node k, and W_S, W_T are linear layers that extract useful information to pass to the spatial and temporal neighbors respectively. We ignore writing biases for brevity. Note that the messages are generated by utilizing the complete hidden states $\mathbf{h}^{p-1}_i \in \mathbb{R}^D$, thus allowing communication between spatial and temporal contexts.

As the temporal window considered for atomic actions is small (one second), we see that many actions persist over several keyframes (see Fig. 2 right). To model this behavior, we load the diagonal of W_T with an identity matrix, as it encourages averaging state representations for the same person over time and empirically helps improve performance.

State update Node hidden states are updated using separate gated recurrent cells (GRU) [5, 29] for spatial and temporal contexts, whose parameters are shared at each node. They are concatenated to form the final state as

$$\mathbf{h}^p_{k,C_s} = \text{GRU}_S(\mathbf{m}^p_{k,C_s}, \mathbf{h}^{p-1}_{k,C_s}) \,, \tag{4}$$

$$\mathbf{h}^p_{k,C_t} = \text{GRU}_T(\mathbf{m}^p_{k,C_t}, \mathbf{h}^{p-1}_{k,C_t}) \,, \tag{5}$$

$$\mathbf{h}^p_k = [\mathbf{h}^p_{k,C_s}; \mathbf{h}^p_{k,C_t}] \,. \tag{6}$$

Message propagation and update steps are performed P times.

Computing action predictions We use a combination of initial $\mathbf{h}^0_k$ and final hidden states $\mathbf{h}^P_k$ to allow the model to effectively predict actions that do not need spatio-temporal context (*e.g. work on a computer*) along with actions that benefit from such context (*e.g. hug a person*). This also follows studies on including residual connections in graph networks [3]. We combine the states and max-pool across spatial and temporal cues to obtain final node representations

$$\phi_{k,C_s} = W^P_S(\mathbf{h}^P_{k,C_s} + \mathbf{h}^0_{k,C_s}) \,, \tag{7}$$

$$\phi_{k,C_t} = W^P_T(\mathbf{h}^P_{k,C_t} + \mathbf{h}^0_{k,C_t}) \,, \tag{8}$$

$$\phi_k = \max\left(\alpha_S \cdot \phi_{k,C_s}, \alpha_T \cdot \phi_{k,C_t}\right) . \tag{9}$$

Here, $\alpha_S = \sigma(W^{\text{attn}}_S \phi_{k,C_s})$ and similarly α_T are self-attention factors that trade-off importance of spatial and temporal cues, and $\sigma(\cdot)$ is the sigmoid function. Finally, the joint node representation ϕ_k is passed through a linear classifier to obtain a probability for the presence of label l,

$$p_{lk} = \sigma(W^l_{\text{cls}} \phi_k) \,, \tag{10}$$

where $W^l_{\text{cls}} \in \mathbb{R}^D$ is a linear classifier for each label $l = \{1, \ldots, L\}$.

As mentioned before, we are primarily interested in learning the mapping Φ from feature to label space. To prevent overfitting, we apply dropout after computing the initial and final states $\mathbf{h}^0_k$ and $\mathbf{h}^P_k$. We also balance the GNN messages by applying LayerNorm [1], and ensure that the ℓ_2 norm of the hidden states does not decay at each step of propagation.

Loss function and class weights As each person box is associated with several actions, we use the Binary Cross-Entropy loss to train our model. The loss is computed simultaneously on all nodes of the graph

$$\mathcal{L} = \frac{1}{L \cdot |\mathcal{N}|} \sum_{k=1}^{|\mathcal{N}|} \sum_{l=1}^{L} \alpha_l \cdot y_{lk} \cdot \log p_{lk} + (1 - y_{lk}) \cdot \log(1 - p_{lk}) \,, \tag{11}$$

where α_l are weights used to balance the positive instances ($y_{lk} = 1$) against the negative, y_{lk} is ground-truth indicating whether box k is performing action l, and p_{lk} is the predicted probability. We set $\alpha_l = \max(1, \log(|\mathcal{D}|/|\mathcal{D}_l|))$ where $|\mathcal{D}|$ is the total number of training samples, and $|\mathcal{D}_l|$ is the number of samples where action l is observed (y_{lk} = 1).

Action detection inference The joint probability of a person k with detection score $p_{\text{det},k}$ performing an action l is $p_{lk} \cdot p_{det,k}$. We find that simply multiplying the two scores performs better than selecting detections based on some threshold. This approach is used to compute evaluation metrics for all methods.

4. Experiments and Results

We first present a short overview of the dataset, followed by some implementation details. Results and discussions on a variety of ablation studies, comparison against state-of-the-art, and qualitative results are included thereafter.

4.1. AVA dataset

We evaluate our model on the "Atomic Visual Actions" (AVA) benchmark [11] for multi-person, multi-label spatial action localization. The dataset consists of 430 15-minute movie clips sourced from YouTube, and is split into 235 train, 64 validation and 131 test videos. Annotations

are available at one second intervals for all people in the keyframes. Each person is annotated with multiple action labels concerning their *pose* and interactions with other *people* and *objects*. The annotations also include person identifiers (ID)[1] that associate boxes within a short time interval. We follow the standard evaluation protocol and predict actions at every one second interval on 60 classes. Performance is reported using mean average precision (mAP) with an IoU threshold of 0.5.

4.2. Implementation Details and Context-free Baseline

Shot detection We perform shot detection on the videos by computing the displaced frame difference [67], and using a simple threshold. Examples of shots are provided in the appendix. While we detect most hard cuts (abrupt changes in viewpoint), dissolves and slow fades that are popular in older movies are missed. Nevertheless, as temporal links across such missed detections are unlikely (due to differences in appearance), the graphs are disconnected and do not suffer from misinformation.

