Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization

Junting Pan[1⋆], Siyu Chen[2⋆], Zheng Shou[4], Jing Shao[3], Hongsheng Li[1]

[1]CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
[2]Peking University [3]SenseTime [4]Facebook AI

Abstract. Localizing persons and recognizing their actions from videos is a challenging task towards high-level video understanding. Recent advances have been achieved by modeling either "actor-actor" or "actor-context" relations. However, such direct first-order relations are not sufficient for localizing actions in complicated scenes. Some actors might be indirectly related via objects or background context in the scene. Such indirect relations are crucial for determining the action labels but are mostly ignored by existing work. In this paper, we propose to explicitly model the **Actor-Context-Actor Relation**, which can capture indirect high-order supportive information for effectively reasoning actors' actions in complex scenes. To this end, we design an Actor-Context-Actor Relation Network (ACAR-Net) which builds upon a novel *High-order Relation Reasoning Operator* to model indirect relations for spatio-temporal action localization. Moreover, to allow utilizing more temporal contexts, we extend our framework with an *Actor-Context Feature Bank* for reasoning long-range high-order relations. Extensive experiments on AVA dataset validate the effectiveness of our ACAR-Net. Ablation studies show advantages of modeling high-order relations over existing first-order relation reasoning methods. The proposed ACAR-Net is also the core module of our **1st place solution in AVA-Kinetics Crossover Challenge 2020**. Training code and models will be available at https://github.com/Siyu-C/ACAR-Net.

1 Introduction

Spatio-temporal action localization, requiring localizing persons while recognizing their actions, is an important task that has drawn increasing attention in recent years [5,6,10,25,34]. Unlike object detection which can be accomplished solely by observing visual appearances, activity recognition usually demands for reasoning about the actors interactions with the surrounding context, including other people and objects. Take Fig. 1 as an example: to recognize the action "ride" of the person in the red bounding box, we need to observe that he is inside a car, and there is a driver next to him. Therefore, recent progresses in spatio-temporal action detection have been driven by the success of relation modeling. Such attempts on relation modeling can be categorized into two types:

[⋆] Equal contribution

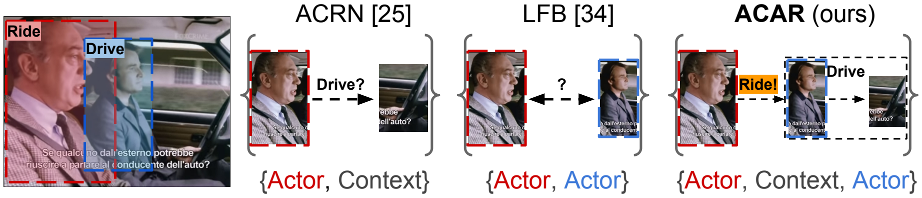


Fig. 1. Reasonsing actor-context or actor-actor relations may not be sufficient for correctly predicting the action labels of all individuals. Our method does not only reason relations between actors, but also model connections between different actor-context relations. As an example, the relation between the blue actor and the steering wheel (drive) serves as a crucial clue for recognizing the action being performed by the red actor (ride).

(1) **Modeling relation between actors** [6, 34] focuses on interactions between persons, *e.g.* the relation between the actor in the red bounding box (red actor) and the other in the blue bounding box (blue actor) in Fig. 1. These methods take feature vectors extracted from cropped actor bounding boxes as inputs. They utilize attention mechanisms to directly infer relations between actor features. However, this type of methods only use information within cropped boxes, and discard important contextual information, such as spatio-temporal locations as well as contextual objects' dynamics and appearances that are helpful for more accurate reasoning.

(2) **Modeling relation between actor and context** [25], on the other hand, leverages spatio-temporal context for recognizing the behavior of an actor. As shown in Fig. 1, it performs relation reasoning by identifying spatial regions (*e.g.* steering wheel) that have highest correlation with the blue actor to recognize his action. This type of methods do not explicitly model the relationship among actors. The advantage of these methods is that they preserve structural information which could be important for understanding the entire scene.

However, these two attempts will struggle on recognizing the action "ride" being performed by the red actor, since neither of these relations (actor-actor or actor-context) can provide sufficient clues. More concretely, in Fig. 1, it is difficult to infer the action of the red actor solely given its relation with the blue actor or with the scene context (steering wheel). To overcome this problem, we propose to capture the implicit *high-order* relation between the two actors based on their respective *first-order* relations with the context. In this way, we will be able to identify the action (ride) of red actor by reasoning over the interaction between the blue actor and the context (drive).

Given the need for high-order relation reasoning to understand videos, we propose an Actor-Context-Actor Relation Network (ACAR-Net) that deduces indirect relations between multiple actors and the context while being trained on the action localization task. The ACAR-Net takes both actor and context features as inputs. It first encodes the *first-order actor-context* relations, and then applies a **High-Order Relation Reasoning Operator** in charge of modeling links established on those first-order relations. Finally, to model long-term high-order relations, we build an **Actor-Context Feature Bank**, which contains actor-context relations at different time stamps across the whole video.

ACAR-Net learns to reason high-order relations among actors and the context, while preserving the spatial structure of the scene. Partly similar to our approach, there exist works that explicitly model interactions between actors and objects [33, 42]. However, in these approaches, when deducing the action of one person, the interactions of other persons with contextual objects are ignored. In other words, they do not explicitly model the higher-order relations built on direct actor-context relations. In contrast, our method emphasizes modeling those indirect relations, and does not need the extra step of object detection.

We conduct extensive experiments on the challenging Atomic Visual Actions (AVA) dataset [10] for spatio-temporal action localization. This dataset contains a large number of complex realistic scenes, and most of its action classes are human-object or human-human interactions. We show that our proposed ACAR-Net provides a clear advantage over previous methods on this benchmark.

