THORN: Temporal Human-Object Relation Network for Action Recognition

Mohammed Guermal, Rui Dai, Francois F Bremond

THORN: Temporal Human-Object Relation Network for Action Recognition

Mohammed Guermal, Rui Dai, and François Brémond
Inria, Université Côte d'Azur, 2004 Route des Lucioles, 06902 Valbonne
{mohammed.guermal, rui.dai, francois.bremond}@inria.fr

***Abstract*—Most action recognition models treat human activities as unitary events. However, human activities often follow a certain hierarchy. In fact, many human activities are compositional. Also, these actions are mostly human-object interactions. In this paper we propose to recognize human action by leveraging the set of interactions that define an action. In this work, we present an end-to-end network: THORN, that can leverage important human-object and object-object interactions to predict actions. This model is built on top of a 3D backbone network. The key components of our model are: 1) An object representation filter for modeling object. 2) An object relation reasoning module to capture object relations. 3) A classification layer to predict the action labels. To show the robustness of THORN, we evaluate it on EPIC-Kitchen55 and EGTEA Gaze+, two of the largest and most challenging first-person and human-object interaction datasets. THORN achieves state-of-the-art performance on both datasets.**

I. Introduction

Human activity recognition in video is a fundamental problem in computer vision, due to its large field of applications, such as human-computer interaction [1] or video surveillance [2]. Machine learning and computer vision models have achieved interesting results in this field. Unfortunately, most of the State-of-the-art methods focus on simple activities such as *walking* or *drinking*, while the recognition of longe-term, complex, and composite activities such as *assembling furniture* or *food preparation* has been rarely addressed. These methods make use of end-to-end models that produce a video level label, and do not explicitly decompose the action into a hierarchical set of sub-actions or interactions. Moreover, neuroscience [3], [4] has shown that the human perception of action is actually based on decomposing an action into different groups of interactions which enables him/her to understand other human behaviors. In this paper we decide to visit this composite actions, that we refer to as actions of Human-Object Interaction (HOI). Not only that we also focus on first-person view HOI action recognition.

first-person action recognition also comes with its challenges, one of which is the narrow field of view that makes actions sometimes happen outside the video viewing range. Also, the huge ego-motion caused by the sharp movements of the camera can make it harder to recognize actions. Finally, in ego-vision, the field of view usually covers the human hands and an ensemble of objects. In this case, actions are generally involving interactions between the human and objects. Hence the challenge is also to recognize which of these objects are relevant to the action and which are distractors.

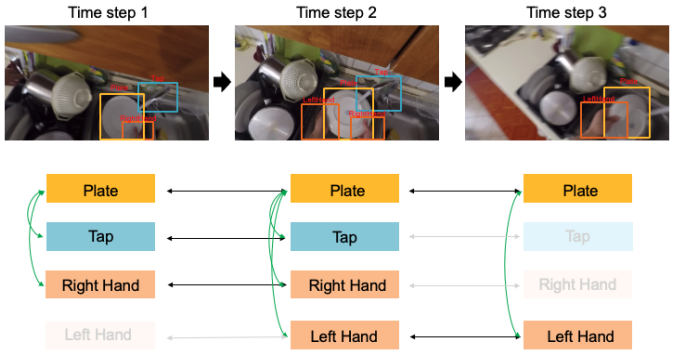

Fig. 1. An example of the Human-Object Interactions of *wash plate* in an first-view video. Green arrows represent interactions at the same time step (i.e., spatial relation) while black arrows represent interactions across time. In practice, the model captures all the objects detected. For simplicity, here we highlight only the relevant objects to *wash plate*.

A HOI action can be seen as combination of verbs and nouns, for instance the action *cutting bread with knife* is the combination of the verb *cut* and the nouns *knife* and *bread*. Hence recognizing an action of HOI, is a class of visual relationship detection, where the task is to not only recognize the objects (the noun), but also to infer the relation and motion (the verb) between different objects and the human. Fig. 1 represents an example of an object-based action: *wash plate*. Such action requires highlighting objects like the *hand*, the *plate* and the *tap* while giving less attention to other objects that are not important to the action.