Clip representations Our method is complementary to approaches that train or fine-tune deep CNNs [8, 9, 10, 26, 28, 57] on the AVA dataset. Specifically, we demonstrate that our model effectively captures spatio-temporal context for various representations. We evaluate our method on several features extracted from publicly available models: **(i) preI3D**: an I3D model, pre-trained on the Kinetics dataset [4]; **(ii) I3D**: a model fine-tuned on AVA, provided by Ulutan *et al*. [49]; **(iii) ACAM**: an I3D model that has been extended to incorporate scene context via attention [49]; and **(iv) SlowFast**[2]: a state-of-the-art video representation model fine-tuned on AVA [8].

We extract a box feature $\mathbf{x}_k$ using 3D CNNs that implicitly analyze durations longer than a single frame. For models based on I3D, we follow [49] and represent the video clip around the keyframe using I3DHead. We compute box representations by ROI-Pooling these video features followed by an I3DTail. To each box feature, we append a max-pooled frame representation to capture global frame context. For SlowFast, we extract features from the second-last layer, and use ROI-Align to represent boxes. We notice that SlowFast box features do not see additional benefits from scene context.

As we will show, our model is able to achieve consistent improvements in performance without requiring heavy computational efforts of further fine-tuning the backbone CNNs. Training our model only takes a few hours on a single TitanX GPU.

[1]Only for train and validation sets. We need tracking for evaluation.

[2]SlowFast Model Zoo: R101, Kinetics 600, 8x8, 28.2 mAP on val v2.1.

Person detection For a fair comparison, we use person detections provided on the AVA dataset by Wu *et al*. [57]. The detector is a Faster-RCNN model [36] with ResNeXt-101-FPN [30] backbone pre-trained on ImageNet [6] and COCO [31] datasets and fine-tuned on AVA bounding boxes. The detections achieve 93.9AP @0.5IoU on the validation set [57]. We extract features on the predicted person boxes and use them for evaluation as well as training. For each ground-truth box, we assign its label to a predicted detection with highest IoU overlap. This ensures that bounding boxes are consistent across train and test.

Temporal linking of person detections We fine-tune a ResNet-50 [12] CNN to compute feature representations for person boxes in consecutive frames. The input is a person detection crop resized to 160×80. The embedding layer has an output dimension of 128, and is obtained through a linear layer after the global average pooling. We adopt hard-negative mining within a batch for computing the triplet loss, and use a margin of 0.5 [13]. We fine-tune the CNN with the Adam optimizer for 25 epochs, a batch size of 50, an initial learning rate of 0.0002 decreased by a factor of 0.1 after 15 and 22 epochs. While linking, we associate only those boxes that have a similarity score greater than $\tau = 0.35$. Our temporal links on the AVA validation set obtain 96.4% precision and 73.5% recall.

Context-free baseline As a baseline that does not use any context, we propose to use a two-layer perceptron (MLP) to predict the actions performed by each person independent from one another. Specifically, for any box with feature $\mathbf{x}_k$, we compute the probability for action class l as

$$p_{lk} = \sigma(W^l_{\text{cls}} \cdot \text{ReLU}(W_{\text{box}}\mathbf{x}_k)) \,. \tag{12}$$

Action detection Both the MLP and our model use a hidden dimension of 512. We use a batch size of 50 for the MLP corresponding to 50 individual person detections, while a batch size of 5, corresponding to all people appearing in 5 video shots is used for the Spatio-Temporal GNN. During training, we construct batches with video shots limited to a maximum duration of 10 seconds. However, our model can be applied on entire shots at test time. We find that the number of GNN propagation steps $P = 2$ works well for our task. We train the models for 250K iterations with SGD, a learning rate of 0.01, and adopt multi-step decay by a factor of 0.1 after 150K and 200K iterations. We implemented our model in PyTorch and will release the code and shot boundaries.

Computational costs We report wall clock time for our Pytorch 1.0.1 implementation on an Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz and a GTX Titan X (12 GB RAM)

GPU. One training epoch takes roughly 200s (50s to load pre-computed features and 150s for the forward and backward passes). During validation, our model loads necessary features and computes predictions over all graphs in about 27s, averaging ∼3ms for each video shot (graph).

4.3. Results: Ablations and Comparisons

We conduct several ablation studies to justify the proposed modifications to GNNs. We also show improvements obtained at various levels of contextual information. *GNN Spatial* (GNN-S) considers inter-person interactions in one keyframe (a 3s feature extraction window); *GNN Temporal* (GNN-T) considers long-term temporal context across the entire video shot; and *GNN Spatio-Temporal* (GNN-ST) considers both within-keyframe and across-time contexts. Unless mentioned otherwise, all the ablation studies are conducted with SlowFast features [8] on the AVA v2.1 validation set. Finally, in this section, we compare our model against state-of-the-art and present some qualitative results.

Learning details for AVA In Table 1, we show the impact of two modifications that improve performance for both the baseline and our final model (GNN-ST). We adopt both these modifications for all following experiments. **(i) Det*Act:** Person detections are often filtered using a low threshold to obtain a high recall on action detection. However, we noticed that such low confidence boxes present varying action scores that degrade overall performance. While [8, 57] only keep detections > 0.9, computing a joint detection and action probability provides a considerable performance improvement. **(ii) PosW:** refers to the class balancing weights for the positive class (α_l) in the loss function. As the dataset is highly imbalanced, we observe noticeable improvements using the weights during learning.

Table 1. Ablation study showing the impact of learning details that improve both our baseline and proposed model.

Det*Act	PosW.	MLP	GNN ST
-	✓	27.19	28.31
✓	-	26.46	28.88
✓	✓	**28.13**	**29.61**

Effect of propagation steps Our GNN models propagate messages and update hidden node representations P times. We study the effect of P on our models in Fig. 4. Since our graphs are fully connected in both space and time, we observe that $P = 2$ is a good choice.