Our contributions are summarized as the following:

- We propose to model high-order actor-context-actor relations for spatio-temporal action localization. Such relations are ignored by previous methods but crucial for achieving accurate action localization.
- We propose a novel Actor-Context-Actor Relation Network for improving spatio-temporal action localization by explicitly reasoning about high-order relations between actors and the context.
- We achieve state-of-the-art performances with significant margins on the AVA dataset.

2 Related Work

Action Recognition. Research works on action recognition generally fall into three categories: action classification, temporal localization and spatio-temporal localization. Early works mainly focus on classifying a short video clip into an action class. 3D-CNN [1,27], two-stream network [23,30] and 2D-CNN with RNN [4, 41] are the three dominant network architectures adopted for this task. While progresses are made for short trimmed video classification, the main research stream moves forward to understand long untrimmed videos, which requires not only to recognize the category of each action instance but also to locate its start and end times. A handful of works [22, 38] consider this problem as a detection problem in 1D temporal dimension by extending from object detection frameworks.

Spatio-Temporal Action Localization. Recently, the problem of spatio-temporal action localization was proposed, and datasets such as AVA [10], where atomic actions of all actors in the video are continuously annotated, are introduced. It brings the action detection problem into a finer level, since the action instance needs to be localized in both space and time. Typical approaches used by early works adopted R-CNN detectors for object detection on 3D-CNN features [10]. Several more recent works have exploited graph-structured networks to leverage contextual information [6,25,42]. In particular, some approaches utilize the self-attention mechanism to learn relationships among actors. Among them, Wu *et al.* [34] proposed to use long-term feature banks (LFB) to provide temporal supportive information up to 60s; ACRN [25] models relations between human actors and scene elements through a relation network; Chen *et al.* [26] integrated a modified graph attention network with an RNN to anticipate future actions, which is not directly related to the task of spatio-temporal action localization,

Relational Reasoning. We propose a relational reasoning module to model and learn the high-order relations between actors and the context. Our network is able to automatically select links for aggregating informative context. Relational reasoning has been adopted for a wide range of tasks in natural language processing and computer vision. The Transformer network [28] has become a dominant architecture for modeling sequential data with the introduced scaled dot-product attention mechanism. Similarly, Graph Attention Networks [29] and Non-local Networks [32] also leverage attention mechanisms to capture dependencies between different entities. There have also been a lot of works on modeling relations for recognizing human-human and human-object interactions [8,19,31]. Different from our method, their approaches only focus on modeling relations for static images, and require strong supervision such as annotations of objects or relations.

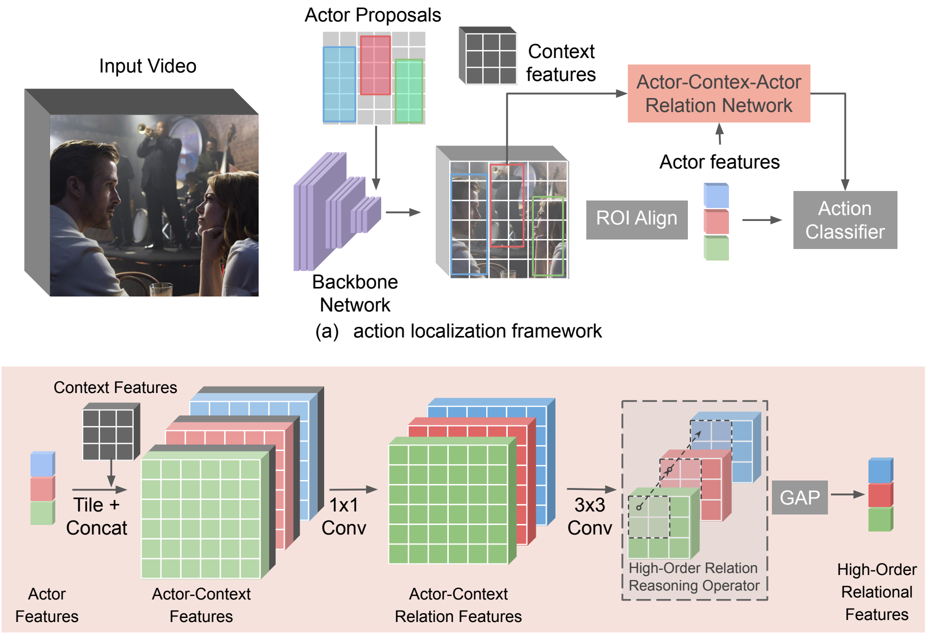
3 Method

In this section, we give a detailed description of our proposed Actor-Context-Actor Relation Network (ACAR-Net). Existing works on spatio-temporal action localization either model inter-actor relations or actor-context relations for tackling this problem, which are, however, insufficient for correctly classifying all actions in a video clip. Our method ACAR-Net gives an efficient yet effective algorithm to model and utilize the useful higher-order relations built upon the basic actor-actor and actor-context relations for assisting action localization.

3.1 Overall Framework

We first introduce our overall framework for action localization, where our proposed ACAR-Net is its key module for high-order relation modeling to achieve accurate action localization. The framework is designed to detect all persons in an input video clip and predict their action labels.