Previous works such as two-stream CNNs [5],[6],[7] or 3D CNNs [8] [9] [10] have achieved very good results on third view and video level label datasets [11] [12] [13] [14]. However, when it comes to HOI actions they still lack in performance. That is due mainly to the fact that CNNs capture shareable local features in the image/videos, and they can not handle complex or fine-grained actions. Another major challenge is the fact that such activities can often be performed in a wide variety, making it harder for CNNs to learn significant patterns.

Thus, our intuition is to build a model that can, extract detailed and object specific semantics in the videos, as well as explore the cross-object relation at different time-steps. By doing so we can firstly, improve object recognition in actions of HOI (the noun). Moreover, we can refine the motion recognition (the verb) by having a clearer idea about the interaction

of these objects and their roles in the action. Finally, by encoding the scenes into a graph of objects interactions, we make it easier to learn patterns for actions even if they have many variations, since the interactions are usually the same.

To step-up to the aforementioned challenges, we propose a new module built on top of 3D-CNNs, this module is divided into two sub parts. Firstly, we design an **Object Representation filter**. This first sub-module acts as a filter that retrieves specific and object-related semantics from the overall and mixed representation (extracted from the 3D-CNN). Secondly, we add an **Object Relation Reasoning** module that uses the detailed representations to explore cross-objects relations (interactions). Finally, we obtain an object-centric model that can predict actions of HOI by exploring human-object and object-object interactions.

To summarize, our main contributions are:
1. A model that can find and extract detailed semantics of specific objects;
2- A graph-based module capable of exploring interactions between different objects.

II. Related work

Human-object interaction action recognition became the focus of many research subjects lately, especially with the development of important datasets such as [15], [16], [17]. Several approaches have been proposed to tackle this problematic. In the following, we review some of these approaches.

A. 3D-CNNs

3D-CNNs methods focus on getting the overall appearance of the videos without considering the objects interactions. Since these methods cannot capture specific or detailed semantics, they are still limited in case of actions of HOI. Making this architectures more adequate to video level labels. We cite as an example I3D [9]. Although it achieves good results on many action recognition datasets, its performance is still poor on actions of HOI. To improve the performances on these 3D-CNNs, Long Features Bank [18] for instance, tries to capture HOI actions by extracting and fusing features from local clips as well as globally from the whole video. This method uses object detection and ROI-Align to capture detected object features. And though they successfully capture richer features and more temporal information, they fail to do any object interaction modeling. Hence, they cannot improve much on HOI actions. In the same direction, Temporal Binding Networks (TBN) [19] proposes to capture local clip features from different clips and fuse them for later prediction. In addition to that, TBN uses multi-modalities as they capture audio-visual features using audio, RGB, and optical flow. However, we believe that this multi-modality will not always bring much information about the objects. sounds can be very noisy and very similar which can confuse the prediction. Moreover, fusing multi-modalities can be hard and requires lot of efforts the may not lead to significant improvements. Finally, other works such as [20] use also multi-modality reasoning. However, we argue that HOI actions recognition requires more focus on objects and their interactions.

B. Graph's Convolutions

Recently, graphs have also been considered a way for solving action recognition [21], [22], [23], [24].

As for human-object interaction, videos as a space-time region graph [21] propose to model the interaction between objects and humans in two steps as they build two different graphs. This allows to correlate objects across space-time. Similarly, in [25], the authors construct the nodes of the graph with consideration to the node class. For instance, the node for the scene is computed using the aforementioned I3D. While for objects, they use the Faster-RCNN network [26] trained on MS COCO. All these methods mentioned above try to define their nodes by using ROI-Align. However, this is not optimal as,in most cases, multiple objects are present at the scene and some of them are too close to each other. In this case, the projected coordinates of different objects tend to be in the same set of pixels. Therefore, extracting an object's specific feature from a feature map with low resolution becomes difficult. Not only that these methods rely on pre-trained object detetctors, hence they can not leverage only objects relevant to the action. Whereas in our work, we learn to filter only relevant objects and learn specific representation to different object-classes in an end-to-end way.

In the domain of semantic modelling, Class Temporal Relational Network (CTRN) [24] is proposed for the action detection tasks. However, CTRN is a two steps method, which is built on top of pre-extracted flattened 1-dimensional features. The dissociation between the visual encoder and temporal module makes the model overlook the appearance and spatial information in the video, while such information is critical to the HOI action recognition. In this work, we propose a one-step method THORN for HOI action recognition. Different from CTRN, our method leverages the object detector to extract the object semantics directly from the spatio-temporal features. After that, graph reasoning is applied to refine the object representation and to jointly model inter-object relations. This design allows the model to capture the latent relations among the objects in the videos, which results in higher accuracy in HOI action recognition.