Variations in the GNN architecture We perform experiments to measure the impact of edge connections and separate hidden state representations for spatial and temporal contexts (see Table 2). Comparing row 2 with row 5 suggests that having separate hidden states (spatial and temporal) that are combined through attention is beneficial. In fact, without this separation, we see that temporal context dominates spatial, and they do not complement each other (Table 2 row 2 is comparable to Table 3 row 3). Furthermore, the obtained benefits are not merely due to the increased model capacity (Table 2 row 3, $2D$ has twice the capacity). Secondly, fully connected temporal edges help improve message flow compared to connections between adjacent nodes (see Table 2 rows 1 vs. 2, and 4 vs. 5).

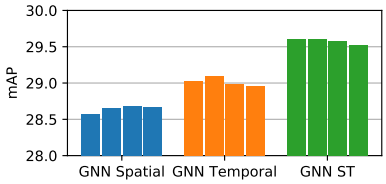

Figure 4. Effect of varying the number of GNN propagation steps. $P = [1, 2, 3, 4]$, corresponding to the four bars for each model. We choose $P = 2$ for all other experiments.

Table 2. Ablation study for the GNN Spatio-Temporal model showing the importance of: (i) fully-connected temporal edges vs. connecting adjacent keyframes, and (ii) explicit modeling of hidden states vs. combining information from spatial and temporal context into one hidden state.

	Fully-Connected Temporal Edges	Separate Hidden States	val v2.1 mAP I3D	val v2.1 mAP SlowFast
1	-	-	21.11	29.06
2	✓	-	21.37	29.08
3	✓	$2D$	21.44	29.03
4	-	✓	21.70	29.37
5	✓	✓	**22.00**	**29.61**

Table 3. Our model consistently improves performance independent of features or detections. Results on multiple features with boxes by [57] (AVA val v2.1) and SlowFast features with ground-truth boxes from v2.1 and v2.2. GNN Spatial is comparable to [68].

	Detections	FRCNN [57]				gt-v2.1	gt-v2.2
	Features	preI3D	I3D	ACAM	SlowFast	SlowFast	
1	MLP	12.76	20.16	21.33	28.13	33.74	34.21
2	GNN-S	13.04	20.77	21.84	28.65	34.42	35.00
3	GNN-T	14.31	21.39	22.36	29.09	35.06	35.32
4	GNN-ST	**14.51**	**22.00**	**22.58**	**29.61**	**35.24**	**35.69**

Consistent improvement across features and detections The first set of columns in Table 3 presents results using alternative feature backbones with person detections provided by Wu *et al*. [57]. The second set of columns shows performance with ground-truth detections. The last column reports results for AVA version v2.2.

Table 4. Ablation study with SLOWFAST features showing the performance improvement for each action group defined by the AVA dataset. Numbers indicate mAP (%). The last column corresponds to numbers shown in Table 3.

	Model	Pose	PObj	PPer	Overall
1	MLP	48.76	19.46	28.75	28.13
2	GNN Spatial	49.41	19.61	29.94	28.65
3	GNN Temporal	50.17	20.10	29.99	29.09
4	GNN Spatio-Temporal	**50.99**	**20.25**	**31.03**	**29.61**

Across the rows of Table 3, we compare our models against the MLP baseline. Note that the results indicated in MLP match their original papers. Rows 2 and 3 show the benefits of spatial and temporal context as compared to the MLP and indicate that long-range temporal context outperforms spatial context within a keyframe. Row 4 shows that our model effectively combines the two contexts to achieve a consistent 1.5-1.8% absolute gain in mAP, on a variety of features and detections.

Person-person interactions see largest improvement The action labels in the AVA dataset stem from three main categories: (i) *Pose* groups actions related to person pose, (ii) *PObj* groups actions related to person-object interactions, and (iii) *PPer* groups actions related to person-person interactions. We present improvements by our models in each of these three categories in Table 4. Overall, we observe larger improvements for person-person (+2.3%) and person-pose (+2.2%) action classes, and less gain for person-object actions (+0.8%). This is consistent with our expectations since our model is specifically designed to handle multi-person actions and interactions performed in groups, *e.g. martial art*.

The number of training samples differs greatly for classes from the AVA dataset – few hundred to hundreds of thousand instances. We observe higher gains for classes with more training samples, *i.e.* +2% on average for classes with >1K samples vs. +0.9% for others. This further indicates the potential for improvement if our model is trained with additional data, for example, using the recently released annotations on the Kinetics dataset, AVA-Kinetics [25].

Structured Model [68] Previous work on using context works within a 3-second window. Actor representations are obtained by pooling features within that window, and action predictions are made by attending to other people and objects in the keyframe. As the code for [68] is not available, and owing to differences in detections and backbones, we are unable to reproduce results. [68] is comparable to *GNN Spatial* (Table 3 row 2) that also uses temporally pooled features, followed by message passing on nodes within the keyframe. Different from [68], the goal of our paper is to incorporate long-term temporal context over the course of a video shot.

Table 5. Increase in performance in mAP (absolute %) based on shot duration (short = 1-5s, long = 6-10s).

Duration	GNN-T/MLP	GNN-ST/MLP	GNN-ST/GNN-S
Short	0.94	1.27	0.95
Long	**1.34**	**2.19**	**1.40**

Table 6. Impact of automatic vs. ground-truth detections and tracks.

Tracking	GNN-T		GNN-ST	
	v2.1	v2.2	v2.1	v2.2
Automatic	35.06	35.32	35.24	35.69
Ground-truth	35.04	35.32	35.25	35.69

Table 7. Top-5 action classes with improvement (mAP %) over the MLP.

GNN Spatial		GNN Temporal		GNN Spatio-Temporal	
close (door/box)	4.6	sing to	6.1	swim	7.1
listen to (person)	2.9	close (door/box)	4.8	martial art	4.7
martial art	2.6	martial art	3.5	dance	4.3
write	2.1	play music instr	3.1	play music instr	4.2
take obj. from per.	2.0	hug (person)	3.0	sing to	4.2

Shot duration To analyze whether our model truly benefits from long-term context, we separate shots based on their durations into two groups: short (1-5s) and long (6-10s). Table 5 shows the difference in performance between several pairs of models, *i.e.* the gain in performance for GNN Spatio-Temporal against the MLP baseline is 2.19% for longer shots, and 1.27% for shorter ones. Across all models, we see that modeling long-term context in shots with multiple keyframes shows higher improvement.