As shown in Fig. 2, following state-of-the-art methods [5,34,35], we combine an *off-the-shelf person detector* (*e.g.* Faster R-CNN [20]) with a *video backbone network* (*e.g.* I3D [2]). In details, the detector operates on the center frame (*i.e.*



(a) action localization framework

(b) Actor-Context-Actor Relation Network

Fig. 2. Overview of the action localization framework and our proposed **ACAR-Net**. Note that (b) corresponds to details of the red module (ACAR-Net) in (a).

key frame) of the clip and obtains N detected actors. Such detected boxes are duplicated to other frames of the clip. In the mean time, the backbone network extracts a spatio-temporal feature volume from the input video clip. We perform average pooling along the temporal dimension considering computational efficiency, which results in a feature map $V \in \mathbb{R}^{C\times H\times W}$, where C, H, W correspond to channel, height and width respectively. We apply RoIAlign [11] (7×7 spatial output) followed by spatial max pooling to the feature map V and the N actor boxes, producing a series of N actor features, $A_1, A_2, \cdots, A_N \in \mathbb{R}^C$. Each actor feature describes the spatio-temporal appearance and motion of one Region of Interest (RoI).

The unique design of Actor-Context-Actor Relation Network, taken as a module marked in red in Fig. 2 (a), is embedded into the whole framework. This module takes the aforementioned video feature map V and RoI features $\{A_i\}_{i=1}^N$ as inputs, and outputs the final action predictions after relation reasoning. Our design can be summarized into two parts. (1) We first encode the first-order actor-context relations between each actor and each spatial location of the spatio-temporal context. Based on the actor-context relations, we further add a **High-order Relation Reasoning Operator** (HR^{2}O) for modeling the connections established on first-order relations, which are indirect relations mostly ignored by previous methods. (2) Our reasoning module can be extended with an

Actor-Context Feature Bank (ACFB). The bank contains actor-context relations at different time stamps, so it can provide more complete spatio-temporal context than the existing long-term feature bank [34] which only consists of features of actors. We will elaborate these two parts in the following sections.

In general, our high-order relation reasoning block is weakly-supervised, which only requires action labels as supervision. Experimental results in Section 4 demonstrate the effectiveness of our proposed reasoning module.

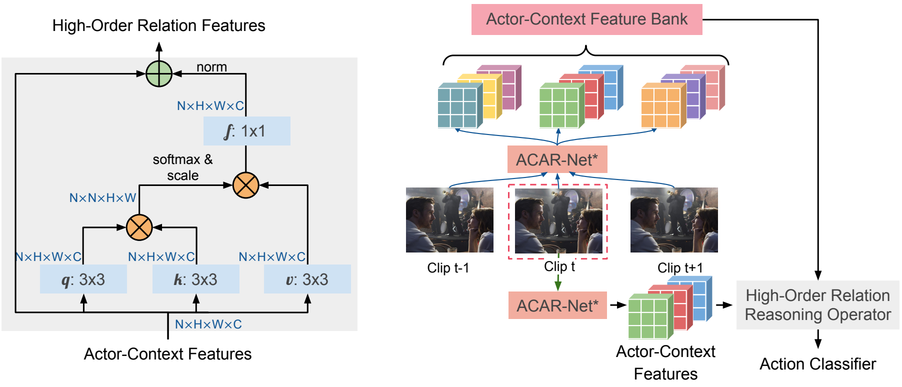
3.2 High-Order Relation Reasoning Operator

We begin with a brief review of the Actor-Centric Relation Network (ACRN) [25]. ACRN learns first-order actor-context relations by combining RoI features $A_1, \cdots, A_N$ with the context feature V. More specifically, it concatenates each actor feature $A_i \in \mathbb{R}^C$ to all $H \times W$ spatial locations of the context feature $V \in \mathbb{R}^{C\times H\times W}$ to form a concatenated feature map $F_i' \in \mathbb{R}^{2C\times H\times W}$. Actor-context relation features for each actor can then be encoded by applying convolutions to this concatenated feature map.

ACRN only provides for each actor a spatial grid ($H \times W$) of first-order relation features with the context. However, as we introduced before, missing higher-order relation reasoning makes the framework incapable of predicting some complex action labels. Let $F^i_{x,y}$ record the first-order feature between the actor A_i and the scene context V at the spatial location (x, y). We introduce a new concept - *High-order Relation Reasoning*, in order to model the relations between first-order actor-context relations, which are high-order relations encoding more informative scene semantics. However, since there are a large number of actor-context relation features, $F^i_{x,y}$, $i \in \{1, \cdots, N\}, x \in [1, H], y \in [1, W]$, the number of their possible pairwise combinations are generally overwhelming. We therefore propose to focus on learning the high-order relations between different actor-context relations at the same spatial location (x, y), i.e. $F^i_{x,y}$ and $F^j_{x,y}$. In this way, the proposed relational reasoning operator limits the relation learning to second-order actor-context-actor relations, *i.e.* two actors i and j can be associated via the same spatial context as $i \leftrightarrow (x, y) \leftrightarrow j$ to help the prediction of their action labels.

Instantiations. We investigate possible instantiations for our High-order Relation Reasoning Operator, denoted by HR^{2}O. The operator takes as input a set of first-order actor-context relation feature maps F^i, possibly as well as the actor features A_i and the video feature map V. The output $\{H^i\}^N_{i=1} = \text{HR}^2\text{O}(\{F^i\}^N_{i=1}, \{A_i\}^N_{i=1}, V)$ encodes second-order actor-context-actor relations for every actor. The high-order relation map H^i will be spatially average-pooled, and then channel-wise concatenated to the basic actor RoI feature vector A_i for final classification. All relation vectors are of dimension $d = 512$ in our implementation.

Location-wise Attention. Our default HR^{2}O is a location-wise attention operator, which is natural for modeling the connections between multiple first-order



(a) High-Order Relation Reasoning Operator

(b) Actor-Context Feature Bank

Fig. 3. Design details of ACAR-Net. (a) Our modified non-local block design for $\mathrm{HR}^2\mathrm{O}_{\mathrm{NL}}$. (b) Illustration of Actor-Context Feature Bank, where ACAR-Net* refers to the first-order relation extraction part of our proposed module.

relations at the same spatial location. We use two specific implementations for the location-wise attention. For efficiency, we by default perform 2×2 spatial max pooling on the first-order relation maps before feeding them into our operator.

