

## Relational Action Forecasting

Chen Sun<sup>1</sup>, Abhinav Shrivastava<sup>2</sup>, Carl Vondrick<sup>1</sup>,  
Rahul Sukthankar<sup>1</sup>, Kevin Murphy<sup>1</sup>, and Cordelia Schmid<sup>1</sup>

<sup>1</sup>Google Research

<sup>2</sup>University of Maryland

### Abstract

*This paper focuses on multi-person action forecasting in videos. More precisely, given a history of  $H$  previous frames, the goal is to detect actors and to predict their future actions for the next  $T$  frames. Our approach jointly models temporal and spatial interactions among different actors by constructing a recurrent graph, using actor proposals obtained with Faster R-CNN as nodes. Our method learns to select a subset of discriminative relations without requiring explicit supervision, thus enabling us to tackle challenging visual data. We refer to our model as Discriminative Relational Recurrent Network (DR<sup>2</sup>N). Evaluation of action prediction on AVA demonstrates the effectiveness of our proposed method compared to simpler baselines. Furthermore, we significantly improve performance on the task of early action classification on J-HMDB, from the previous SOTA of 48% to 60%.*

### 1. Introduction

In this paper, we consider the task of forecasting what high level actions people will perform in the future, given noisy visual evidence. For example, consider Figure 1: given the current video frame, and a sequence of past frames, we would like to detect (localize) the people (man and woman), and classify their current actions (woman rides horse, man and woman are talking), as well as predict future plausible action sequences for each person (woman will get off horse, man will hold horse reigns).

More formally, our task is to compute the probability  $p(N, b_{1:N}^0, a_{1:N}^{0:T} | V^{-H:0})$ , where  $V^t$  is the frame at time  $t$  (where  $t = 0$  is the present),  $V = V^{-H:0}$  is the visual history of  $H$  previous frames,  $N$  is the number of predicted actors,  $b_n^0$  is the predicted location (bounding box) of actor  $n$  at time 0, and  $a_n^t$  is the predicted action label for actor  $n$  at time  $t$ , which we compute for  $t = 0 : T$ , where  $T$

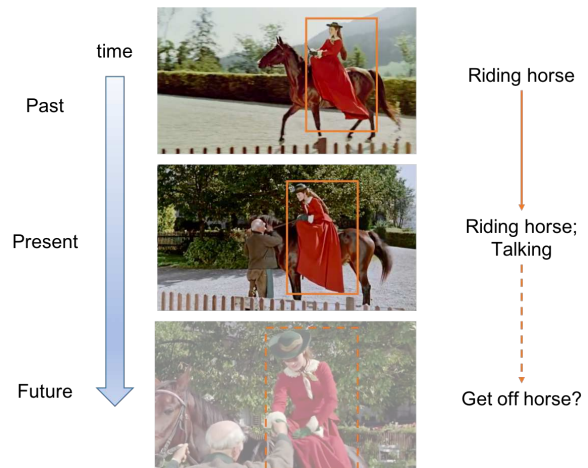


Figure 1: Action prediction from a single frame (middle) is ambiguous, but requires temporal context for the actors and their interactions. It only becomes apparent that the lady will get off the horse, if we know that she was riding towards the man, and the man is holding the horse.

is the maximum forecasting horizon. This formulation is closely related to but different than prior work. In particular, video classification focuses on computing a single global label in the offline scenario,  $p(c | V^{0:T})$ ; spatio-temporal action detection focuses on multi-agent localization and classification, but in the offline scenario,  $p(a_{1:N}^{0:T}, b_{1:N}^{0:T} | V^{0:T})$ ; action anticipation focuses on the future, but for a single class label,  $p(c^{0:T} | V^{-H:0})$ ; and trajectory forecasting focuses on multiple agents in the future, but generally focuses on locations, not actions, and assumes that the past locations (and hence the number of agents) is observed:  $p(b_{1:N}^{1:T} | V^{-H:0}, b_{1:N}^{-H:0}, N)$ . By contrast, we only observe past frames, and want to predict future high level actions of agents. This could be useful for self-driving cars, human-robot interaction, etc. See Section 2 for a more detailed discussion of related work.

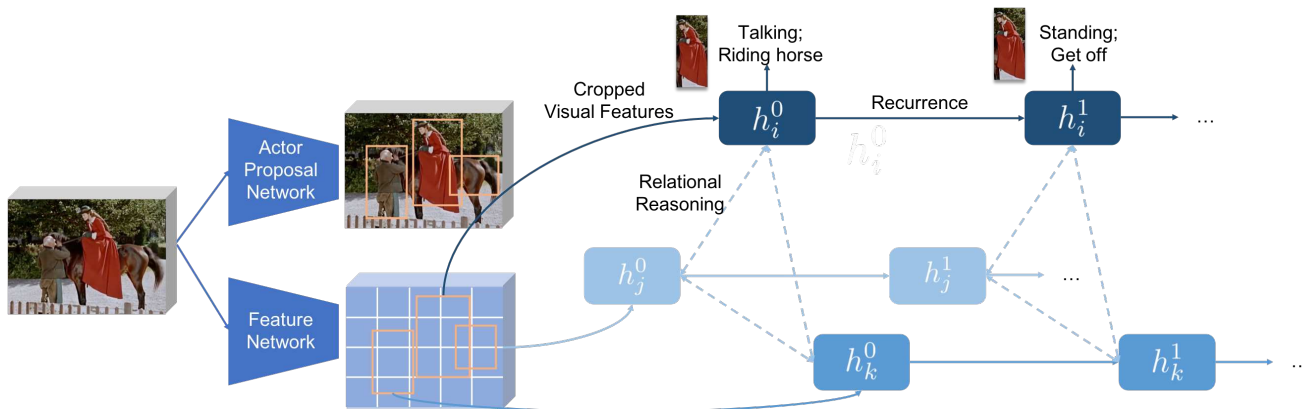


Figure 2: Overview of our approach Discriminative Relational Recurrent Network (DR<sup>2</sup>N). Given actor proposals and their spatio-temporal descriptors at a given time  $T=0$ , we model their relations by a graph neural network and its recurrence over time (here with a GRU).

