Hand-Object Interaction Reasoning

Jian Ma
University of Bristol
United Kingdom
jian.ma@bristol.ac.uk

Dima Damen
University of Bristol
United Kingdom
dima.damen@bristol.ac.uk

Abstract

This paper proposes an interaction reasoning network for modelling spatio-temporal relationships between hands and objects in egocentric video. The proposed interaction unit utilises a Transformer-style module to reason about each acting hand, and its spatio-temporal relations to the other hand as well as objects being interacted with. We show that modelling two-handed interactions are critical for action recognition in egocentric video, and demonstrate that by using positionally-encoded trajectories, the network can better recognise observed interactions. We train and evaluate our proposed network on large-scale egocentric EPIC-KITCHENS-100 and crowd-sourced Something-Else datasets, with an ablation study to showcase our proposal.

1. Introduction

Different from general actions (e.g. jumping), object interactions involve actors influencing objects (e.g. playing an instrument or kicking a ball). Of particular interest to this work is hand-object interactions (HOI) which feature regularly in the activities of daily living. HOIs include one-handed (e.g. "open drawer") as well as two-handed interactions (e.g. "open bottle"), and many interactions include tools that extend our hands' abilities (e.g. cutting a vegetable requires a knife). However, most video understanding methods aim to recognise both actions and interactions alike as general video datasets involve a mix of classes [3, 16]. Recently, a handful of large-scale datasets [15, 5, 6] that focus on HOIs have fueled works that specifically reason about interactions.

Recent progresses in interaction reasoning have been driven by the success of object detectors, e.g. [26]. Due to the datasets used, previous works [30, 13, 24, 1] detect the person in the middle frame of the video or all the objects that appear in the video. This means that trajectories of interactions are not explicitly emphasised.

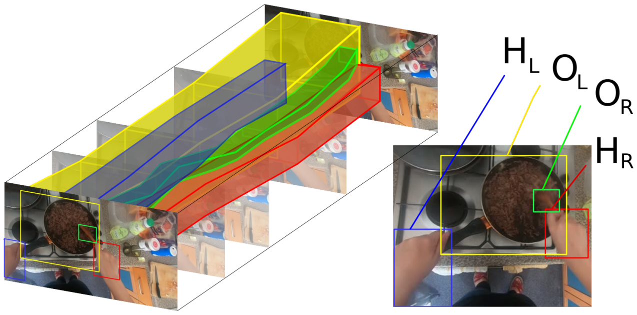


Figure 1. Illustration of Spatio-Temporal HOI reasoning for the action of stirring food in the pan. Both hands play critical roles. The left hand (H_L, in blue) steadies the pan, which we refer to as the left object (O_L, in yellow) given its direct interaction with the left hand. The right hand (H_R, in red) holds the wooden spoon (O_R, in green). While H_L, O_L are steady over time, they are action critical as the red and green trajectories (H_R, O_R) demonstrate the stirring motion relative to the pan.

In this work, we focus on the interaction between hands and objects, through their trajectories that encode motion and positions, thus discriminating between different interactions (Fig. 1). We propose an **I**nteraction **R**easoning **N**etwork (IRN) that jointly reasons about interactions between both hands and active objects.

We propose encoders that learn hand-object and hand-hand interactions and decoders that enrich this learned interaction with action representation knowledge. We automatically detect hands and active objects, i.e. those with which the hand interacts, using the approach from [27], and link these detections over time to form trajectories with pooled features from a spatio-temporal backbone. We then reason about pairwise interactions, distinguishing left from right hand interactions as well as global motion. The proposed framework is trained end to end.

Our contributions: (i) this paper proposes to separately reason about interactions between the left and right hands with their corresponding objects in HOI through an encoder-decoder transformer module; (ii) to the best of the authors' knowledge, the proposed method is the first work that uses the trajectories of hands and objects to enrich the