Performance on ground-truth boxes The quality of person detections play a large influence in action detection performance on the AVA dataset. While we use detections provided by Wu *et al.* [57] for a fair comparison, we also presented results on ground-truth detections in the last two columns of Table 3. A large gap between results with ground-truth (GT) vs. predicted boxes hints towards the complexity due to localization.

In Table 6, we further analyze the influence of our automatic tracking method. In particular, our approach results in reliable tracks as we see negligible differences in performance while using ground-truth or automatically predicted tracks.

Per-class accuracy We show the performance improvement for each action label as a bar graph in Fig. 5. In Ta-

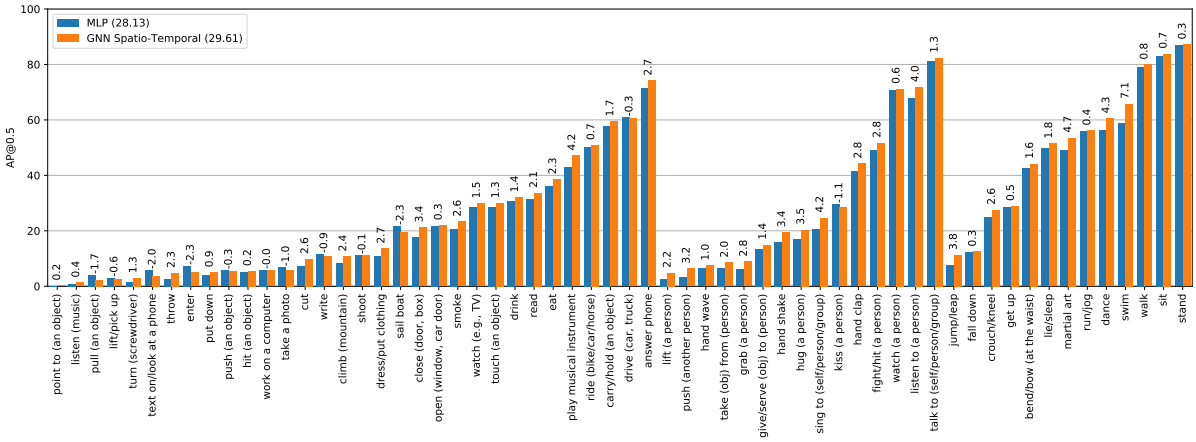

Figure 5. Per class performance comparison between the MLP and our GNN Spatio-Temporal model. Columns are sorted and categorized based on AVA action groups: *person-object* interactions, *person-person* interactions, and *person-pose* related actions.

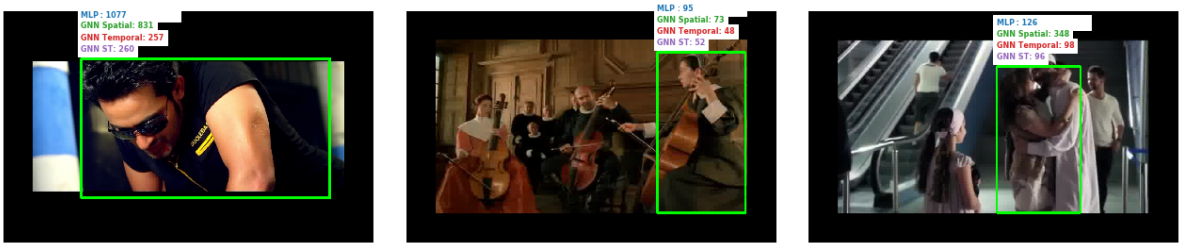

Figure 6. Keyframes with action labels *bend/bow*, *play musical instrument*, and *hug a person* where our GNN Spatio-Temporal model is able to use the context to achieve a better prediction rank (see discussion). Note that we present the improvement in rank for a single action, for a single person to improve clarity. Our models predict all actions of all actors in the video shot at once.

ble 7, we list the top 5 actions that benefit most with different kinds of context. Our spatial model benefits from the presence of other people in the scene performing similar (*martial art*) or complementary actions (*listen to*). For our temporal model, we observe improvements for actions that last longer (*e.g. sing*, *play instrument*), but also actions that follow a sequence (*e.g. close (a door)*). We see a combination of these factors help our spatio-temporal model.

Comparison with state-of-the-art We report performance of our model with state-of-the-art (SOTA) approaches in Table 8. Different choices for CNN architectures and different detectors make direct comparison difficult in most cases. We also note that advances in video representation models (LFB [57] and SlowFast [8]) can benefit multi-label spatial action localization. While several methods [10, 44, 49, 68] have used context at varying degrees, most of them are limited to a short temporal span covered by modern 3D CNNs. When applied with features such as SlowFast [8] (28.2% v2.1, 29.0% v2.2), our model achieves a new state-of-the-art on the AVA benchmark (29.6% v2.1, 30.0% v2.2).

Table 8. State of the art comparison on AVA dataset. FRCNN: Faster RCNN [36] detections; RPN: region proposal network; Inc-V1: Inception-V1; NL: non-local network [53]; 2S: two-stream, S3D: gated separable 3D network [60]; GCN: Graph Convolutional Network [20]; GNN ST (Ours) is on features from SlowFast model trained on v2.1.