III. Proposed Method

In this section, we detail each sub-part of the proposed model, THORN. The main components in this model are: a **3D Visual Encoder** which encodes the video into a spatio-temporal embedding. Then, the previously extracted embeddings are passed to the **Object Representation Filter** (ORF). This filter extracts class-specific features. Finally, the **Object Relation Reasoning** module computes the relation between the different objects to predict the action. Fig. 2 provides an overview of the model.

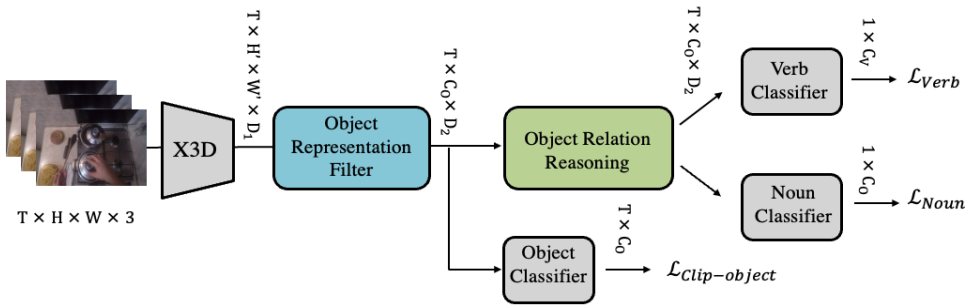

Fig. 2. THORN architecture contains three main components: (1) a **Visual encoder** (i.e., X3D) encodes the input RGB clip into a primary spatio-temporal representation. (2) The obtained representation is fed to the **Object Representation Filter**, which maps the previous representation into object-class representation. To ensure a discriminative object representation, an object classifier is added on top of the object-class representation. This classifier is trained with the pseudo-object ground truth provided by an object detector. (3) The object-class representation is also sent to the **Object Relation Reasoning** module to model the temporal-object relation in a dissociated manner. Finally, two classifiers are used to predict the verbs and nouns relevant to the action.

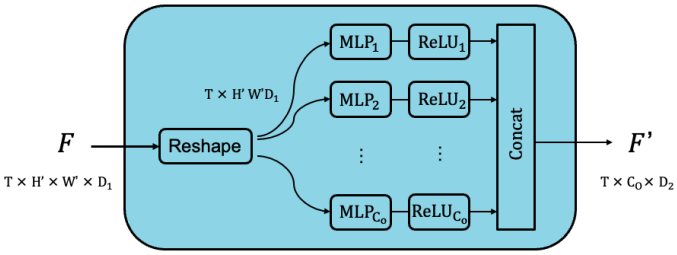

Fig. 3. Representation of our Object Representation Filter (ORF). The input is the feature map from the 3D encoder reshaped to $T \times H'W'D$ and the duplicated C_o times, where C_o is the number of classes. Finally, we have a representation specific to each object class.

A. Visual Encoder

We start by using a visual encoder to extract an embedding that serves as a full understanding of the scene, and carries the global information of the input frames. We choose X3D [27] as our visual encoder. X3D has many advantages as it does not do any temporal pooling and keeps the full temporal information, providing richer temporal information. Moreover, X3D is a lighter model compared to other architectures such as I3D [9]. The input to the 3D encoder is a set of video-clip frames. The output is a spatio-temporal representation F of shape $(T \times H' \times W' \times D_1)$, where: $H' = W' = 7$, $D_1 = 432$, while T is the same as the input.
This embedding carries both spatial and temporal information. The spatial information is important, as it provides object related information, such as its appearance, shape and position (e.g. drawers usually appear at the bottom of the image). That is why instead of using the X3D final output of shape $(T \times 2048)$ to construct our nodes, we use a finer spatial representation of shape $(T \times 7 \times 7 \times 432)$, making nodes of our graph contain more and finer information about the objects. We provide more details on this in the ablation study, by comparing both settings. Finally, as X3D is a light-weighted model it is easier to train the *Visual Encoder* jointly with the following modules.