978-1-6654-6382-9/22/$31.00 ©2022 IEEE

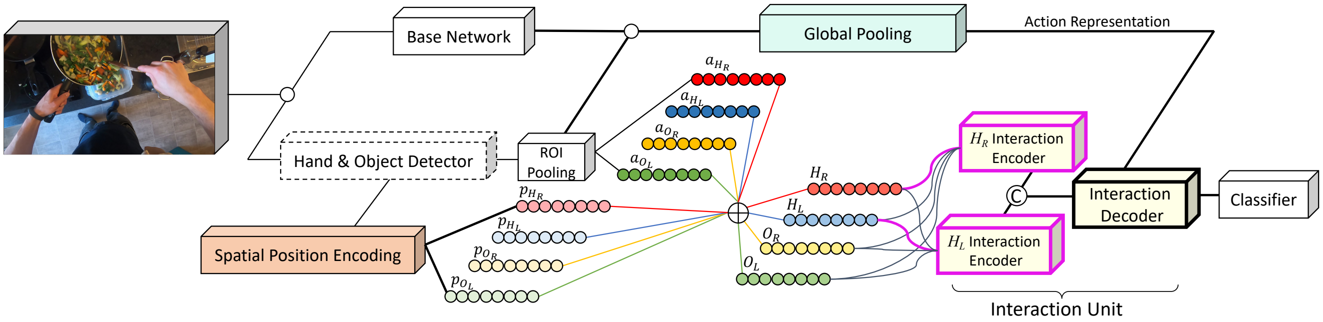


Figure 2. Proposed Architecture for Interaction Reasoning Network (IRN). A base 3D ConvNet extracts spatio-temporal features. These are ROI pooled based on hand-object detections and combined with positional encoding to form trajectories. The trajectories are fed into interaction unit to reason about hand-object relations. A decoder combines the action representation and two-hand encoders for classification (Fig. 4 for details) . ○ is for multiple outputs, ⊕ is for summation and C for concatenation. Dashed box for 'Hand & Object Detector' highlights frozen weights. Thicker lines highlight backpropagation pathways.

relational representation of interactions for action recognition; (iii) we showcase the importance of this proposed reasoning on the large-scale egocentric HOI dataset EPIC-KITCHENS-100 [6] as well as on the crowd-sourced HOI dataset Something-Else [22].

2. Related Work

Action Recognition. In recent years, the success of deep learning in computer vision has promoted the rapid development of action recognition models. From the 2D ConvNets [38, 7, 34, 39, 20] and multi-stream networks [28, 9, 3] to 3D ConvNets [18, 31, 10, 8] and transformer-based networks [23, 2], these models have progressively improved the understanding of actions. Of relevance to our model, SlowFast network [10] is a 3D convolutional model that can combine the spatial features with temporal information from two rates of sampling the input video. SlowFast results remain competitive particularly for HOI datasets we analyse. These approaches form the base for interaction reasoning which we review next.

Interaction Recognition. Recently, interest in actor-centric video understanding employed increasingly reliable person detectors, to localise actors in movies [16] or players in sports [29]. As a branch of action recognition, interaction reasoning has also received significant attention in recent years. Many works [14, 4, 21, 33, 11, 12, 24, 37, 13, 22, 27, 40] utilise an object detector (*e.g.* Faster R-CNN [26]) to locate bounding boxes of human or objects for input into a relational module. However, the detector is likely to yield object proposals that are not relevant to the interaction due to the lack of annotations for training. Instead, [14] learns to predict the location of related objects based on the appearance of actors. Similarly, [4, 11] introduce pairwise streams for interaction patterns to encode the spatial relative locations of human and objects. Based on pairwise streams, human poses are considered for modelling HOIs in [21, 33]. Alternatively, Graph Convolutional Network (GCN) can explore image-based object interactions [40, 12].

Different from image-based methods, the motion is also particularly important in video analysis. Given person detections from the middle frame of a video, in [13], detections are pooled as a query to attend to the whole frame's 3D features, in a transformer encoder block. Longer-term reasoning is proposed in [37] by learning contextual information through short-term person feature banks and long-term feature banks from non-local blocks [35]. Apart from the contextual interactions in the temporal dimension, [1] studies relational interactions between objects via training a Gate Recurrent Unit with pairing current and previous frames. Similarly, [36] regards video as a graph of objects, conducting interaction recognition reasoning over the graph. Moreover, [22] generalises the performance to unseen actions, decomposing each action into a verb, subject, and one or more goals and proposing the Something-Else dataset for exploring hand-object interactions. Detections over time are combined through the Hungarian algorithm to track the detected persons and objects. However, detected objects may not be part of the interaction, which introduces noise in all but the simplest scenes. Besides, expensive-to-collect annotations of objects are required to train the model.