Method	Backbone	v2.1		v2.2	
		val	test	val	test
Single frame	FRCNN R-50	14.7	-	-	-
AVA	I3D, FRCNN R-50	15.6	-	-	-
Better Baseline	I3D Inc-V1, RPN	22.8	21.9	-	-
Deformable Tubes	3D R-50, RPN, DTRN	25.8	-	-	-
LFB	I3D-R101-NL, FRCNN-101	27.7	27.2	-	-
SlowFast Networks	SlowFast R-101 8x8, NL	28.2	-	29.0	-
Methods that explicitly use spatial or temporal context					
ACRN	S3D, R-50	17.4	-	-	-
Action Transformer	I3D Inc-V1, Transf.	25.0	24.9	-	-
Recurrent Tubes	2S VGG-16, LSTM	21.9	-	-	-
Structured Graphs	I3D-R50-NL, GCN	22.2	-	-	-
ACAM	I3D Inc-V1, FRCNN R-101	22.7	-	-	-
GNN-ST (Ours)	SlowFast R-101 8x8, NL	**29.6**	-	**30.0**	**29.9**

Qualitative Results We visualize some results from our models in Fig. 6. As the evaluation metric is mAP, we com-

pare the rank at which a positive sample (when action is in ground-truth) appears for that class. We present ranks for the baseline MLP, and our models: GNN Spatial, GNN Temporal, and GNN Spatio-Temporal. Note that lower rank is better. The figure shows increase in rank for a single action and for a single actor to improve clarity, however, note that our models predict actions for all people over the entire video shot. Additional examples are in the supplementary material.

5. Conclusion

We presented an approach to jointly model and predict actions performed by several people in a video shot. In particular, a graph structure was used to represent person detections in keyframes, and fully connected spatial and temporal edges were used to model context. We presented a Graph Neural Network that explicitly models spatial and temporal contexts by tracking the states separately, performs a couple steps of message propagation, and produces outputs by effectively combining the contextual information. We performed experiments on the Atomic Visual Actions (AVA) dataset, achieving consistent gains in performance across multiple video descriptors and detection types, and set a new state-of-the-art on the dataset.

Acknowledgements This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), the Louis Vuitton ENS Chair on Artificial Intelligence and the DGA project DRAAF.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. *arXiv:1607.06450*, 2016.

[2] S. Biswas, Y. Souri, and J. Gall. Hierarchical Graph-RNNs for Action Detection of Multiple Activities. In *International Conference on Image Processing (ICIP)*, 2019.

[3] X. Bresson and T. Laurent. Residual Gated Graph ConvNets. *arXiv:1711.07533*, 2017.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An Empirical Study of Context in Object Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[8] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *International Conference on Computer Vision (ICCV)*, 2019.

[9] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. A better baseline for AVA. *arXiv:1807.10066*, 2018.

[10] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video Action Transformer Network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[14] R. Herzig, E. Levi, H. Xu, H. Gao, E. Brosh, X. Wang, A. Globerson, and T. Darrell. Spatio-Temporal Action Graph Networks. In *Workshop at International Conference on Computer Vision (ICCV-W)*, 2019.

[15] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The THUMOS challenge on action recognition for videos in the wild. *Elsevier Computer Vision and Image Understanding (CVIU)*, 2017.

[16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Joint learning of object and action detectors. In *International Conference on Computer Vision (ICCV)*, 2017.

[18] S. Khamis and L. S. Davis. Walking and Talking: A Bilinear Approach to Multi-Label Action Recognition. In *CVPR Workshops*, 2015.

[19] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[20] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[21] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A Large Video Databasefor Human Motion Recognition. In *International Conference on Computer Vision (ICCV)*, 2011.

[22] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.

[23] J. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning (ICML)*, 2001.

[24] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[25] A. Li, M. Thotakuri, D. A. Ross, J. Carreira, A. Vostrikov, and A. Zisserman. The AVA-Kinetics Localized Human Actions Video Dataset. *arXiv:2005.00214*, 2020.

[26] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei. Recurrent tubelet proposal and recognition networks for action detection. In *European Conference on Computer Vision (ECCV)*, 2018.

[27] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler. Situation recognition with graph neural networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[28] W. Li, Z. Yuan, D. Guo, L. Huang, X. Fang, and C. Wang. Deformable Tube Network for Action Detection in Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[32] B. London, S. Khamis, S. H. Bach, B. Huang, L. Getoor, and L. S. Davis. Collective Activity Detection using Hinge-loss Markov Random Fields. In *CVPR Workshops*, 2013.

[33] S. Ma, J. Zhang, S. Sclaroff, N. Ikizler-Cinbis, and L. Sigal. Space-Time Tree Ensemble for Action Recognition and Localization. *Springer International Journal of Computer Vision (IJCV)*, 126:314–332, 2018.

[34] P. Mettes and C. G. M. Snoek. Spatial-Aware Object Embeddings for Zero-Shot Localization and Classification of Actions. In *International Conference on Computer Vision (ICCV)*, 2017.

[35] D. Parikh, C. L. Zitnick, and T. Chen. Exploring Tiny Images: The roles of appearance and contextual information for machine and human object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(10):1978–1991, 2012.

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[37] R. Shetty, B. Schiele, and M. Fritz. Not Using the Car to See the Sidewalk - Quantifying and Controlling the Effects of Context in Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[38] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Skeleton-based action recognition with directed graph neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[41] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv:1212.0402*, 2012.

[43] M. Suhail and L. Sigal. Mixture-Kernel Graph Attention Network for Situation Recognition. In *International Conference on Computer Vision (ICCV)*, 2019.

[44] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *European Conference on Computer Vision (ECCV)*, 2018.

[45] C. Sun, A. Shrivastava, C. Vondrick, R. Sukthankar, K. Murphy, and C. Schmid. Relational Action Forecasting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[46] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[47] A. Torralba, K. P. Murphy, and W. T. Freeman. Using the Forest to See the Trees: Exploiting context for visual object detection and localization. *Communications of the ACM*, 53(3):107–114, 2010.

[48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[49] O. Ulutan, S. Rallapalli, M. Srivatsa, and B. Manjunath. Actor Conditioned Attention Maps for Video Action Detection. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[50] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. *International Conference on Learning Representations (ICLR)*, 2018.

[51] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[52] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, 2016.

[53] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] X. Wang and A. Gupta. Videos as Space-Time Region Graphs. In *European Conference on Computer Vision (ECCV)*, 2018.