B. Object Representation Filter

Our main objective through this work is to have object-based reasoning. Hence the first step is to obtain object in scene representations. Therefore, we developed the *Object Representation Filter* module, capable of extracting semantic representation specific to each object class from the previous overall representation. This module serves as a filter to obtain the object-specific representation from the output of visual encoder. In practice, firstly, we reshape the representation F from the visual encoder to shape $(T \times H'W'D_1)$. After that, we duplicate the reshaped features F' for C_o times, where C_o indicates the number of object classes in the dataset. For each class, we use a channel-mixer MLP (i.e., linear transformation layer), followed by non-linear activation and dropout. In Fig. 3, we show an overview of the ORF module. We argue that each MLP layer learns to filter features specific to a certain object class. The equations in this module can be formulated as:

$$F'_i = ReLU(MLP(F)) \tag{1}$$

$$F' = DropOut([F'_1, F'_2, F'_3 ...., F'_{C_o}]) \tag{2}$$

With $F' \in \mathbb{R}^{T \times C_o \times D_2}$. Where D_2 is smaller than D_1 to shallow the channel size. Finally, we add another MLP layer on top of F' that would represent the object classifier in Fig. 2.

$$F'' = ReLU(MLP(F')) \tag{3}$$

Here $F'' \in \mathbb{R}^{T \times C_o \times 1}$. To ensure the object-specific representation, we add a frame-level object classifier on F''. As the frame-level object label is not provided by the dataset, the object classifier is trained with the pseudo label provided by an object detector (i.e. Fast-RCNN [26]). In the video, multiple objects can appear in a frame, thus, we train the object classifier with binary cross-entropy loss: $\mathcal{L}_{clip-objects}$. Finally the ORF module outputs a representation for each object-class. However, we still need to correlate and refine these object representations to explore their interactions and model the actions. To do so, we introduce the next module of our pipeline in the next section.

C. Object Relation Reasoning Module

To correlate between the aforementioned representations in the previous section, we introduce the *Object Relation Reasoning Module*.

In order to extract the relations between the filtered object classes, we propose to make use of graph convolutions. In the previous section, we transform the clip representation into a class-specific representation. Then, we map it to a graph-like structure, where each vertex of the graph represents an object class at a time step; the vertex would be the previously extracted embedding of a certain class. In total, the graph consists of $C_o \times T$ nodes whose topology is defined by its vertex and an adjacency matrix A'_{C_o}. The adjacency matrix represents the connectivity or relation between the different nodes (objects) and its weights represent how strong their relationship is at different time steps. Fig. 4 represents an overview of this module.

1) ***Graph reasoning****:* The graph reasoning aims to do cross-class reasoning on the previously constructed graph. The objects relations are video dependent, and so multiple GCN blocks are stacked to learn multiple levels of semantics. Moreover, the adjacency matrix is also parameterized so that it can be learned and optimized with the pipeline during the training phase. Moreover, it can learn to adapt to the data itself. We also make use of self-attention mechanisms. Consequently, our adjacency matrix learns better to differentiate class relations owing to different videos. Fig. 4 represents a block of the graph convolution reasoning.

As the object relations are complex, it is hard to predefine the inter-object relations for each video. Therefore, by leveraging the self-attention mechanism [28], [22], our graph adjacency matrix is learnable and can vary with the videos. In practice, the adjacency matrix A_{C_o} is initialized with a fully connected matrix. Finally, the full topology of our graph is $A_{C_o} \in \mathbb{R}^{C_o \times C_o}$ and the vertexes representation $G_{in} \in \mathbb{R}^{D_2 \times T \times C_o}$. First, we embed the input G_{in} using bottleneck convolutional layer (i.e. 1 × 1), then the output feature maps are rearranged into $\mathbb{R}^{D_2 \times T \times C_o}$ and $\mathbb{R}^{C_o \times D_2 \times T}$ followed by a matrix multiplication. The value of the resultant matrix is then normalized by a softmax activation. Now, the superimposed adjacency matrix A'_{C_o} can be formulated as:

$$A'_{C_o} = A_{C_o} + softmax(W_1^T G_{in}^T W_2 G_{in}) \tag{4}$$

Where W_1 and W_2 are learnable weights of the bottleneck convolutions, and G_{in} being F' the stacked class representations in section B. G_{out}, the output of the graph layer is passed to the next graph layer and follows the same equations. In this work, we use 5 blocks of graph convolutions. As for the A'_{C_o}, each value represents an edge between two nodes (objects). We learn a graph that is shared across different time-steps but depends on each layer and for each video, as we said earlier we learn different semantics at each level.