Closest to our work, [25] proposes to focus on positions and sizes of interacting objects and learns an encoder-decoder that attends to 3D action features. The proposed CACNF [25] is trained on Something-Else dataset [22] and it performs SoTA by ensembling a spatio-temporal model and 3D CNN. Our proposed IRN is inspired by [1, 13], but we focus on hands as actors, thus requiring to model two-hand interactions including hand-to-hand interactions as well as hand-to-object interactions. Different from [22, 25], **we do not require additional annotations** and instead use automatic, thus potentially noisy, detection of hands and active objects to reason about interactions in busy scenes efficiently. We detail our method next.

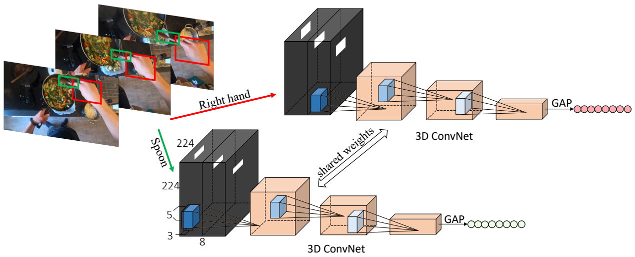


Figure 3. Spatial Position Encoder. Detections, per object, over a sequence are represented by binary maps. A 3-layer 3D ConvNet produces per-frame spatial encoding.

3. Methodology

In this section, we give an overall of the Interaction Reasoning Network (IRN) architecture (visualised in Fig. 2). We then discuss detections and the spatial position encoder followed by details of our proposed interaction unit (IU).

3.1. Network Overview

IRN captures the relationships between interacting hands and active (i.e. action-relevant) objects in HOIs. A base 3D convolutional network is utilised to extract spatio-temporal representations over the action. We utilise this in two ways. First, similar to standard approaches, we pool the spatio-temporal features for a global action representation F. Additionally, we propose an interaction unit, where spatio-temporal features are pooled within hand H and object O detections along with their spatial positional representations P. We model interactions through i) encoders that focus on hand-object and hand-hand interaction; and ii) decoders that enrich the action representation with the encoded hand interactions, trained jointly.

To get a better explanation of our proposal, we formulate the interaction unit as:

$$\begin{aligned} E &= e(H, O, P; \omega_e), \\ I &= d(F, E; \omega_d), \end{aligned} \tag{1}$$

where ω_e and ω_d represent the parameters of encoders $e()$ and decoders $d()$. The encoder aims to model the hand H and interacting object O representations along with their spatial position encoding P. The encoders' output, E, and the pooled action representation F are fed into the decoders.

3.2. Detections and Positional Encoding

We use the hand-object detector [27] that aims to find links between action performers (hands) and active objects by optimising offset vector [14]. It not only detects left/right hands and their active objects instead of segmenting all objects in one frame, but also yield directions and distances between hands and contacted objects. We use a pretrained hand and object detector [27] to predict hands and active object detections per frame on EPIC-KITCHENS 100 [6].

We distinguish four bounding box detections of $\{b_{H_L}, b_{H_R}, b_{O_L}, b_{O_R}\}$ where H/O mean hand and object, L/R denote the side, left or right. From these detections, we extract frame-level hands and objects features. We adopt a RoI average pooling that can extract the features based on the output of the backbone as in [1, 13]. Given a layer in the backbone with C channels, we extract a feature a per detection of size $\mathbb{R}^C$ by RoI pooling, of which we have $\{a_{H_L}, a_{H_R}, a_{O_L}, a_{O_R}\}$.

In addition to the features, the absolute positions of hands and objects offer significant information to distinguish interactions. We thus propose to use spatial positional representation p for each detection, such that $\{p_{H_L}, p_{H_R}, p_{O_L}, p_{O_R}\}$. Inspired by [17], we consider a binary map for each bounding box detection b, per frame. These show the absolute position and scale of an object in the image. We learn position (and scale) encodings from this binary input and use the consecutive binary maps to capture the positional/scale changes. In order to represent the absolute positions of objects, we design a weights-shared spatial position Encoder (SPE), as shown in Fig. 3, that consists of convolutions with zero-padding as in [17]. Distinctly, we use 3D convolutional layers with both spatial and temporal zero paddings to learn local motion information. The learnt encodings $p \in \mathbb{R}^{T \times C}$ are pooled on spatial dimension only by a spatial global average pooling (GAP). Note that while we use local 3D convolution to model motion, we have the position encoding per frame to correspond to the bounding box b.