[55] Y. Wang and M. Hoai. Pulling actions out of context: Explicit separation for effective combination. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[56] P. Weinzaepfel, X. Martin, and C. Schmid. Human Action Localization with Sparse Spatial Supervision. *arXiv:1605.05197*, 2016.

[57] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[58] L.-F. Wu, Q. Wang, M. Jian, Y. Qiao, and B.-X. Zhao. A Comprehensive Review of Group Activity Recognition in Videos. *International Journal of Automation and Computing*, 18:334–350, 2021.

[59] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A Comprehensive Survey on Graph Neural Networks. *arXiv:1901.00596*, 2019.

[60] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision (ECCV)*, 2018.

[61] H. Xu, A. Das, and K. Saenko. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *International Conference on Computer Vision (ICCV)*, 2017.

[62] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[63] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[64] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. Davis, and J. Kautz. STEP: Spatio-Temporal Progressive Learning for Video Action Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[65] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *European Conference on Computer Vision (ECCV)*, 2018.

[66] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[67] Y. Yusoff, W. Christmas, and J. Kittler. A Study on Automatic Shot Change Detection. *Multimedia Applications and Services*, 1998.

[68] Y. Zhang, P. Tokmakov, C. Schmid, and M. Hebert. A Structured Model For Action Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[69] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[70] Z. Zhou, K. Li, X. He, and M. Li. A Generative Model for Recognizing Mixed Group Activities in Still Images. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Table 9. Hyperparameters used in different stages of our model

Pre-processing		Temporal linking		Action detection: Graph Neural Network	
Shot detection threshold	7	Similarity threshold	0.35	#Propagation steps	2
		Training time	25 epochs	Training time	250,000 iterations
		Optimizer	Adam	Optimizer	SGD
		Scheduler	$\times$0.1 at 15, 22 epochs	Scheduler	$\times$0.1 at iter 150K, 200K
		Triplet loss margin	0.5	# propagation steps (P)	2
				Graph hidden state (D)	1024

A. Hyperparameters

Table 9 presents a summary of all hyperparameters used in this work.

B. Qualitative Results

We present qualitative results for predicting actions using SlowFast features and ground-truth person boxes as we use detections from previous work. We compare the rank at which a positive (action is in ground-truth) sample appears for each class as the primary metric for evaluation is mean average precision. We present ranks for our baseline MLP, and all our models: GNN Spatial, GNN Temporal, and GNN Spatio-Temporal. Note that lower rank is better. As indicated in the main paper, note that our models predict actions for all people over the entire video shot, even if the figure shows increase in rank for a single action and for a single actor. This is done to improve clarity.

We show examples of **success** and **failure** cases for six classes from each AVA action group – Fig. 7 for *person movement* or *pose* classes; Fig. 8 for *person-object interactions*; and Fig. 9 for *person-person interactions*. Columns 1 and 2 show improvement in rank, while column 3 shows the failure cases where the rank increases. Most failure cases correspond to poor image resolution, presence of only a single person in the frame, or are often dominated by other action labels.

C. Data Insights

We present a few example shot boundaries and complete label correlation matrices that provide additional insights into the dataset used in this work.

Example shot detections. Fig. 10 shows example of shot detection on the AVA videos. In most cases we are able to detect the shot boundary precisely. The average shot duration is 5.46 seconds. We will release these annotations for future works.

In the figure, we show a temporal span of several YouTube videos (id indicated above), with keyframes that belong to the same shot. The temporal extent of each shot is indicated in seconds below it. The last row shows a failure case where a smooth *dissolve* makes it hard to find the abrupt *cut* transitions typically used to create video shots. Nevertheless, as our temporal links are unlikely to join these people across a failed boundary, we operate on spatio-temporal graphs that only span the extent of each shot.

Label correlations. Fig. 11 shows normalized correlations between labels from all classes. These are similar to Fig. 2 of the main paper that showed correlations between manually selected classes. We see a similar trend as before. Within box correlations show that multiple labels occur together simultaneously. Within frame correlations indicate that multiple people in a frame are likely to perform the same activity, except for a few classes such as *give object to person, carry/hold, listen to, or talk to*. Across two simultaneous frames, we see a very strong diagonal correlation, indicating that even though the actions are atomic (small and localized), they occur for more than 1 second (keyframe). Finally, we see a slightly weaker diagonal, and more cross-label correlations in the shot. For example, *sit, stand, talk to, and watch* all show correlations with almost all other classes in the dataset.