After bottleneck convolutions, we do the graph convolution operation with the formulation in [29]:

$$G_{out} = A'_{C_o} G_{in} W_3 \tag{5}$$

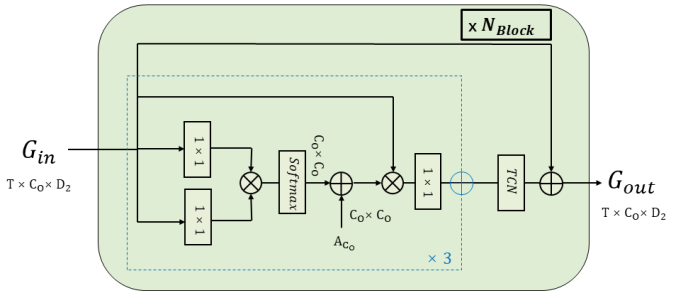

Fig. 4. Overview of one layer of the Object Relation Reasoning module, using a graph architecture [22]. As we can see, the input is a graph representation between different classes and the output is an updated representation of the graph. The $\times N_{block}$ stands for the number of blocks used in total, while the $\times$ 3 at the bottom in blue stands for the number of used multi-head attentions.

W_3 is a learnable parameter where $W_3 \in \mathbb{R}^{D_2 \times D_2}$. The equation 5 represents the message passing and node feature updating, and finally G_{out} is rearranged to $\mathbb{R}^{D_2 \times T \times C_o}$.

From equation 5, we can understand how graph convolutions work. The graph convolutional layer represents each node as an aggregate of its neighborhood, hence each node gathers information from its neighborhood and adapts itself accordingly. In other words, at each graph block, each object collects information about other objects and finally finds to which it is most correlated, and thus whether there is an interaction or not. That is why we judge that the use of graphs is a promising idea in this domain.

2) ***TCN****:* stands for Temporal Convolution Network. The graph reasoning is capable of extracting the relation between objects. However, in our study, we aim at modelling the spatio-temporal interaction in a large time span. To do so, we add a 1D convolution layer on top of the previous output of *the graph reasoning* (i.e., G_{out}). As shown in Fig. 4, each *Object Relation Reasoning Module* contains a *TCN*. This 1D-convolution layer is used to aggregate the information across time. While stacking multiple object relation reasoning blocks, each block is used to model the object relation in a specific temporal scale. Finally, the output of the *Object Relation Reasoning Module* is:

$$G_{out} = Conv1D(G_{out}) + G_{in} \tag{6}$$

As mentioned earlier, the output of each block G_{out} is the input G_{in} to the next block.

D. Predictions

Predictions are based on the learned nodes and adjacency matrix. However, since in our case our actions are composed of verbs and nouns, we show that using the adjacency matrix for predicting the verb and the object feature representation for noun prediction is more effective. This makes sense since the adjacency carries more information about how different objects interact with each others, while the nodes carry a refined object representations, after been processed through the different graph convolutions blocks. Our final layers are two fully-connected layers one projecting G_{out} from $\mathbb{R}^{D_2 \times C_o}$

TABLE I

ABLATION STUDY ON DIFFERENT SETTINGS. THIS EVALUATION IS ON EPIC-KITCHEN DATASET. TEMPORAL NODES MEANS USING THE FINAL OUTPUT OF X3D OF SIZE $T \times 2048$ TO CREATE NODES, WHILE SPATIO-TEMPORAL NODES MEANS USING A MID LAYER OF SIZE $T \times 7 \times 7 \times 432$ WITH MORE SPATIAL INFORMATION. FINALLY ADJ-MATRIX STANDS FOR USING THE ADJACENCY MATRIX FOR PREDICTING THE VERBS INSTEAD OF USING ONLY NODES FOR NOUNS AND VERBS.