As interactions focus on the temporal evolution of the relationships between hands and objects, we combine features and their positional encodings to form trajectories. We use standard summation but also ablate this against concatenation. For each hand, we represent the trajectory $H = (a^1 + p^1, \cdots, a^T + p^T)$ over T frames, and similarly for objects. We thus have four trajectories $\{H_L, H_R, O_L, O_R\}$, which form the input for our interaction unit.

Note that one or more of these frames might be missing detections – when hands or objects are not present in a frame or have not been detected. In this case the feature a is set to 0 and the binary map is blank. We showcase experimentally that our method can recover from missed detections. Similarly, some trajectories might be missing altogether. We consider the presence as well as the absence of hands and objects as informative evidence for interaction reasoning.

We next describe how we can reason about interactions using these trajectories.

3.3. Interaction Unit

In order to reason about HOIs, we consider a pair of trajectories (H, O) where H is the actor - a left/right hand and O is the object with which the hand interacts. Importantly O can be the other hand or an object. As Fig. 4 demonstrates, we have up to 3 interactions per actor/hand. For the left

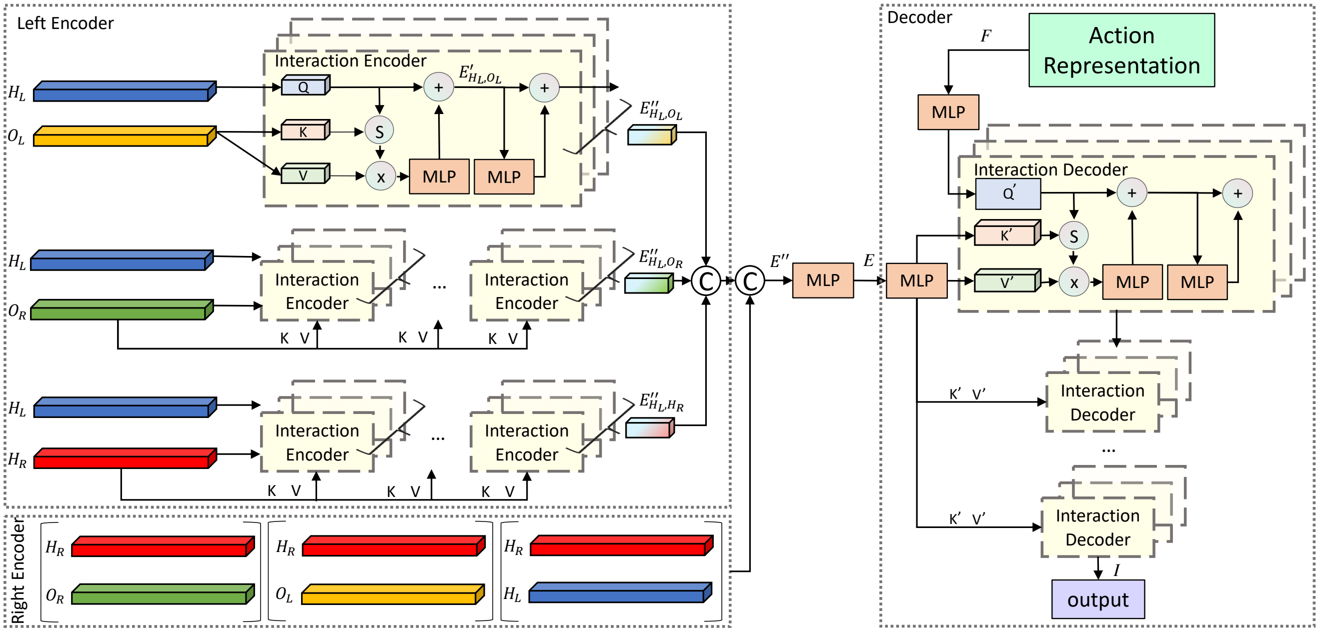


Figure 4. The Architecture of proposed Interaction Unit (IU). This module includes encoders and decoders. Encoders reason about hand-object interactions by multi-head transformers with hands as query and objects as key and value. Encoders outputs from left and right hands are concatenated and fed to decoders. Decoders model the interaction with action representation.

hand, these would be (H_L, O_L), (H_L, O_R) and (H_L, H_R). We intuitively describe what these capture using the example from Fig. 1 on stirring food in the pan.