Our proposed approach is to create a graph-structured recurrent neural network (GRNN), in which nodes correspond to candidate person detections (from an object detector trained on person examples), and edges represent potential interactions between the people. Each node has an action label and bounding box location associated with it. (Note that some nodes may be false positive detections arising from the detector, and should be labeled as such.) We use a modified version of graph attention networks [61] applied to a fully connected pairwise graph to capture interaction effects. Nodes are also linked over time via RNNs to capture temporal patterns in the data. We name our proposed framework Discriminative Relational Recurrent Network (DR<sup>2</sup>N). See Figure 2 for an illustration.

We train the model in a weakly supervised way, in the sense that we have a ground truth set of labeled bounding boxes for people in frame 0, and we have action labels (but not location labels) for the people for frames  $1 : T$ . However, we do not observe any edge information. Our model is able to learn which edges are useful, so as to maximize node classification performance. On the challenging AVA dataset [18], we show that our model outperforms various strong baselines at the task of predicting person action labels for up to 5 seconds into the future.

To be more directly comparable to prior work, we also consider another task, namely “early classification” of video clips, where the task is to compute  $p(c|V^{0:t})$ , where  $c$  is the class label for the video clip of length  $T$ , and  $t < T$  is some prefix of the clip (we use the first 10% to 50% of the frames). We modify our model for this task, and train it on the J-HMDB dataset, as used in prior works [54, 55]. We achieve significant improvements over the previous state of the art, increasing the classification accuracy given a 10% prefix from 48% (obtained by [54]) to 60%.

## 2. Related work

**Action recognition.** Human action recognition in videos is dominated primarily by three well-established tasks: action classification [52, 7, 38, 26, 56, 1, 28, 41, 29], temporal action localization [7, 24, 75, 53], and spatio-temporal action detection [30, 76, 48, 26, 56, 71, 40, 18]. Given a video (a set of frames), the goal of action classification is to assign action labels to the entire video, whereas temporal action localization assigns labels to only a subset of frames representing the action. Spatio-temporal action detection combines the temporal action localization with actor detection, i.e., detecting actors per-frame in a subset of frames and assigning action labels to per-actor spatio-temporal tubes [30, 76, 26].

The three standard tasks discussed above assume that the entire video is observed, therefore prediction in any frame can utilize past or future frames. In contrast, this paper introduces the task of actor-level action prediction, i.e., given a video *predict* or *anticipate* what actions *each actor* will perform in the future. This task operates in a more practical setup where only the past and present frames are observed, and predictions have to be made for unobserved future frames. Note that the proposed task inherently requires spatio-temporal actor detection in the observed frames.

**Future prediction.** Our work follows a large number of works studying future prediction [64, 44, 39, 45, 14, 3, 27, 35, 73, 66, 57, 67, 77, 63, 32, 16, 78, 2, 47, 68, 62, 15, 33, 13, 75, 36]. Broadly speaking, the research on future prediction follows two main themes: generating future frame(s) [64, 44, 39, 45, 14, 3, 27, 35, 73, 57] and predicting future labels or state(s) [67, 77, 63, 32, 16, 78, 2, 47, 62, 15, 33, 13, 75, 36, 66, 49]. For future frame generation, there is a wide variety of approaches ranging from predicting intermediate representations (e.g., optical flow [45, 15, 68],

human pose [62, 69]) and using it to generate future pixels, to directly generating future pixels by extending generative models for images to videos [64, 44, 14, 3]. Though the quality of generated frames has improved over the past few years [65, 3], it is arguably solving a much harder task than necessary (after all, humans can predict likely future actions, but cannot predict likely future pixels).

The second theme in future prediction circumvents pixel generation and directly predicts future states [67, 77, 63, 32, 16, 78, 2, 47, 62, 15, 33, 13, 75, 36, 66, 49]. These future states can vary from low-level trajectories [67, 32, 2, 47, 49, 33] for different agents to high-level semantic outputs [36, 13, 75, 37, 58]. The trajectory-based approaches rely on modeling and predicting agent behavior in the future, where an agent can be an object (such as human [32, 2, 47] or car [49]) or an image patch [67]. Most of these methods require a key assumption that the scene is static; i.e., no camera motion (e.g., VIRAT dataset [42]) and no movement of non-agent entities [32, 2, 47, 49]. Our method is similar to these trajectory-based methods, in the sense that our state prediction is also about agents. However, we do not assume anything about the scene. In particular, the AVA dataset has significant camera motion and scene cuts, in addition to agent motion and cluttered, dynamic backgrounds.