	verbs		nouns		actions	
	top1	top5	top1	top5	top1	top5
X3D	46.5	79.8	34.3	65.3	21.0	38.7
THORN/temporal nodes	55.8	82.86	39.9	66.37	26.8	44.0
THORN/temporal nodes + ADJ-matrix	60.3	86.0	41.1	66.9	30.1	47.3
THORN/spatio-temporal nodes + ADJ-matrix	61.0	85.9	42.9	67.9	30.5	47.5

to $\mathbb{R}^{1\times C_o}$, and the other fully-connected layer projecting A'_{C_o} from $\mathbb{R}^{C_o\times C_o}$ into $\mathbb{R}^{1\times C_v}$, where C_o and C_v stand for the number of object classes and verb classes respectively.
Since we have 3 outputs, our loss is a sum of three losses and can be formulated as :

$$\mathcal{L} = \mathcal{L}_{verbs} + \mathcal{L}_{nouns} + \mathcal{L}_{clip-objects} \tag{7}$$

Where $\mathcal{L}_{verbs}$ and $\mathcal{L}_{nouns}$ are the negative log-likelihood losses (since each action is composed of one verb and one noun). As described earlier, the $\mathcal{L}_{clip-objects}$ is the loss to ensure the semantic of the object representation.

IV. EXPERIMENTS

Dataset. We have evaluated our model on two of the largest and challenging datasets for first-view and human-object interaction action recognition. **Epic-Kitchen55** [30] contains 55 hours of recording of 32 different kitchens in 4 cities. This dataset has a total of 125 verbs and 352 nouns. **EGTEA Gaze+** [31] contains 28 hours of cooking activities from 86 unique sessions of 32 subjects, with over 10k video clips of 106 fine-grained egocentric activities. In both datasets, each action is a combination of a verb and a noun. Actions are relevant to different steps of preparing food (e.g. *cleaning the kitchen*, *cutting vegetables*, *preparing table*).
Implementation. We implement our method using X3D as the visual encoder where D_1 = 432, H'= W'= 7 and D_2 is 128. We input a clip of 16 RGB frames for Epic-Kitchen and 25 frames for EGTEA Gaze+. We use a dropout probability of 0.3. For the *object relation reasoning* module, N_{Block} is 5 blocks.

For the temporal convolution network, we run our model with different values of the kernel size. As there was no impact on the results, we kept a kernel size of 9. In training phase, we utilized Adam [32] to optimize the model with an initial learning rate of 0.00005. We scaled the learning rate by a factor of 0.1 with the patience of 5 epochs. The network was trained on a 4-GPU machine for 30 epochs. We evaluated our model using top1 and top5 accuracy on verbs and nouns for Epic-Kitchen, while for EGTEA Gaze+ we evaluated directly on actions using top 1 accuracy.

A. Ablation Study

In this section, we validate our model design for the modules in the THORN. The evaluation is conducted on the EPIC-Kitchen dataset. We propose different settings, and see how each setting can improve the performance. In table I, we can notice different results:

Firstly, we compare our baseline model X3D with THORN. Note that, in THORN, the graph nodes can be constructed either using the output of the last layer of X3D (temporal nodes) or using its intermediate layer (spatio-temporal nodes). Here, we first compared X3D with THORN (temporal nodes), i.e., we construct the nodes by the features in shape $T \times 2048$. In this setting, nodes would serve to predict both verbs and nouns. In this scenario, we improve nouns prediction by **+5.6%**, while, the verbs accuracy increased by **+9.3%** . Proving the importance of the cross-object reasoning, compared to only capturing visual information from 3D-CNNs.
Secondly, we study the importance of the adjacency matrix for predicting the verbs. To do so, we use the adjacency matrix (ADJ-matrix) to predict verbs, while keeping the nodes to predict the nouns. In this setting, the verb prediction improves by **+4.5%** compared to the previous setting and by **+13.8%** to the baseline X3D. This is because the adjacency matrix captures the object interaction, hence, it is more suitable for verb prediction.
Thirdly, we study the effect of changing the temporal nodes with the spatio-temporal nodes. Spatio-temporal nodes are the nodes constructed by the middle layer of X3D which contains the spatial information ($T \times 7 \times 7 \times 432$). With spatio-temporal nodes, THORN improves **+1.8%** on nouns. This is because, with spatial dimensions, the ORF can better capture the object relative locations and the size of the object, then embed them in the node representation. As a result, the noun accuracy improves. This setting also brings **+0.7%** improvement on verbs.
Our overall architecture obtains **+13.8%** more accuracy on verbs and **+8.6%** on nouns w.r.t. vanilla X3D. This reflects the importance of our proposed modules in THORN and how an object-centric method can improve results on human-object interaction actions.