- (H_L, O_L) captures the left hand holding the pan. The hand gesture and relative positions of hand and pan would be captured using this interaction.
- (H_L, O_R) captures the left hand versus the spoon as it stirs through the food in the pan. Note that the spoon is not directly interacting with the left hand.
- (H_L, H_R) captures the absolute positions and gestures of both hands as one holds the pan and the other stirs the food.

We first describe our interaction encoders. These are concatenated and their output is passed to stacked decoders described after.

Interaction Encoder. We introduce query Q, key K and value V which are projections by three different linear maps (q, k and v, respectively). We explain the encoder for one interaction pair namely (H_L, O_L).

$$\begin{aligned} Q_{H_L,O_L} &= q(H_L), \\ K_{H_L,O_L} &= k(O_L), \\ V_{H_L,O_L} &= v(O_L), \end{aligned} \tag{2}$$

where q, k, v all linearly project $\mathbb{R}^{T\times C}$ input to $\mathbb{R}^N$. Fig. 4 shows the module of left-hand interactions. The encoder is then a residual attention unit:

$$E^{'}_{H_L,O_L} = \sigma\left(\frac{Q_{H_L,O_L}K_{H_L,O_L}}{\sqrt{N}}\right)V_{H_L,O_L} + Q_{H_L,O_L}, \tag{3}$$

where $E^{'}_{H_L,O_L}$ is preliminary interaction representation between actor H_L and object O_L and σ is the softmax operator. Following [32], we also add a linear feedforward network $FFN()$ to reason about the interaction $E^{''}_{H_L,O_L}$ based on preliminary one $E^{'}_{H_L,O_L}$:

$$E^{''}_{H_L,O_L} = \delta(FFN(E^{'}_{H_L,O_L})) + E^{'}_{H_L,O_L}, \tag{4}$$

where δ is the dropout operation and $E^{''}_{H_L}, O_L \in \mathbb{R}^N$ is the encoded representation for (H_L, O_L).

Similarly, $E^{''}_{H_L,O_R}$ and $E^{''}_{H_L,H_R}$ are computed for the left hand (see Fig. 4) as well as three encoders for $E^{''}_{H_R}$. We concatenate all outputs to form overall encoding $\mathbf{E}^{''} \in \mathbb{R}^{6N}$. Subsequently, the dimension of $\mathbf{E}$ is reduced from 6N to M. We set M to the size of action representation F by a linear projection to simplify our decoder.

Interaction Decoders. Having reasoned about all pairwise interactions using the encoder, we use $\mathbf{E}$ to enrich the action representation, as in [13]. The pipeline of the decoder is similar to the encoder. Specifically, the features $\mathbf{E}$ from the encoder is projected to key $K^{'} \in \mathbb{R}^M$ and value $V^{'} \in \mathbb{R}^M$. We use the action representation features $F \in \mathbb{R}^M$ pooled on temporal dimension as query, and directly map F linearly to $Q^{'} \in \mathbb{R}^M$. Similarly, we adopt dropout, a

We keep the terms from [32] for consistency feedforward network as well as residual connections, like in Eq. 4, to learn decoder's output $I \in \mathbb{R}^M$.

At last, we stack the multi-head interaction encoders and decoders. The output I is fed to a fully connected layer, and trained to classify actions using standard cross entropy loss. During training, we backpropagate through all components of the IRN including IU and SPE, as well as the base network. The weights of the detector remain frozen.

4. Experiments and Results

In this section, we experimentally evaluate the model on two datasets featuring hand-object interactions from egocentric and crowd sourced videos respectively.

4.1. Datasets and Implementation Details

EPIC-KITCHENS-100 [6] is the largest video dataset in egocentric vision. We report on the full dataset but the conduct ablation study on a subset of the validation set. The subset was collected by the first participant (P01) and contains 5,509 / 885 action segments for the training and validation set, respectively. We use a fixed random seed and a single run during ablations to ensure results are directly comparable.

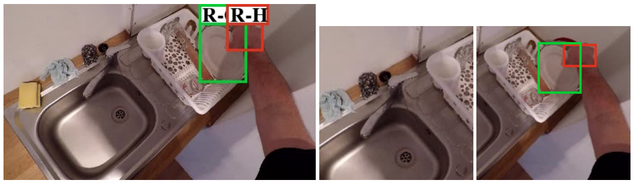
Something-Else [22] is based on [15] proposing a new split for novel verb-noun combinations in the test set. Active objects have been densely annotated for training by [22] – however hands are not annotated for side L/R and accordingly the active objects are not associated with a hand. We use this proposed split without the spatial annotations, instead using automatic, potentially noisy, detections.