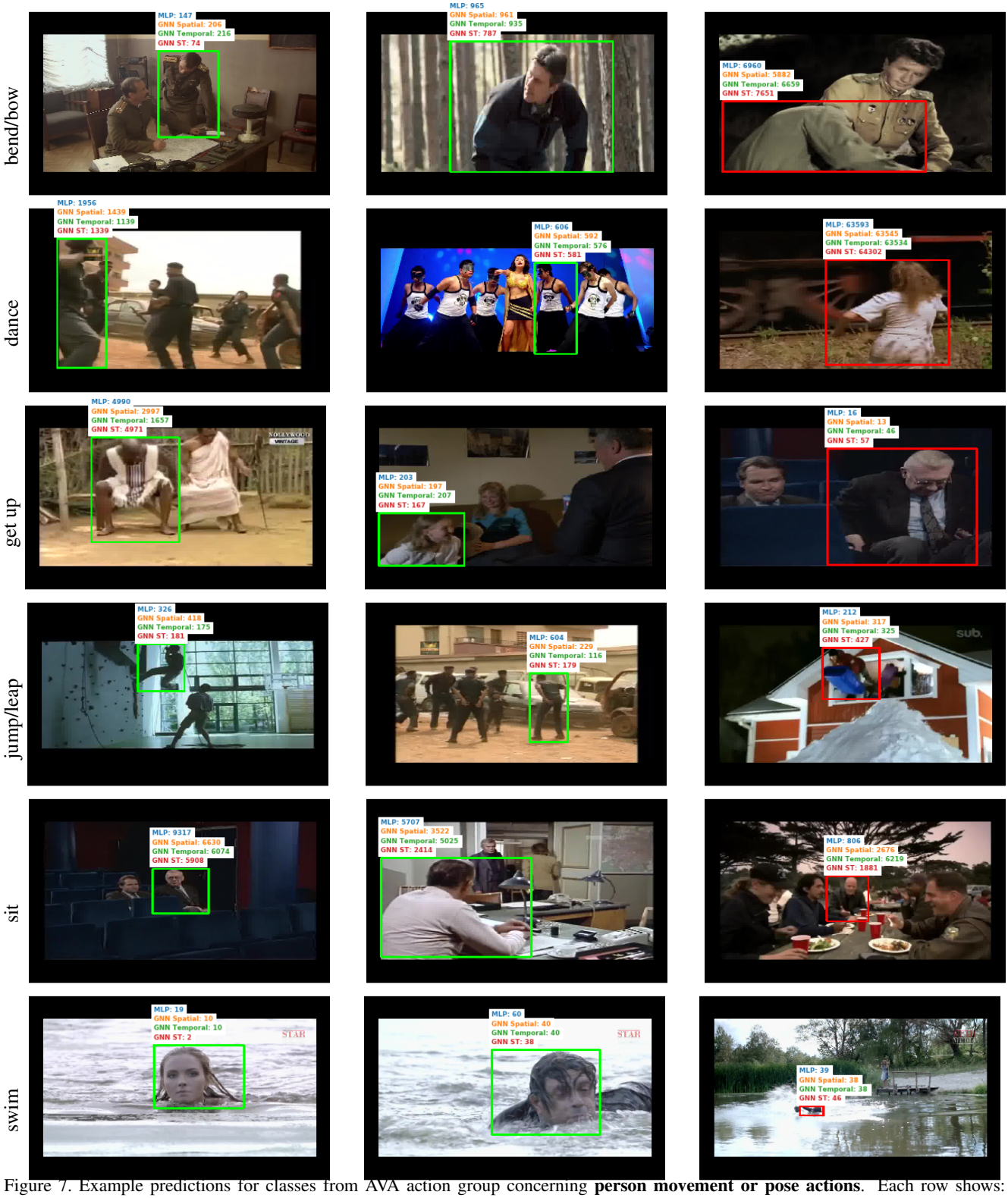

Figure 7. Example predictions for classes from AVA action group concerning **person movement or pose actions**. Each row shows: *bend/bow at the waist*; *dance*; *get up*; *jump/leap*; *sit*; and *swim* (from top to bottom). Columns 1 and 2 show improvements in rank (action probability scores) using our GNN Spatio-Temporal model, while the last column shows some failure cases.

Figure 8. Example predictions for classes from AVA action group concerning **person-object interactions**. Each row shows: *answer phone*; *cut*; *play musical instrument*; *smoke*; *pull an object*; and *read* (from top to bottom). Columns 1 and 2 show improvements in rank (action probability scores) using our GNN Spatio-Temporal model, while the last column shows some failure cases.