We then study the components for predicting the nouns in our model. In table II, we show that fusing scores of object detection and the scores obtained by the THORN nodes representation works better than using only one of them. We also find that predictions using only our model are better than the object detector itself. This shows that our model can refine the objects represented by the other objects (nodes) using our graph-based module.

TABLE II

ABLATION STUDY ON FUSING THE SCORES OF THORN WITH THE SCORES FROM THE OBJECT DETECTOR (FASTER RCNN). THIS EVALUATION IS ON EPIC-KITCHEN DATASET. FUSING BOTH SCORES BRINGS SIGNIFICANT IMPROVEMENT ON TOP-1 ACCURACY. FOR THE OBJECT DETECTOR, WE USE AN AVERAGE POOLING ON ALL THE VIDEO CLIP FRAMES OBJECT DETECTION SCORES AND ADD A THRESH-HOLD OF 0.3

Faster-RCNN scores	THORN	Nouns
✓	×	31.5
×	✓	32.8
✓	✓	42.9

B. Comparison with the State-of-the-Art

We then compare our proposed method with the state-of-the-art methods on EPIC-Kitchen and EGTEA Gaze+ in table III and IV.

In Table III, we compare our results with the state-of-the-art methods. Among these methods, Long Features Bank (LFB) [18] proposes to use global as well as local features for action recognition. To do so, they extract features on both clip and video levels, and combine them to have a better understanding of the scene. Nevertheless, this method still lacks accuracy for the objects. Moreover, LFB is a two step method which trains separately an object and verb recognizer modules. For our THORN, we train a single model for predicting both entities. As a result, we have a **+8.5%** improvement on top 1 nouns and a **+4.9%** w.r.t. LFB on action recognition.

Our method achieves the overall best performance. We claim that AssembleNet++ utilizes additional modality such as optical flow in both training and inference time. Even though, we still have the lead in top 1 accuracy for the verbs, nouns and actions, which proves again that having an object-centric and specific reasoning on object interactions is a key solution for having a better action recognition on HOI datasets. Finally, our results prove that using only RGB with an object-centric model achieves better or similar results compared to methods relying on heavy multi-modality reasoning.

In table IV, we compare our method with the state-of-the-art on EGTEA Gaze+ dataset. We have the best accuracy w.r.t. the others methods, which shows the generalization and robustness of our model on actions of HOI.

To sum up, compared to other methods, ours is lightly weighted as we use X3D, while other methods rely on heavy 3D-CNNs such as I3D. THORN is trained jointly on nouns and verbs as opposed to other methods such as LFB [18], and we only need RGB frames and object classes per-frame.

C. Qualitative Study

In this section, we conduct a qualitative study of THORN. In Fig. 5, we show the impact on some classes after adding our proposed module w.r.t. vanilla X3D. In EPIC-Kitchen, we significantly improve accuracy on 28 verb classes. Only the accuracy of 3 out of 125 verbs decreases, while the decrease is negligible. This improvement on verbs shows

TABLE III

COMPARING THORN MODEL WITH OTHER STATE-OF-THE-ART METHODS ON THE VALIDATION SET. EVEN THOUGH SOME OF THESE COMPARISONS ARE NOT FAIR SINCE THESE MODELS ARE USING MULTI-MODALITIES, WE STILL HOLD THE OVERALL BEST ACCURACY, WHICH SHOWS THE STRENGTH OF OUR MODEL

Model	Obj	RGB	Flow	Audio	Verbs	Nouns	Actions
					top1	top1	top1
Baradel[33]	×	✓	×	✓	40.9	-	-
3D-CNN	×	✓	×	×	49.8	26.1	19.0
STO[18]	✓	✓	×	×	51.0	26.6	19.5
LFB[18]	✓	✓	×	×	52.6	31.5	22.8
AssembleNET++ ODF+SDF[20]	✓	✓	✓	×	**60.0**	37.1	25.2
THORN	✓	✓	×	×	**61.0**	**42.9**	**30.5**