(1) $\mathrm{HR}^2\mathrm{O}_{\mathrm{NL}}$ where the relation operator consists of up to three modified non-local blocks [32] (see Fig. 3 (a)). Since we are operating on a spatial grid of features, we replace the fully-connected layers in the non-local block with convolutional layers, and the attention vector is computed separately at every spatial location. Following [34], we also add layer normalization and dropout to our modified non-local block for improving regularization.

(2) $\mathrm{HR}^2\mathrm{O}_{\mathrm{GAttn}}$ where the attention is computed by a simpler graph attention mechanism [29]. In details, for calculating the attention map, we apply two convolutional layers with concatenation in the middle, and then a LeakyReLU activation as well as a Softmax function for normalization afterwards. The attention map is used as coefficients for a linear combination of the first-order actor-context relations, which gives our desired second-order actor-context-actor relations.

Relation Network. We also exploit a different instantiation $\mathrm{HR}^2\mathrm{O}_{\mathrm{RN}}$ which directly encodes second-order actor-context-actor interaction features from actor features $\{A_i\}_{i=1}^N$ and the context feature V by a Relation Network [21]. More specifically speaking, for two actors i and j, we tile the pair of actor features $[A_i, A_j]$ to over the context feature map V, and then apply two convolutions layers to reason relations from the actor-actor-context feature triplets. The high-order relation of an actor i is simply the average of all relations related to that actor. This method can be computationally expensive when the number of actors N is large, since the number of feature triplets depends on N^2.

Average. To demonstrate the effectiveness of the location-wise attention, we also experiment on a simple instantiation $\mathrm{HR}^2\mathrm{O}_{\mathrm{avg}}$ which directly outputs the average of all first-order relations.

3.3 Actor-Context Feature Bank

Inspired by the Long-term Feature Bank (LFB) [34], which creates a feature bank over a large time span to facilitate first-order actor-actor relation reasoning across a long period of video, we consider creating an Actor-Context Feature Bank F_{bank} which is built upon the first-relation features computed in our ACAR-Net. Formally, $F_{\mathrm{bank}} = [F_0, F_1, \cdots, F_{T-1}]$, where F_t is the first-order actor-context relation map extracted from a short video clip ($\sim$ 2s) around time t. As is illustrated in Fig. 3 (b), this bank of features is obtained by running an independently trained ACAR-Net over the entire video at evenly spaced intervals and saving the intermediate first-order relation maps. Different from the original LFB, our relational feature preserves the spatial context information. Equipped with such a relational feature bank, our ACAR-Net can leverage the High-order Relation Reasoning Operator described in the section above for reasoning actor-context-actor relations over a much longer time span, and thus better capture what is happening in the entire video for achieving more accurate action localization at the current time stamp.

Implementation Details. We only experiment on ACFB with the $\mathrm{HR}^2\mathrm{O}_{\mathrm{NL}}$ instantiation. We stack two modified non-local blocks mentioned in Section 3.2. We replace the self-attention mechanism in the $\mathrm{HR}^2\mathrm{O}$ with an attention between current and long-term actor-context relations.

4 Experiments

In this section, we evaluate our proposed ACAR-Net on the challenging AVA dataset [10]. We first introduce some implementation details.

4.1 Implementation Details

Dataset. AVA is a video dataset of spatio-temporally localized atomic visual actions. We use version 2.2 of this dataset by default. Its data source are 430 15-minute movie clips. Actor box annotations and their corresponding action labels are provided on key frames in these video clips with a stride of 1 second. This dataset is challenging since movie scenes are often highly complex and contain multiple actors, each of which may perform several atomic actions simultaneously. Following previous approaches, we only evaluate on 60 action classes, and the performance metric is mean Average Precision (mAP) using a frame-level IoU threshold of 0.5.

Person Detector. As for person detection on key frames, we use pre-computed human bounding box proposals from [34], which are generated by a Faster R-CNN [20] with a ResNeXt-101-FPN [16,37] backbone. The model is pre-trained with Detectron [7] on ImageNet [3] and the COCO human keypoint images [17], and then fine-tuned on the AVA dataset.

Backbone Network. We use SlowFast networks [5] as the backbone in our localization framework, and we also increase the spatial resolution of res_5 by $2\times$. We carry out our ablation experiments using a SlowFast R50 instantiation with input sampling $T \times \tau = 8 \times 8$ (without non-local) pre-trained on the Kinetics-400 dataset[1].

Training. We use per-class binary cross entropy loss as the training loss function. Since one person should only have one pose label, following [35], we apply a softmax function instead of sigmoid to the logits corresponding to pose classes.

We train all models in an end-to-end fashion (except the feature bank part) using synchronous SGD with a minibatch size of 32 clips. We freeze batch normalization layers in the backbone network. For models without ACFB, we train for 35k steps with a base learning rate of 0.064, which is decreased by a factor of 10 at iterations 33k and 34k. We find models with ACFB exhibit overfitting when using this 35k schedule, so we decrease the number of iterations for training these models to 29k. We perform linear warm-up [9] during the first 6k iterations. We use a weight decay of 10^{-7} and Nesterov momentum of 0.9. For a video clip, we use 32 frames centered at the key frame as input, sampled with a temporal stride of 2. In order to better preserve spatial structure, we do not use spatial random cropping augmentation. Instead, we only scale the shorter side of the input frames to 256 pixels, and zero pad the longer side to the same size in order to simplify mini-batch training. We use both ground-truth boxes and predicted human boxes with scores at least 0.9 for training. Following [34], we assign labels of a ground-truth box to a predicted box if they overlap with IoU at least 0.9. We use bounding box jittering augmentation, which randomly perturbs box coordinates by a scale at most 7.5% relative to the original size of the bounding box during training.