Much work on future forecasting focuses on the spatio-temporal extent of detected actions [54, 55, 20, 37]. However, there is some work on forecasting high-level semantic states, ranging from semantic segmentation masks [36] to action classes [13, 75]. However, our method has several key distinctions. First, many works require the semantic states to be part of the input to the forecasting algorithm, whereas our method detects actors and actions from past/present frames. Second, most of the state prediction methods operate on MoCap data (e.g., 3D human keypoints) [16, 25], whereas our approach works from pixels. Third, many methods assume static cameras and possibly single agents, whereas we can forecasting labels for multiple agents in unconstrained video.

**Relational reasoning.** Our proposed approach builds on the field of relational reasoning [6, 51, 43, 31, 50, 61, 21, 70, 12, 19, 5, 4, 43]. This is natural because the current and future actions of an actor rely heavily on the dynamics it shares with other actors [2]. Relational reasoning [6], in general, can capture relationships between a wide array of entities. For example, relationship between abstract entities or features [51, 70], different objects [4], humans and objects [10, 9, 17, 74], humans and context [59], humans and humans [2, 23], etc. Our work aims to capture human-human relationships to reason about future actions.

In terms of modeling these relationships, the standard tools include Interaction Network (IN) [5], Relation Network (RN) [51], Graph Neural Network (GNN) [19], Graph Attention networks [61], as well as their contemporary ex-

tensions to videos, such as Actor-centric Relation Network (ACRN) [59] and Object Relation Network (ORN) [4]. Similar to ACRN and ORN, our proposed DR<sup>2</sup>N tries to capture relation between different entity in videos for spatio-temporal reasoning. However, as opposed to modeling exhaustive relationships (RN [51] and ORN [4]), our method discovers discriminative relationships for the task of actor-level action prediction. Compared to ACRN, our method focuses on forecasting future actor labels given past visual information. In addition, we model interaction between the agents, but ignore any objects in the scene.

### 3. Approach

In this section, we describe our approach in more detail.

#### 3.1. Creating the nodes in the graph

To generate actor proposals and extract the initial actor-level features for action prediction, we build our system on top of the two-stage Faster RCNN [46] detector. The first stage is a region proposal network (RPN) that generates a large set (e.g. hundreds) of candidate actor proposals in terms of 2D bounding boxes on a single frame. We apply this RPN module on the last observed frame  $V^0$  to locate actors whose actions are to be predicted. These become nodes in the graph.

The second stage associates features with these nodes. We first extract visual features from video inputs,  $V^{-H:0}$ , using a 3D CNN (see 3.6 for details), and then crop out features inside the bounding boxes for all (candidate) actors using ROI Pooling. Let  $\mathbf{v}_i$  be the visual features for actor  $i$ .

The third stage is to connect the nodes together into a fully connected graph. However, since not all nodes are equally useful for prediction, in Section 3.3 we explain how to learn discriminative edge weights.

#### 3.2. Modeling node dynamics

We model the action dynamics of individual actors using RNNs. Given  $\mathbf{h}_i^t$  as the latent representation for actor  $i$  at time  $t$  and  $\mathbf{a}_i^t$  as the set of actor labels, we have

$$\mathbf{h}_i^t = f_{\text{RNN}}(\mathbf{h}_i^{t-1}, \mathbf{a}_i^{t-1}) \quad (1)$$

$$\mathbf{a}_i^t = f_{\text{CLS}}(\mathbf{h}_i^t) \quad (2)$$

where  $f_{\text{RNN}}(\cdot)$  is the RNN update function, and  $f_{\text{CLS}}(\cdot)$  is an action classifier that decodes the latent states  $\mathbf{h}$  into action labels (we use a simple MLP for this, see 3.6 for details). The initial state  $\mathbf{h}_i^0$  is set to  $\mathbf{v}_i$ , the visual features extracted for this bounding box.

To make the model scalable over a varying number of actors, the RNN function  $f_{\text{RNN}}(\cdot)$  is shared over all actors. Similarly, the action classifier  $f_{\text{CLS}}(\cdot)$  is shared over all actors and all time steps.

$$\begin{aligned}
p(N, b_{1:N}^0, a_{1:N}^{0:T} | V^{-H:0}) &= \delta(N, b_{1:N}^0 | f_{\text{RPN}}(V^0)) p(a_{1:N}^0, h_{1:N}^0 | b_{1:N}^0, V^{-H:0}) \prod_{t=1}^T p(a_{1:N}^t, h_{1:N}^t | a_{1:N}^{1:t-1}, h_{1:N}^{t-1}) \\
p(a_{1:N}^0, h_{1:N}^0 | b_{1:N}^0, V) &= \prod_{n=1}^N \text{Cat}(a_n^0 | f_{\text{CLS}}(h_n^0)) \delta(h_n^0 | f_{\text{ROI}}(f_{\text{S3D}}(V^{-H:0}), b_n^0)) \\
p(a_{1:N}^t, h_{1:N}^t | a_{1:N}^{t-1}, h_{1:N}^{t-1}) &= \prod_{n=1}^N \text{Cat}(a_n^t | f_{\text{CLS}}(h_n^t)) \delta(h_n^t | f_{\text{RNN}}(\tilde{h}_n^{t-1}, a_n^{t-1})) \delta(\tilde{h}_n^{t-1} | f_{\text{GNN}}(h_{1:N}^{t-1}))
\end{aligned}$$