TABLE IV

COMPARING THORN MODEL WITH OTHER STATE-OF-THE-ART METHODS ON EGTEA GAZE+ SPLIT1. WE HOLD THE BEST ACCURACY ON ACTIONS

	Two-stream	I3D [9]	TSN [34]	ego-rnn [35]	LSTA [36]	SAP [37]	**THORN**
ACC %	43.8	54.2	58.0	62.1	62.0	64.1	**67.5**

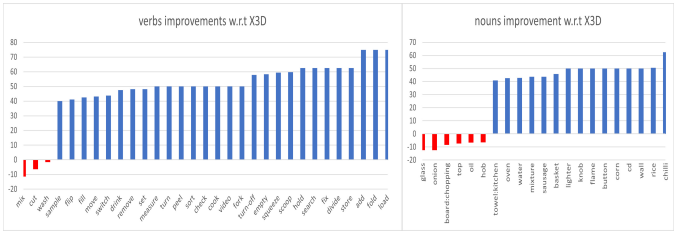

Fig. 5. Accuracy improvement on nouns (right) and verbs (left) w.r.t X3D.

that understanding the inter-relation of different objects is important for HOI.

For noun recognition, it is interesting to find that THORN can now predict some classes such as *water* and *wall*. These classes are barely detected with the object detector. This is a result of the reasoning process on cross object classes, which refines the nodes and can finally predict overlooked object classes.

We provide more interesting qualitative studies in the supplementary materials.

V. CONCLUSION

First-view action recognition relies on capturing the visual relationships between different objects and the human. In this work, we propose an object-centric model, which projects the standard CNN features into object class-specific features. After that, we compute the inter-object relations in graph reasoning, where each node corresponds to an object class and each edge represents the relation between two different objects. We evaluate our model on two large and challenging datasets. THORN achieves state-of-the-art performance on both datasets, which shows the effectiveness and robustness of our method. As our method relies on object detection precision, our future work aims at developing an architecture that can combine object detection and action recognition tasks. We also want to extend our model for first-view action detection for untrimmed video.

REFERENCES

[1] X. Jiang, K. Xu, and T. Sun, "Action recognition scheme based on skeleton representation with ds-lstm network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2129–2140, 2019.

[2] T. V. Nguyen and B. Mirza, "Dual-layer kernel extreme learning machine for action recognition," *Neurocomputing*, vol. 260, pp. 123–130, 2017.

[3] R. G. Barker and H. F. Wright, "One boy's day; a specimen record of behavior." 1951.

[4] ——, "Midwest and its children: The psychological ecology of an american town." 1955.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.

[6] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4768–4777.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[8] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[9] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[10] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks in: Proceedings of the ieee conference on computer vision and pattern recognition," 2018.

[11] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *2011 International conference on computer vision*. IEEE, 2011, pp. 2556–2563.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[14] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[15] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[16] K. Alahari, "Actor and observer: Joint modeling of first and third-person videos," in *Proceedings of the 1st Workshop and Challenge on Comprehensive Video Understanding in the Wild*, 2018, pp. 3–3.

[17] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," in *European Conference on Computer Vision*. Springer, 2012, pp. 314–327.

[18] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 284–293.

[19] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5492–5501.

[20] L. Wang and P. Koniusz, "Self-supervising action recognition by statistical moment and subspace descriptors," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4324–4333.

[21] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 399–417.

[22] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 026–12 035.

[23] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[24] R. Dai, S. Das, and F. Bremond, "CTRN: Class Temporal Relational Network For Action Detection," in *The British Machine Vision Conference*, Virtual, United Kingdom, Nov. 2021.

[25] P. Ghosh, Y. Yao, L. Davis, and A. Divakaran, "Stacked spatio-temporal graph convolutional networks for action segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 576–585.

[26] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[27] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 203–213.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[30] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[31] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 619–635.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, "Object level visual reasoning in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 105–121.

[34] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[35] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," *arXiv preprint arXiv:1807.11794*, 2018.

[36] S. Sudhakaran, S. Escalera, and O. Lanz, "Lsta: Long short-term attention for egocentric action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9954–9963.

[37] X. Wang, Y. Wu, L. Zhu, and Y. Yang, "Symbiotic attention with privileged information for egocentric action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 249–12 256.