Inference. At test time, we use detected boxes with scores at least 0.85. We scale the shorter side of input frames to 256 pixels, and apply the backbone network fully-convolutionally.

4.2 Ablation and Validation Experiments

Relation Type. In order to show the importance of high-order relation reasoning, we performed experiments on different types of relation modeling applied to

[1] This pre-trained SlowFast R50 model is downloaded from the repository at https://github.com/facebookresearch/SlowFast. The SlowFast R101+NL pre-trained on Kinetics-600 mentioned below can also be found in this repository.

	mAP
3D-CNN	25.39
Actor-Actor	26.10
Actor-Context	26.71
HR^{2}O	**27.83**

(a) **Relation Type**

	mAP
Avg	26.97
RN	27.18
GAttn	27.25
NL	**27.54**

(b) **HR^{2}O Type**

	mAP
Global Pool	27.31
Pool 4x4	27.51
Pool 2x2 (default)	27.54
No Pool	**27.63**

(c) **Input Structure**

	mAP
Actor First	26.91
Context First	**27.54**

(d) **Relation Order**

	mAP
HR^{2}O$_{\mathrm{NL}}$-1L	27.54
HR^{2}O$_{\mathrm{NL}}$-2L	**27.83**
HR^{2}O$_{\mathrm{NL}}$-3L	27.25

(e) **NL Depth**

	mAP
3D-CNN	25.39
LFB	27.49
ACFB	**28.84**

(f) **Feature Bank**

Table 1. AVA ablations. 3D-CNN: a simple linear classifier after the backbone network and RoIAlign; LFB: Feature Bank Operator (FBO) with a long-term feature bank; ACFB: our proposed High-order Relation Reasoning Operator (HR^{2}O) with an Actor-Context Feature Bank. The results demonstrate that high-order relation reasoning is beneficial, and also validate various design choices of our operator.

the same backbone network. As listed in Table 1a, our proposed Actor-Context-Actor relation significantly improves over the baseline methods. We observe that adding context (Actor-Context) performs better than the Actor-Actor relation, yet our high-order relation still outperforms both types of first-order relations by a considerable margin.

HR^{2}O Function Design. We tested different instantiations of the High-order Relation Reasoning Operator described in Section 3.2. We only use 1-layer version of the location-wise attention instantiations for comparison. We can see from Table 1b that two location-wise attention mechanisms (GAttn and NL) work better than the simple average (Avg). In addition, the instantiation with relation network (RN) also performs well. Nonetheless, we select the NL instantiation as our default choice, because it is computationally lighter than RN which introduces feature triplets, and its performance is relatively better.

Input Structure. We investigate several max pooling strategies applied on the first-order relation maps which are inputs to our HR^{2}O. As shown in Table 1c, preserving more spatial structure leads to a bit better performance. Considering efficiency, we choose 2×2 max pooling as the default setting.

Relation Ordering. There are two possible orders for reasoning actor-context-actor relations - aggregating actor-actor relations first or encoding actor-context relations first. Note that our ACAR-Net is designed according to the latter order. We implemented the former order by performing self-attention over actor features before incorporating context features. The results in Table 1d validate that the latter order is indeed better than the former one, suggesting that context information be introduced earlier for better relation reasoning.

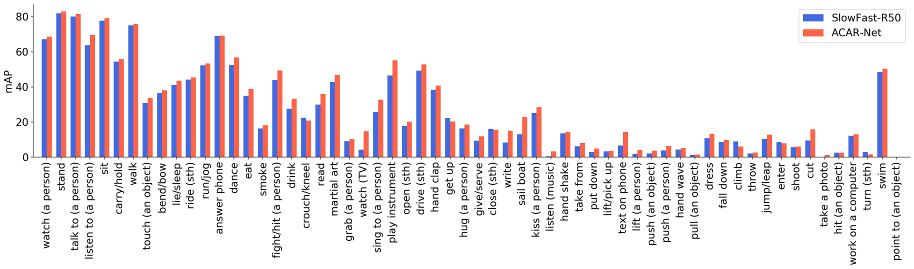


Fig. 4. Impact of High-order Relation Reasoning. We compare on AVA per-class AP between the 3D-CNN baseline (25.39 mAP) and our ACAR-Net (27.83 mAP).

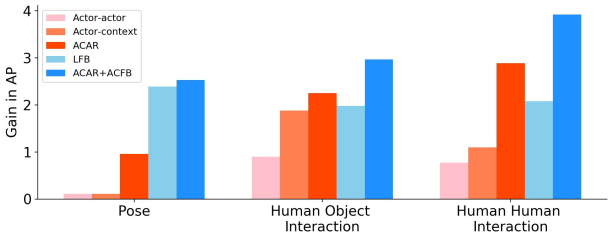


Fig. 5. Gain in mAP on three super-classes. Our ACAR-Net consistently outperforms first-order relation reasoning methods, and adding ACFB gives performances surpassing LFB on all three super-classes, especially on the interaction classes.

Deeper HR^{2}O. In Table 1e, we observe that stacking two modified non-local blocks in HR^{2}O$_{\text{NL}}$ gives higher mAP than the one-layer version, yet adding one more non-local block produces worse performance possibly due to overfitting. Note that the number (27.83) recorded in Table 1a corresponds to the best-performing two-layer setting.

Actor-Context Feature Bank. In this set of experiments, we show the effectiveness of ACFB introduced in Section 3.3. Our ACFB contains first-order actor-context relational features from 19 consecutive clips spanning 21 seconds. Compared to previous experiments, adding an ACFB to our ACAR-Net offers another significant boost in performance (27.83 → **28.84**). Moreover, as a comparison to our method, we also experimented on the long-term feature bank (LFB) [34] using the same backbone (SlowFast R50) and training schedule, under its default 2-layer setting with a temporal support of 60 seconds. As presented in Table 1f, ACFB with HR^{2}O is capable of outperforming LFB even if the latter has longer temporal support. This again highlights the importance of contextual information and high-order relation reasoning.