Table 1: Formal specification of DR<sup>2</sup>N model.  $h_n^t$  is the hidden state of RNN  $n$  at time  $t$ ;  $\delta(a|b)$  denotes a deterministic probability distribution. Here  $f_{\text{RPN}}$  is a region proposal network applied to frame  $V^0$  which predicts the location of  $N$  boxes  $b_{1:N}^0$ .  $f_{\text{ROI}}$  is a region of interest feature extractor applied to S3D features derived from frames  $V^{-H:0}$  at the locations specified by the boxes.  $f_{\text{CLS}}$  is an MLP classifier with softmax output, and  $\text{Cat}(ap)$  is a categorical distribution over action labels  $a$  with parameters  $p$ .  $f_{\text{GNN}}$  is a graph neural network where the input nodes are the old RNN hidden states,  $h_{1:N}^{t-1}$ , and the output nodes are denoted  $\tilde{h}_{1:N}^{t-1}$ ; its definition is given in Table 2. Finally,  $f_{\text{RNN}}$  is a recurrent neural net update function applied to the previous predicted label and GNN output.

$$\tilde{h}_{1:N} = f_{\text{GNN}}(h_{1:N}) = \prod_{i=1}^N \delta(\tilde{h}_i | f_{\text{node}}([h_i, z_i])) \delta(z_i | \sum_j \alpha_{ij} h_j) \prod_{j=1}^N \delta(\alpha_{ij} | \mathcal{S}_j(f_{\text{attn}}(e_{i,1:N}))) \delta(e_{ij} | f_{\text{edge}}(h_i, h_j))$$

Table 2: Formal specification of the graph neural network.  $f_{\text{node}}$  is an MLP that computes node states,  $f_{\text{edge}}$  is an MLP that computes edge states,  $f_{\text{attn}}$  is a self-attention network, and  $\mathcal{S}_j(l)$  is the  $j$ 'th output of the softmax function with logits  $l$ .

### 3.3. Modeling the edges

Our current model captures action dynamics for a set of independent actors. Many actions involve interactions among actors (*e.g.* hugging and kissing). To capture such interactions, it is important to model the relations between different actors. Motivated by the recent success on relational reasoning, we combine graph neural networks (GNN) with recurrent models. We treat each actor as a node and use the RNN's latent representation  $\mathbf{h}_i^t$  as node feature.

We first consider a general graph network definition. For simplicity, we ignore the time step  $t$ , and denote  $\mathbf{h}_i$  as the feature representation for node  $i$ . Let's also denote  $\mathcal{N}_i$  as the neighbors of  $i$  in the graph. We compute the output representation  $\tilde{\mathbf{h}}_i$  from the input features of the other connected nodes:

$$\mathbf{e}_{ij} = f_{\text{edge}}(\mathbf{h}_i, \mathbf{h}_j) \quad (3)$$

$$\tilde{\mathbf{h}}_i = f_{\text{node}}(\{\mathbf{e}_{ij} : j \in \mathcal{N}_i\}) \quad (4)$$

where  $\mathbf{e}_{ij}$  is the derived features for edge  $(i, j)$ . Both  $f_{\text{edge}}(\cdot)$  and  $f_{\text{node}}(\cdot)$  can be implemented with neural networks. Note that  $f_{\text{node}}(\cdot)$  is a function mapping a set to a vector. To make it permutation invariant, it is often imple-

mented as

$$\tilde{\mathbf{h}}_i = f_{\text{node}} \left( \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij} \right). \quad (5)$$

*i.e.* the output node feature is the average over all edge features connected to the node.

The overall graph-RNN update function can thus be expressed as

$$\mathbf{h}_i^t = f_{\text{RNN}}(\tilde{\mathbf{h}}_i^{t-1}, \mathbf{a}_i^{t-1}) \quad (6)$$

*i.e.* we apply GNN on the hidden states at time  $t - 1$ , then run the standard RNN update function.

In practice, the number of actors are not known before hand, neither are the graph structures (relations) provided as supervision. Besides, the outputs from person detectors are typically over-complete and noisy (*e.g.* hundreds to thousands of proposals per frame are generated by FasterRCNN). One method to handle unknown graph is via "relation networks" [51, 59], which assumes the graph is fully connected (*i.e.*  $\mathcal{N}_i = \{\mathbf{v}_j | j \neq i\}$ ). According to Equation 5, this leads to an average over all edge features, and is sensitive to noisy nodes.

To mitigate this problem, we introduce the concept of "virtual node"  $\mathbf{z}_i$  for node  $i$ . The virtual node is connected

to all nodes in  $\mathcal{N}_i$ , and aggregates the node features with a weighted sum:

$$\mathbf{z}_i = \sum_j \alpha_{ij} \mathbf{h}_j \quad (7)$$

The distributions of soft weights for neighboring nodes are given by

$$\alpha_{ij} = \text{softmax}(f_{\text{attn}}(\mathbf{e}_{ij})) \quad (8)$$

where  $f_{\text{attn}}(\cdot)$  is an attention function that measures the importance of node  $j$  to node  $i$ .  $f_{\text{attn}}(\cdot)$  can be efficiently implemented as the self-attention mechanism [60, 61] with neural networks. Its parameters can be jointly learned with the target task using back propagation, and thus requires no additional supervision.

Once  $\mathbf{z}_i$  is computed, we assume node  $i$  is connected only to this virtual node, and updates the output feature by

$$\tilde{\mathbf{h}}_i = f_{\text{node}}([\mathbf{h}_i; \mathbf{z}_i]) \quad (9)$$

The difference from graph attention networks [61] is that they use  $\tilde{\mathbf{h}}_i = f_{\text{node}}(\mathbf{z}_i)$ . We have found this gives worse results (see Section 4), perhaps because the model is not sure if it should focus on features from itself or features from neighbors.