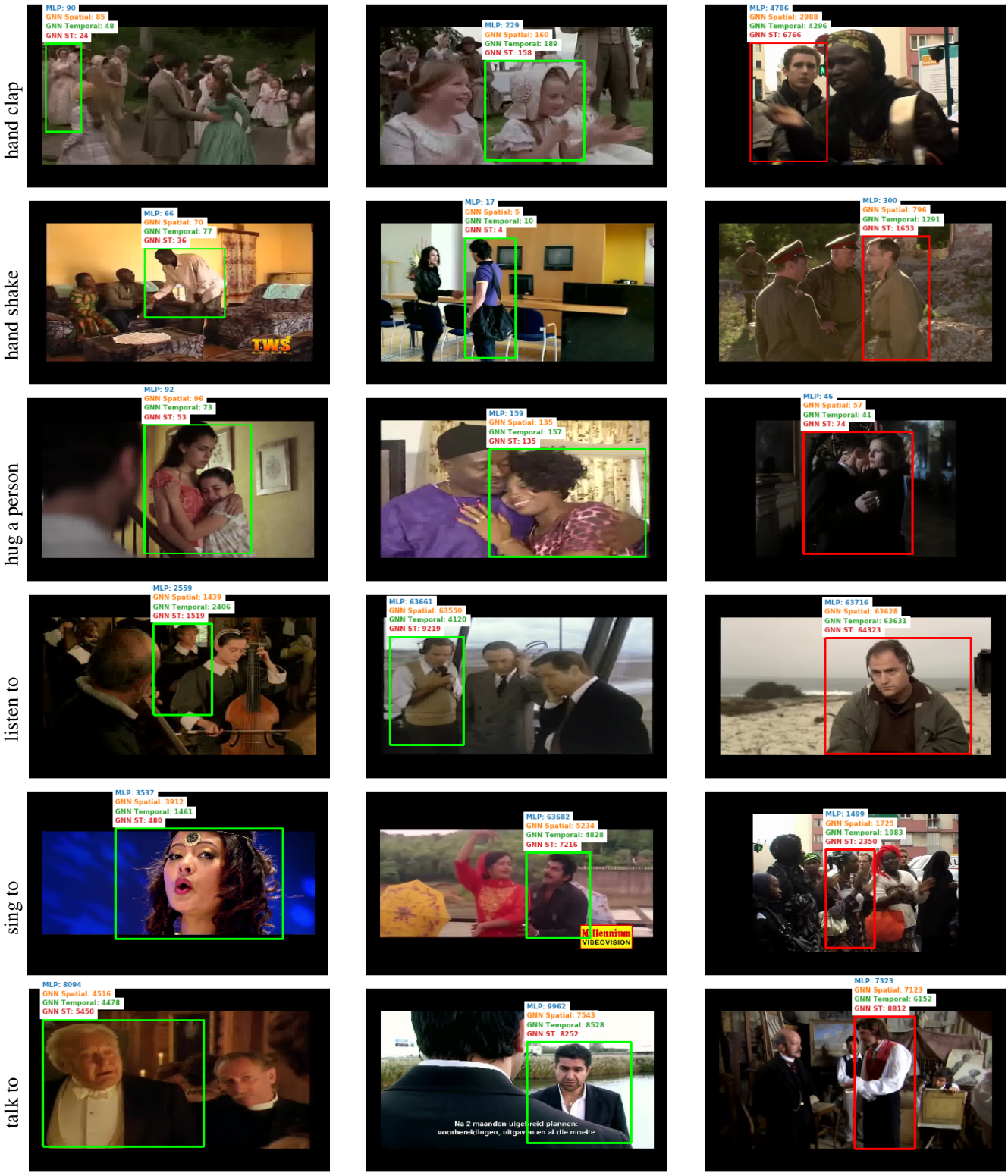

Figure 9. Example predictions for classes from AVA action group concerning **person-person interactions**. Each row shows: *hand clap*; *hand shake*; *hug a person*; *listen to*; *sing to*; and *talk to* (from top to bottom). Columns 1 and 2 show improvements in rank (action probability scores) using our GNN Spatio-Temporal model, while the last column shows some failure cases.

Figure 10. Example shot boundaries for few videos from the AVA dataset (YouTube video id indicated above the example frames). Each shot has the corresponding duration in seconds indicated below. Black separating lines are correctly detected examples, the red line is a *dissolve* transition, and is missed by our shot boundary detection approach.

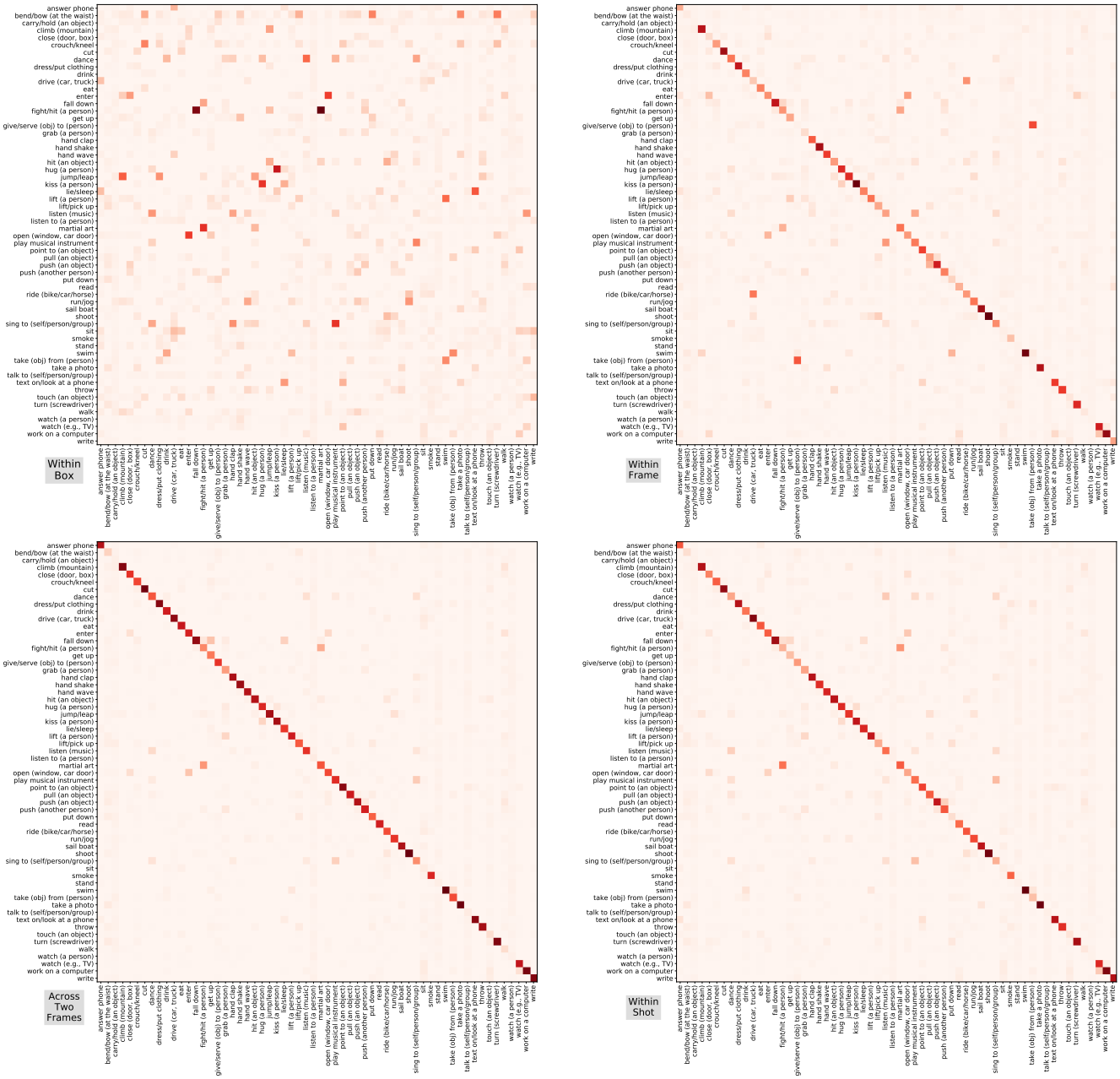

Figure 11. Normalized label correlations between atomic actions from the AVA dataset. Darker colors indicate higher correlation. From left-to-right, top-to-bottom: (a) Within a **box**: explores correlations in the multi-label setting; (b) Within a **frame**: explores what actions people perform simultaneously; (c) Across **two frames**: explores local temporal correlations between actions performed by the same character; (d) Within a **shot**: presents longer temporal correlations between actions by the same character. Best viewed on screen, please zoom in to read the labels.