Category Analysis. In Fig. 4, we compare per-class performances of our ACAR-Net to the 3D-CNN baseline. The class categories are sorted in descending order

model	inputs	AVA	pre-train	val mAP
I3D [10]	V+F	v2.1	Kinetics-400	15.6
ACRN, S3D [25]	V+F		Kinetics-400	17.4
STEP, I3D [39]	V+F		Kinetics-400	18.6
RTPR [15]	V+F		ImageNet	22.3
Action Transformer, I3D [6]	V		Kinetics-400	25.0
LFB, R50+NL [34]	V		Kinetics-400	25.8
LFB, R101+NL [34]	V		Kinetics-400	27.4
SlowFast, R50, 8 × 8 [5]	V		Kinetics-400	24.8
SlowFast, R101, 8 × 8 [5]	V		Kinetics-400	26.3
AVSlowFast, R101, 8 × 8 [36]	A+V		Kinetics-400	27.8
Ours, R50, 8 × 8	V		Kinetics-400	27.2
Ours+ACFB, R50, 8 × 8	V		Kinetics-400	**28.3**
AVSlowFast, R101, 8 × 8 [36]	A+V	v2.2	Kinetics-400	28.6
SlowFast, R101+NL, 8 × 8 [5]	V		Kinetics-600	29.0
SlowFast, R101+NL, 16 × 8 [5]	V		Kinetics-600	29.8
Ours, R101+NL, 8 × 8	V		Kinetics-600	30.3
Ours+ACFB, R101+NL, 8 × 8	V		Kinetics-600	31.4
Ours+ACFB, R101, 8 × 8	V		Kinetics-700	**32.8**

Table 2. Comparison with the state-of-the-art on AVA. Note that we do not include results tested with multiple scales and flips.

according the number of training samples. We can observe that adding high-order relation reasoning brings benefit to most of the categories (53/60).

All action classes of the AVA dataset can be categorized into three super-classes: poses (*e.g.* stand, sit, walk), human-object interactions (*e.g.* read, eat, drive) and human-human interactions (*e.g.* talk, listen, hug). Fig. 5 compares the absolute gain with respect to the 3D-CNN baseline in terms of mAP on these super-classes. We can see that our high-order relation reasoning brings most benefit in the human-human interaction super-class compared to the two first-order relations, which is consistent with our motivation to model indirect relations between actors. It is also worth mentioning that with only 32 input frames spanning ∼2s, our ACAR-Net is able to excel the performance of LFB with 60s of temporal support on two interaction super-classes. Furthermore, once equipped with our ACFB, our model performs even better on those interaction classes, and can match the performance of LFB on the pose super-class.

Advanced Backbones. We replace the SlowFast R50 backbone with a SlowFast R101+NL instantiation pre-trained on Kinetics-600. With this more advanced backbone, our ACAR-Net with $\text{HR}^2\text{O}_{\text{NL}}$ reaches 30.3 mAP. We also find that adding ACFB brings a similar performance gain (30.3 → 31.4). In addition, we also experimented with a SlowFast R101 backbone (without non-local) pre-trained on the Kinetics-700 dataset, which gives even higher mAP (32.8) with our proposed method. The results are listed in Table 2.

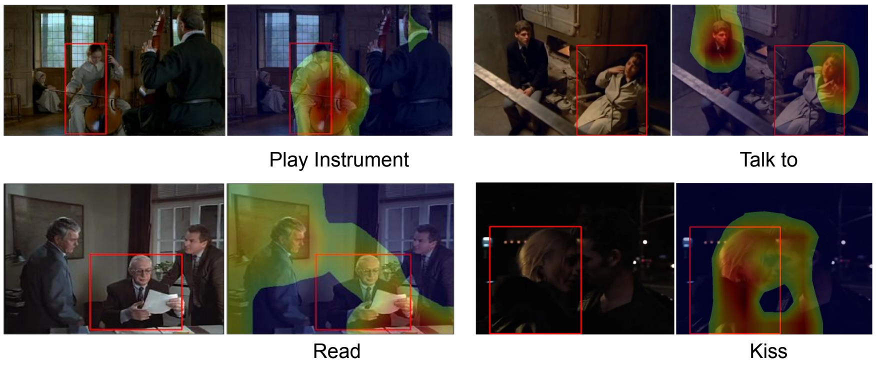


Fig. 6. CAM on high-order relation maps. We only show the heat map corresponding to the red bounding box in each frame. The action labels in correspondence to the heat maps are given below.

Comparison with the State-of-the-Art. We compare our results with state-of-the-art methods on the AVA validation set in Table 2. For comparing with some earlier works, we also used version 2.1 of the AVA dataset to train our ACAR-Net as well as its extension with ACFB on the SlowFast R50 backbone. Our best model with ACFB only dropped 0.5 mAP (28.84 → 28.34) compared to the result on AVA v2.2, showing the robustness of our proposed model against less consistent annotations. This model with **28.3 mAP** surpasses all prior results with Kinetics-400 pre-training on AVA v2.1. On the other hand, with AVA v2.2, more advanced backbones and finer pre-training, our ACAR-Net with ACFB achieves **32.8 mAP** with only single-scale testing, establishing a new state-of-the-art on AVA.

4.3 Qualitative Results

In order to verify the relations learned by our model, we leverage Class Activation Mapping (CAM) [43] to visualize the relation map generated by our High-order Relation Reasoning Operator which contributed most to correct classification. In Fig. 6, we show the heat map corresponding to one actor bounding box (marked in red) on each of the sampled key frames. We observe that our model mainly pays attention to actors and objects that are relevant to the action label.