Evaluation We use evaluation metrics proposed for both dataset - Top-1/5 accuracy for verb, noun and action classes. For Something-Else, we use a single output classifier.

Implementation Details. The hand-object detector [27] aims to find links between hands and interacting objects by optimising an offset vector. It was trained on 100 days dataset, also proposed in [27], that contains both first-person and third-person videos. We use the publicly available trained weights, and detect frame-level hands and objects with the confidence threshold of 0.5. Note, if hands or objects failed to be detected, we set these detections to 0.

We use SlowFast [10] R-50 8×8 due to its performance on both datasets. The input videos are 32-frame clips, where we sample $T = 8$ frames with a temporal stride $\tau = 8$ for the slow pathway, and 32 frames for the fast pathway. We pool hands and objects features $C = 640$ based on the third block of SlowFast as when the layer goes deeper, the features get sparser. We set $N = 5120$ used in the encoder and $M = 2304$ so that it matches the the globally pooled feature size of the last block of SlowFast. For the interaction unit, 16 heads are used in the multi-head attention and 3-layer encoders and decoders are implemented.

For EPIC-KITCHENS-100, the backbone is pre-trained on the training set and $e()$ and $d()$ are trained from random initialisation. We train for 24 epochs with learning rate 0.001 using SGD with 0.0001 weight decay and 0.9 momentum, the learning rate is decayed by the factor of 10 at epochs 10 and 20. For Something-Else the backbone is pretrained on Kinetics-400 [19]. Similarly, the network was optimised by SGD with initial learning rate 0.01 dropped at epochs 12 and 18, 0.0001 weight decay, 0.9 momentum for 20 epochs.



(a) input (b) STD (c) SCR

Figure 5. Visualisation of STD and SCR for data augmentation. The action is "take plate".

It is important to note that we changed the random cropping typically used for data augmentation. This is because randomly cropping the image in EPIC-KITCHENS-100 results in hands and objects being frequently cropped out of the main frame. As we show in Fig. 5, the standard STD crops the right hand and plate out. This significantly harms the interaction unit. In egocentric footage, hands can be in the bottom half of the image or towards a corner. To avoid this while maintaining the power of augmentation, we first randomly scale (**S**) frames to $H' \times W'$, then crop (**C**) frames to $H' \times H'$ and finally resize (**R**) frames to the target resolution 224 × 224. We refer to this augmentation approach as SCR, which we use in all our experiments on EPIC-KITCHENS-100. The two data augmentations perform comparably on Something-Else dataset as the action is typically at the centre of the image. We thus use the standard random cropping data augmentation for this dataset.

4.2. Something-Else Dataset Results

Tab. 1 shows the performance of our method outperforms SoTA reasoning approaches STIN, STRG and CAF. Specifically, we outperform STIN Combined with I3D, when trained jointly or separately. Notably, our class-agnostic IRN is superior to methods using object labels among non-ensemble models. We also report ensemble methods. These metods are not directly comparable but our model remains competitive.

Importantly, we showcase that our model particularly benefits from the usage of a SlowFast backbone. We evaluate our interaction units with different backbones in Tab. 2. We get the best performance when combining SlowFast as a base network with our interaction unit, but also report improvement when using the I3D backbone. In the Something-Else dataset, we find that only 39.63% of clips have both left and right hands, while 50.11% and 10.26% of clips have only one hand or no hands respectively. When

Method	Ens.	Obj.	Top-1	Top-5
STIN [22]	✗	✓	37.2	62.4
I3D+STIN [22]	✗	✓	48.2	72.6
CAF [25]	✗	✓	52.3	78.9
STRG [36]	✗	✗	52.3	78.3
IRN (Ours)	✗	✗	52.9	80.8
I3D-STIN [22]	✓	✓	51.5	77.1
STRG-STIN [22]	✓	✓	56.2	81.3
CACNF [25]	✓	✓	56.9	82.5

Table 1. Results on Something-Else Datasets. +: jointly trained. -: trained separately. Ens.: Ensemble. Obj.: use manual object labels.

I3D	SlowFast	IRN	Top-1	Top-5
✓			46.8	72.2
✓		✓	47.5	73.8
	✓		52.2	80.3
	✓	✓	52.9	80.8

Table 2. Results of different backbones on Something-Else.

no actor/hand is detected, our interaction unit is likely to struggle. Despite this, our proposed method can enrich the representation for interaction reasoning.