### 3.4. Summary of model

We call our overall model Discriminative Relational Recurrent Network or DR<sup>2</sup>N for short. See Figure 2 for a sketch, and Tables 1 and 2 for a precise specification of the model.

### 3.5. Training

The overall framework is jointly optimized end-to-end, where the loss function is

$$\mathcal{L}^{\text{total}} = \alpha \mathcal{L}^{\text{loc}} + \sum_{t=0}^T \beta_t \mathcal{L}_t^{\text{cls}} \quad (10)$$

Here  $\mathcal{L}^{\text{loc}}$  is the box localization loss given by the region proposal network and the bounding box refinement network computed for the last observed frame.  $\mathcal{L}_t^{\text{cls}}$  is the action classification loss at time  $t$ .  $\alpha$  and  $\beta_t$  are scalars which balance the two sets of losses. In practice, one may want to down-weight  $\beta_t$  for larger  $t$  as the prediction task becomes more challenging.

Note that we do not use teacher forcing during training, to encourage the model to predict multiple steps into the future. That is, when computing the predicted labels  $a_i^t$ , we condition on previous *predicted* labels  $a_{1:t}^{0:t-1}$  rather than the ground truth predicted labels. (We use the soft logit scores for the predictions, to avoid the need to sample from the model during training.)

## 3.6. Implementation details

Our implementation of the Faster-RCNN [46, 22] proposal extractor largely follows the design choices of [59]. The region proposal network (RPN) uses a 2D ResNet-50 network and takes a single image as input. It is jointly trained with the whole framework, using the human annotations from target dataset. We apply RPN on the final observed frame of each example to generate actor proposals. To handle temporal context, the feature network uses an S3D-G [72] backbone, which is a type of “inflated” 3D convolutional neural network [8], and takes sequences of frames as inputs. We apply the feature network to frames at  $-H : 0$ , where 0 is the frame number of the last observed frame, and  $H$  is the duration of temporal context. Once the features are extracted, we apply a temporal convolutional layer after the `Mixed_4f` layer of S3D-G to aggregate the 3D feature maps into 2D and then apply the standard 2D ROI Pooling to crop out the features inside actor proposals. Each cropped feature map is passed to the remaining layers of S3D-G. The final outputs are average-pooled into 1024-dim feature vectors.

The weights of ResNet-50 used by proposal network are initialized with an ImageNet pre-trained model. We keep top 300 proposals per image. The weights of S3D-G used by feature network are pre-trained from Kinetics-400 [29]. The weights of newly added layers are randomly initialized from truncated normal distributions. Unlike otherwise mentioned, the inputs to the feature network at 10 RGB frames resized to 400 by 400. We use gated recurrent units (GRU) [11] as the particular RNN architecture to model action dynamics. We set the number of hidden units to 1024, which is the same dimension as the visual input features. For the discriminative relation network, we implement  $f_{\text{edge}}(\cdot)$  and  $f_{\text{attn}}(\cdot)$  as single fully-connected layers.

During both training and inference, the input actions  $\mathbf{a}_i^t$  to the GRU are generated by the model rather than provided by the ground truth. We set the localization weight  $\alpha = 1$ . The classification weight  $\beta_0$  is set to 1, we linearly anneal  $\beta_t$  such that  $\beta_t = 0.5$ . To compute classification loss, we use softmax cross entropy for J-HMDB and the sum of sigmoid cross entropies for AVA (since the action labels of AVA are not mutually exclusive). We optimize the model with synchronous SGD and batch size of 4 per GPU, and disable batch norm updates during training. We use 10 GPUs in total. Two techniques are used to stabilize and speed up training: first, we warm-start the learning rate from 0.008 to 0.08 for  $T_w$  steps and use cosine learning rate decay [34] starting from 0.08 for another  $T_c$  steps; second, we apply a gradient multiplier of 0.01 to gradients computed from DR<sup>2</sup>N to the feature map. For AVA, we set  $T_w$  to 5K and  $T_c$  to 300K. For J-HMDB, we set  $T_w$  to 1K and  $T_c$  to 40K.

Method	Dynamics Model	Relation Model	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Single-head	-	-	19.1	7.8	5.3	4.2	2.6	1.8
Multi-head	-	-	16.0	9.4	6.8	5.4	4.3	3.6
GRU	GRU	-	18.7	13.1	10.3	8.0	6.7	5.7
Graph-GRU	GRU	RN [51]	17.3	12.3	9.9	7.7	6.5	5.3
Graph-GRU	GRU	GAT [61]	16.4	12.3	9.3	7.3	6.2	5.2
Graph-GRU	GRU	DR <sup>2</sup> N (Us)	<b>20.4</b>	<b>14.4</b>	<b>11.2</b>	<b>9.3</b>	<b>7.5</b>	<b>6.8</b>

Table 3: Ablation study on the AVA dataset. We report mean AP@.5 for six different time steps.  $t = 0$  corresponds to the last observed frame, i.e., standard action detection. Each step is one second long.