In Fig. 7 and 8, we compare predictions made by our models and baseline methods. The examples show some positive signs that our models can utilize the spatio-temporal context to reason about indirect and high-order relations. In the first rows of the two figures, it is hard to predict the "ride" labels solely based on the actor bounding boxes, and the baselines utilizing only first-order relations all failed. Yet our models took into account context information, especially the temporal context from previous frames in Fig. 8, and successfully predicted "ride" with relatively high confidence. From the second row of Fig. 7, we can

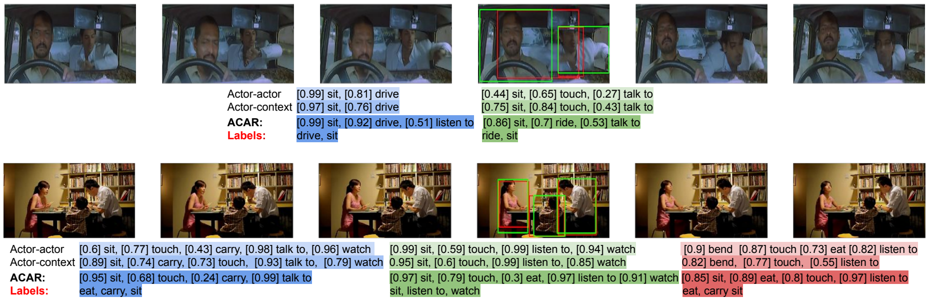

Actor-actor
[0.99] sit, [0.81] drive
[0.44] sit, [0.65] touch, [0.27] talk to
Actor-context
[0.97] sit, [0.76] drive
[0.75] sit, [0.84] touch, [0.43] talk to
ACAR:
[0.99] sit, [0.92] drive, [0.51] listen to
[0.86] sit, [0.7] ride, [0.53] talk to
Labels:
drive, sit
ride, sit
Actor-actor
[0.6] sit, [0.77] touch, [0.43] carry, [0.98] talk to, [0.96] watch
[0.99] sit, [0.59] touch, [0.99] listen to, [0.94] watch
[0.9] bend [0.87] touch [0.73] eat [0.82] listen to
Actor-context
[0.89] sit, [0.74] carry, [0.73] touch, [0.93] talk to, [0.79] watch
0.95] sit, [0.6] touch, [0.99] listen to, [0.85] watch
0.82] bend, [0.77] touch, [0.55] listen to
ACAR:
[0.95] sit, [0.68] touch, [0.24] carry, [0.99] talk to
[0.97] sit, [0.79] touch, [0.3] eat, [0.97] listen to [0.91] watch
[0.85] sit, [0.89] eat, [0.8] touch, [0.97] listen to
Labels:
eat, carry, sit
sit, listen to, watch
eat, carry sit

Fig. 7. Example predictions on AVA. We compare our ACAR-Net with two first-order relations.

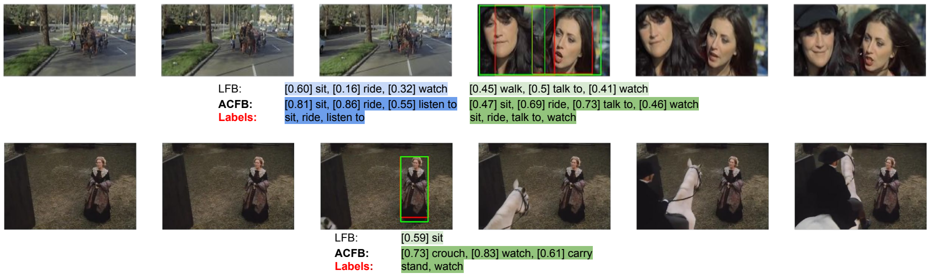

LFB:
[0.60] sit, [0.16] ride, [0.32] watch
[0.45] walk, [0.5] talk to, [0.41] watch
ACFB:
[0.81] sit, [0.86] ride, [0.55] listen to
[0.47] sit, [0.69] ride, [0.73] talk to, [0.46] watch
Labels:
sit, ride, listen to
sit, ride, talk to, watch
LFB:
[0.59] sit
ACFB:
[0.73] crouch, [0.83] watch, [0.61] carry
Labels:
stand, watch

Fig. 8. Example predictions on AVA. We make comparison between our model with ACFB and the LFB baseline.

see that our ACAR-Net gives the little girl the label "eat", although it does not appear in the ground-truth labels, and there is no direct evidence that the girl is eating. This interesting result may be due to our high-order relation reasoning. As for the second example in Fig. 8, there is only one woman in the key frame, yet after seeing the longer temporal context where the man on a horse appears, our model gives the correct label "watch" with high confidence.

5 Conclusion

Given the high complexity of realistic scenes encountered in the spatio-temporal aciton localization task which involve multiple actors and a large variety of contextual objects, we observe the demand for a more sophisticated form of relation reasoning than current ones which often miss important hints for recognizing actions. Therefore, we introduce the concept of modeling the higher-order actor-context-actor relations, which are relations between two actors based on their interactions with the context. We propose Actor-Context-Actor Relation Network for modeling such indirect relations. Extensive experiments on the action localization task show our ACAR-Net with high-order relation reasoning in videos leads to a significant performance gain and achieves state-of-the-art results on the challenging AVA dataset.

References

1. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. arXiv preprint arXiv:1907.06987 (2019)
2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
4. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2625–2634 (2015)
5. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6202–6211 (2019)
6. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 244–253 (2019)
7. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron (2018), https://github.com/facebookresearch/detectron
8. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object interactions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8359–8367 (2018)
9. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
10. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6047–6056 (2018)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
12. Hou, R., Chen, C., Shah, M.: Tube convolutional neural network (t-cnn) for action detection in videos. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5822–5831 (2017)
13. Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C.: Action tubelet detector for spatio-temporal action localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4405–4413 (2017)
14. Köpüklü, O., Wei, X., Rigoll, G.: You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. arXiv preprint arXiv:1911.06644 (2019)
15. Li, D., Qiu, Z., Dai, Q., Yao, T., Mei, T.: Recurrent tubelet proposal and recognition networks for action detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 303–318 (2018)
16. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)

17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
18. Peng, X., Schmid, C.: Multi-region two-stream r-cnn for action detection. In: European conference on computer vision. pp. 744–759. Springer (2016)
19. Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 401–417 (2018)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
21. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Advances in neural information processing systems. pp. 4967–4976 (2017)
22. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1049–1058 (2016)
23. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
24. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
25. Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R., Schmid, C.: Actor-centric relation network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 318–334 (2018)
26. Sun, C., Shrivastava, A., Vondrick, C., Sukthankar, R., Murphy, K., Schmid, C.: Relational action forecasting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 273–283 (2019)
27. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
30. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
31. Wang, T., Anwer, R.M., Khan, M.H., Khan, F.S., Pang, Y., Shao, L., Laaksonen, J.: Deep contextual attention for human-object interaction detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5694–5702 (2019)
32. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7794–7803 (2018)
33. Wang, X., Gupta, A.: Videos as space-time region graphs. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 399–417 (2018)
34. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., Girshick, R.: Long-term feature banks for detailed video understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 284–293 (2019)

35. Xia, J., Tang, J., Lu, C.: Three branches: Detecting actions with richer features. arXiv preprint arXiv:1908.04519 (2019)
36. Xiao, F., Lee, Y.J., Grauman, K., Malik, J., Feichtenhofer, C.: Audiovisual slowfast networks for video recognition. arXiv preprint arXiv:2001.08740 (2020)
37. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
38. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: Proceedings of the IEEE international conference on computer vision. pp. 5783–5792 (2017)
39. Yang, X., Yang, X., Liu, M.Y., Xiao, F., Davis, L.S., Kautz, J.: Step: Spatio-temporal progressive learning for video action detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 264–272 (2019)
40. Ye, Y., Yang, X., Tian, Y.: Discovering spatio-temporal action tubes. Journal of Visual Communication and Image Representation **58**, 515–524 (2019)
41. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4694–4702 (2015)
42. Zhang, Y., Tokmakov, P., Hebert, M., Schmid, C.: A structured model for action detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9975–9984 (2019)
43. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)

A Appendix

A.1 Experiments on UCF101-24

We further demonstrate the effectiveness of our design by evaluating it on another action localization dataset, UCF101-24.

model	mAP
MR-TS [18]	65.7
PntMatch [40]	67.0
T-CNN [12]	67.3
ACT [13]	69.5
STEP [39]	75.0
I3D [10]	76.3
YOWO [14]	87.2
SlowOnly R50, 8 × 2	88.5
Ours, SlowOnly R50, 8 × 2	90.3
SlowFast R50, 8 × 4	90.5
Ours, SlowFast R50, 8 × 4	**93.0**

Table 3. Comparison with previous works on UCF101-24.

Dataset. UCF101-24 is a subset of UCF101 [24]. It contains spatio-temporal annotations on 24 action classes. Following previous works, we experiment on the first split and report frame-mAP under IoU threshold of 0.5.

Implementation Details. We use two different backbones: a SlowOnly R50 8×2 and a SlowFast R50 8×4, both pre-trained on Kinetics-400. We use the person detector from [14]. The SlowOnly backbone takes as input 8 frames with a temporal stride of 2, and the fast branch of the SlowFast backbone uses 32 continuous frames.

For training, we use standard cross entropy loss. We train all the models end-to-end for 5.4k iterations with a base learning rate of 0.002, which is decreased by a factor of 10 at iterations 4.9k and 5.1k. We perform linear warm-up during the first quarter of the training schedule. We only use ground-truth boxes for training. For inference, we use all boxes given by the detector. Other hyper-parameters are the same as the experiments on AVA.

Results. As shown in Table 3, our proposed ACAR-Net consistently improves the two 3D-CNN baselines by large margins, which again indicates the importance of high-order relation reasoning.

A.2 Additional Experiments on AVA

We also tried an advanced backbone on AVA v2.1, which is a SlowFast R101 instantiation (without non-local) pre-trained on Kinetics-400. The results are presented in Table 4. Our best model with ACFB (**30.0 mAP**) achieves a **+3.7 mAP** increase (**14.1%** relative improvement) compared to the 3D-CNN baseline provided by [5].

model	inputs	AVA	pre-train	val mAP
SlowFast, R101, 8 × 8 [5]	V	v2.1	Kinetics-400	26.3
AVSlowFast, R101+NL, 8 × 8 [36]	A+V		Kinetics-400	27.8
Ours, R101, 8 × 8	V		Kinetics-400	28.1
Ours+ACFB, R101, 8 × 8	V		Kinetics-400	**30.0**

Table 4. Comparison with the state-of-the-art on AVA v2.1.

A.3 Pre-training on Kinetics

In our experiments, we used two SlowFast R101 models pre-trained on Kinetics-400 and 700 respectively. Other pre-trained models are already available online. We give some details on the pre-training in this section.

For pre-training on Kinetics-400, we use synchronous SGD with a minibatch size of 256. We train for 196 epochs with a base learning rate of 0.4. Other settings are the same as [5]. The model at convergence gives 76.6% top-1 accuracy on the validation set. As for Kinetics-700, we use the same batch size, yet the training schedule is extended to 288 epochs with a base learning rate of 0.115. The model at convergence gives 69.6% top-1 accuracy on the validation set.