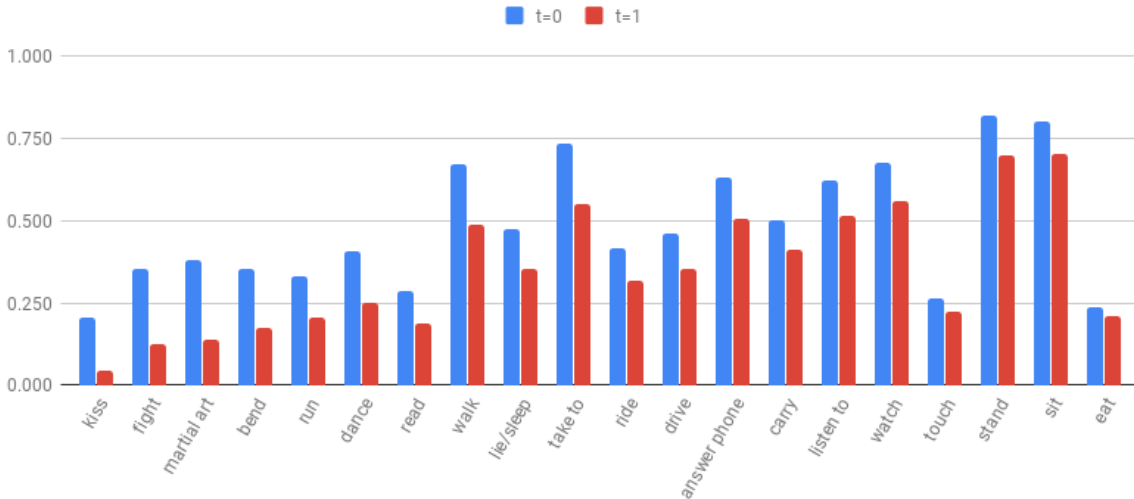


Figure 3: Change in AP performance from  $T = 0$  to  $t = 1$ . The actions with the biggest change are the hardest to predict in the future.

## 4. Experiments

In this section, we conduct experiments to study the impact of different design choices for relational reasoning and temporal dynamics modeling for the action prediction task.

### 4.1. Experimental setup

In the following we present the two datasets used in our experiments, Atomic Visual Actions (AVA) [18] and J-HMDB [26] as well as the metrics used for evaluating action prediction.

AVA [18] is a recently released large-scale action detection dataset with 60 action classes. AVA is sparsely labeled at 1 FPS and each frame may contain multiple actors with multiple action labels. We use the most recent AVA version 2.1, which contains 210K training examples and 57K validation examples. To get the ground-truth for future action labels, we use the bounding box identities (tracks) semi-automatic annotated for this dataset [18]. For actor-

level action prediction on AVA, we measure the IoU of all the actor detections with the ground-truth boxes. If the IoU is above 0.5, and the action label is correct, we consider it a true positive. For prediction, the ground-truth labels come from future boxes that are linked to the last observed ground truth boxes. We compute the average precision, and report per frame-level mean AP for different time steps.

**J-HMDB** is an action detection dataset with 928 clips and 21 categories. Clips have an average length of 1.4 seconds. Every frame in J-HMDB contains one annotated actor with a single action label. There are three train/val splits for J-HMDB. We report results by averaging over the three splits, which is the standard practice on this dataset. Since there is only one action performed in each example, we follow the setup of Soomro *et al.* [54, 55] and treat it as an early action prediction problem. We report accuracy@K, which is the action classification accuracy by watching the first K% of the videos.

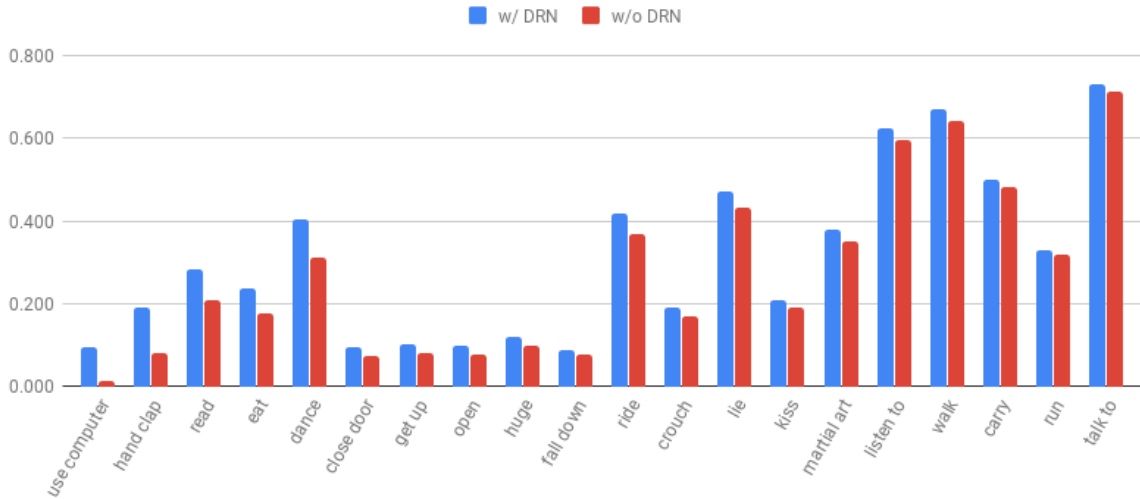


Figure 4: Change in AP performance from adding graph connections at  $t = 0$ . The actions with the biggest change benefit the most from contextual modeling.

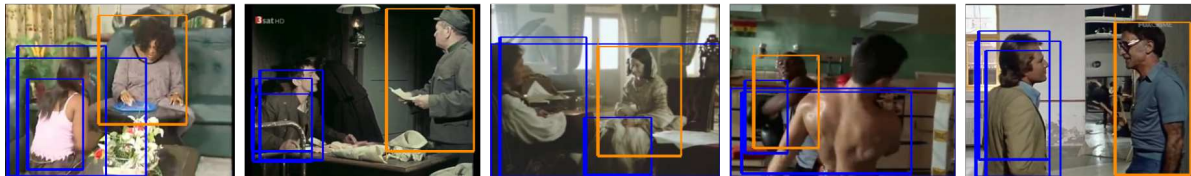


Figure 5: Visualizations of top 3 relations (blue boxes) selected for an actor proposal (orange box) by DR<sup>2</sup>N on AVA. We see that the attended regions provide useful contextual information.

## 4.2. Action prediction on AVA

This section presents and discusses quantitative and qualitative results for action prediction on the AVA dataset. We consider the following approaches:

- **Single-head:** for each future step  $t$ , train a separate model that directly classifies the future actions from visual features  $\mathbf{v}_i$  derived from  $V^0$ .
- **Multi-head:** similar to the single-head model, but jointly trains the classifiers for all  $t$  in the same model. The visual features  $\mathbf{v}_i$  are shared with all classification heads.
- **GRU:** future actions are predicted from hidden states of GRU, where the states are initialized from visual features of the actor proposals. Model is trained jointly for all  $t$ . However, there are no edges in the graph, so all nodes evolve independently.
- **Graph-GRU:** Same as GRU, but with a fully connected graph. We consider 3 versions: the Relation Network (RN) [51], which assigns equal weights to all pairwise relations; Graph Attention Network [61],

which uses a weighted sum of features from itself and all its neighbors; and our proposed method, which uses Equation 5.

The results are shown in Table 3. The first three rows compare the impact of different approaches for dynamics modeling. Our first observation is that, as  $t$  grows larger, the mean AP declines accordingly. This is expected since the further away in time the prediction is, the harder is the task. Our second observation is that the single-head baseline performs worse on all  $t$  except for  $t = 0$ , where the frames are observed. The lower performance of  $t = 0$  for multi-head can be explained by the fact that the joint model has less model capacity compared with 6 independent single-head models. However, we can see that by sharing the visual features in a multi-task setting, the multi-head baseline outperforms its single-head counterpart for future prediction at  $t > 0$ . Our third observation is that using GRU to model action dynamics offers better future prediction performance without sacrificing detection performance at  $t = 0$ , since it can capture patterns in the sequence of action labels.

The last three rows of Table 3 compares the impact of different relational models. We can see DRN outperforms



Figure 6: Example predictions on the AVA validation set at  $t = 1$ . We show last observed frames at  $t = 0$  and render the detected actor boxes on the frames. To the right of each example, we also show the unobserved future frames half and one second ahead. We show top one detections if above threshold of 0.1, and remove the most frequent categories, such as sitting and standing. The top row shows examples where the model can predict what will happen in the future based on the current scene context. The second row shows examples where the model can predict what will happen in the future based on the other actors in the scene. The third row highlights the challenges in action forecasting (e.g. multiple possible futures).

the other two consistently. For RN, one possible explanation for the performance gap is that it assigns equal weights to all edges, which is prone to noise in our case, since many nodes correspond to background detections. For GAT, we notice that the performance at  $t = 0$  is much lower, indicating that it has difficulty distinguishing node features from neighbor features.

Figure 3 compares the classes with biggest performance drops from detection ( $t = 0$ ) to prediction ( $t = 1$ ). We see that it is challenging for the model to capture actions with short durations, such as kissing or fighting. Actions with longer durations, such as talking, are typically easier to predict. Figure 4 compares the effectiveness of DR<sup>2</sup>N over the GRU baseline, without any edges in the graph. We can see that the categories with the most gains are those with explicit interactions (e.g. hand clap, dance, martial art), or where other actors provide useful context (e.g. eat and ride). In Figure 5, we show the top 3 boxes (blue) with the highest attention weights to the actor being classified (orange). We can see that they typically correspond to other actors. Finally, we visualize example predictions in Figure 6.

### 4.3. Early action prediction on J-HMDB

Finally, we demonstrate the effectiveness of DR<sup>2</sup>N on the early clip classification. During training, we feed 10 RGB frames to the feature network, and predict one step into the future. During inference, we feed the first  $K\%$

Model	10%	20%	30%	40%	50%
Soomro <i>et al.</i> [55]	≈ 5	≈ 12	≈ 21	≈ 25	≈ 30
Singh <i>et al.</i> [54]	≈ 48	≈ 59	≈ 62	≈ 66	≈ 66
GRU	52.5	56.2	61.1	65.2	65.9
GAT [61]	58.1	61.8	64.4	68.7	68.8
DR <sup>2</sup> N	<b>60.6</b>	<b>65.8</b>	<b>68.1</b>	<b>71.4</b>	<b>71.8</b>

Table 4: Early action prediction performance on J-HMDB.

frames to the feature network, and take the most confident prediction as the label of the clip. Table 4 shows the results, we can see that our approach significantly outperforms previous state-of-the-art methods. To study the impact of relational models, we also compare with the GRU only and the GAT baselines, and find DR<sup>2</sup>N outperforms both. By inspecting the edge attentions, we observe that some of the RPN proposals cover objects in the scene, which are utilized by DR<sup>2</sup>N to model human-object relations.

## 5. Conclusion

We address the multi-person action forecasting task in videos. We propose a model that jointly models temporal and spatial interactions among different actors with Discriminative Relational Recurrent Network. Quantitative and qualitative evaluations on AVA and J-HMDB datasets demonstrate the effectiveness of our proposed method.



## References

- [1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. [2](#)
- [2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. [2, 3](#)
- [3] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. [2, 3](#)
- [4] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *ECCV*, 2018. [3](#)
- [5] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NIPS*, 2016. [3](#)
- [6] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. J. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. R. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. [3](#)
- [7] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. [2](#)
- [8] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017. [5](#)
- [9] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng. Learning to detect human-object interactions. In *WACV*, 2018. [3](#)
- [10] Y.-W. Chao, Z. Wang, Y. He, J. Wang, and J. Deng. HICO: A benchmark for recognizing human-object interactions in images. In *ICCV*, 2015. [3](#)
- [11] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014. [5](#)
- [12] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017. [3](#)
- [13] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*, 2018. [2, 3](#)
- [14] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016. [2, 3](#)
- [15] K. Fragkiadaki, J. Huang, A. Alemi, S. Vijayanarasimhan, S. Ricco, and R. Sukthankar. Motion prediction under multimodality with conditional stochastic networks. *arXiv preprint arXiv:1705.02082*, 2017. [2, 3](#)
- [16] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *ICCV*, 2015. [2, 3](#)
- [17] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object interactions. In *CVPR*, 2018. [3](#)
- [18] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. [2, 6](#)
- [19] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 2017. [3](#)
- [20] M. Hoai and F. De la Torre. Max-margin early event detectors. *IJCV*, 2014. [3](#)
- [21] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. *arXiv preprint arXiv:1711.11575*, 2017. [3](#)
- [22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017. [5](#)
- [23] M. S. Ibrahim and G. Mori. Hierarchical relational networks for group activity recognition and retrieval. In *ECCV*, 2018. [3](#)
- [24] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The THUMOS challenge on action recognition for videos “in the wild”. *CVIU*, 2017. [2](#)
- [25] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *CVPR*, 2016. [3](#)
- [26] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. In *ICCV*, 2013. [2, 6](#)
- [27] N. Kalchbrenner, A. v. d. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016. [2](#)
- [28] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. [2](#)
- [29] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. [2, 5](#)
- [30] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005. [2](#)
- [31] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. In *ICML*, 2018. [3](#)
- [32] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. [2, 3](#)
- [33] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. K. Chandraker. DESIRE: distant future prediction in dynamic scenes with interacting agents. In *CVPR*, 2017. [2, 3](#)
- [34] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with restarts. *arXiv preprint arXiv:1608.03983*, 2016. [5](#)
- [35] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. [2](#)
- [36] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017. [2, 3](#)

- [37] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, 2016. 3
- [38] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. 2
- [39] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 2
- [40] P. Mettes, J. van Gemert, and C. Snoek. Spot On: Action localization from pointly-supervised proposals. In *ECCV*, 2016. 2
- [41] M. Monfort, B. Zhou, S. A. Bargal, A. Andonian, T. Yan, K. Ramakrishnan, L. Brown, Q. Fan, D. Gutfruehd, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *arXiv preprint arXiv:1801.03150*, 2018. 2
- [42] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 3
- [43] R. B. Palm, U. Paquet, and O. Winther. Recurrent relational networks for complex relational reasoning. *arXiv preprint arXiv:1711.08028*, 2017. 3
- [44] N. Petrovic, A. Ivanovic, and N. Jojic. Recursive estimation of generative models of video. In *CVPR*, 2006. 2, 3
- [45] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 2
- [46] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3, 5
- [47] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016. 2, 3
- [48] M. Rodriguez, J. Ahmed, and M. Shah. Action MACH: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 2
- [49] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. Car-net: Clairvoyant attentive recurrent network. In *ECCV*, 2018. 2, 3
- [50] A. Santoro, R. Faulkner, D. Raposo, J. W. Rae, M. Chrzanowski, T. Weber, D. Wierstra, O. Vinyals, R. Pascanu, and T. P. Lillicrap. Relational recurrent neural networks. *arXiv preprint arXiv:1806.01822*, 2018. 3
- [51] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. P. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017. 3, 4, 6, 7
- [52] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, 2004. 2
- [53] G. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2
- [54] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, 2017. 2, 3, 6, 8
- [55] K. Soomro, H. Idrees, and M. Shah. Online localization and prediction of actions and interactions. *IEEE PAMI*, 2018. 2, 3, 6, 8
- [56] K. Soomro, A. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. Technical Report CRCV-TR-12-01, 2012. 2
- [57] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 2
- [58] S. Su, J. Pyo Hong, J. Shi, and H. Soo Park. Predicting behaviors of basketball players from first person videos. In *CVPR*, 2017. 3
- [59] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *ECCV*, 2018. 3, 4, 5
- [60] 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. 5
- [61] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *ICLR*, 2018. 2, 3, 5, 6, 7, 8
- [62] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*, 2017. 2, 3
- [63] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 2, 3
- [64] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 2, 3
- [65] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017. 3
- [66] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 2, 3
- [67] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014. 2, 3
- [68] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. In *ICCV*, 2015. 2
- [69] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017. 3
- [70] X. Wang, R. B. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018. 3
- [71] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015. 2
- [72] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*, 2018. 5
- [73] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 2
- [74] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 3
- [75] 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*, 2017. 2, 3

- [76] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009. [2](#)
- [77] J. Yuen and A. Torralba. A data-driven approach for event prediction. In *ECCV*, 2010. [2](#), [3](#)
- [78] Y. Zhou and T. L. Berg. Temporal perception and prediction in ego-centric video. In *ICCV*, 2015. [2](#), [3](#)