4.3. EPIC-KITCHENS-100 Dataset Results

As this dataset is egocentric, i.e. participants are using a wearable camera, more actions involve both hands making it more suitable to assess our proposal. To manage the size, we conduct an ablation on a selected subset.

Interaction Components. To evaluate the contribution of the various encoding interactions, we ablate the results by removing one at a time, as well as left/right hand encoders in Tab. 3. We first note that removing right hand interaction encoders (row 6) results in a larger drop than left hand (row 2). This is anticipated with most participants being right-handed. Similarly, the largest drop is associated with removing the encoder of H_R, O_R, which is critical for one-handed interactions (row 7) followed by H_R, H_L (row 9) which is critical for hand-only interactions (e.g. wash hands). The encoder with the least contribution is that for H_R, O_L (row 8), as it is probably compensated by the other pairwise encoders.

Trajectory. We evaluate the importance of using trajectories. Previous work [13] only uses one actor (person) of the middle frame ($middle$) along with video action representation and a recent work [24] detects persons in the middle frame and then duplicate these detections across time ($duplicate$). We compare these options to our proposed approach that encodes object trajectories as well as two-handed interactions. The best performance is achieved when using the complete trajectory of detections, including both hands and objects.

Spatial Positional Encoding. We adopt two ways to fuse positional information with visual features of hands or objects. As we expected in Tab. 5, spatial position encoding SPE improves the performance. Compared to the network without SPE ($none$ in Tab. 5) or concatenations of SPE. This is due to the sparsity of positional features. Concatenating lots of zeros to the visual features may introduce noise. Our proposed approach to sum the positional encoding yields the best action performance.

(H_L, O_L)	(H_L, O_R)	(H_L, H_R)	(H_R, O_R)	(H_R, O_L)	(H_R, H_L)	Top-1
✗	✗	✗	✗	✗	✗	42.37
✗	✗	✗	✓	✓	✓	43.28
✗	✓	✓	✓	✓	✓	42.60
✓	✗	✓	✓	✓	✓	43.28
✓	✓	✗	✓	✓	✓	42.94
✓	✓	✓	✗	✗	✗	42.82
✓	✓	✓	✗	✓	✓	41.47
✓	✓	✓	✓	✗	✓	44.07
✓	✓	✓	✓	✓	✗	41.69
✓	✓	✓	✓	✓	✓	**44.52**

Table 3. Ablation study for Interaction Components.

Det.	H_R	Act. Rep.	H_L	Objects	Top-1
middle [13]	✓	✓			43.05
duplicate [24]	✓	✓			44.07
trajectory	✓	✓			42.37
trajectory	✓	✓		✓	43.28
trajectory	✓	✓	✓	✓	**44.52**

Table 4. Comparison to prior works without trajectory representations as well as active object trajectories.

SPE	Top-1
none	43.39
concat	42.71
sum	**44.52**

Table 5. Ablation study for Spatial Position Encoding.

Act.Rep.	Top-1
none	31.98
concat	43.73
decoder	**44.52**

Table 6. Ablation study for Action Representation.

		Top-1 Accuracy (%)		
Method	DataAug.	Verb	Noun	Act.
Chance [6]	STD	10.42	1.70	0.51
IRN (Ours)	STD	60.94	43.97	31.97
TSN [6]	STD	60.18	46.03	33.19
SlowFast [6]	SCR	63.64	48.58	36.76
IRN (Ours)	SCR	**63.68**	**48.94**	**37.11**

Table 7. Quantitative results on full Validation set of EPIC-KITCHEN-100. STD and SCR denote standard and our data augmentation strategies.

Action Representation. In this ablation study, we evaluate the importance of globally pooled action representation. First, We remove decoders and train the network with only interaction features, i.e. $none$. Tab. 6 shows that results drop significantly if the network does not use the globally-pooled action representation. We also concatenate the hand-object interaction features with action representation, in a late-fusion fashion. What stands out in the Tab. 6 is that the Top-1 action result of the decoder surpasses concatenation.

We next report results on the full validation set of EPIC-KITCHENS-100. In Tab. 7, our method IRN improves over baselines. Adding the interaction unit to SlowFast improves results for verb, noun and action top-1 accuracy by 0.04%, 0.36% and 0.35%, respectively. Comparing row 2 and row 5, we demonstrate that the random cropping of

We'll make the subset available for direct comparisons

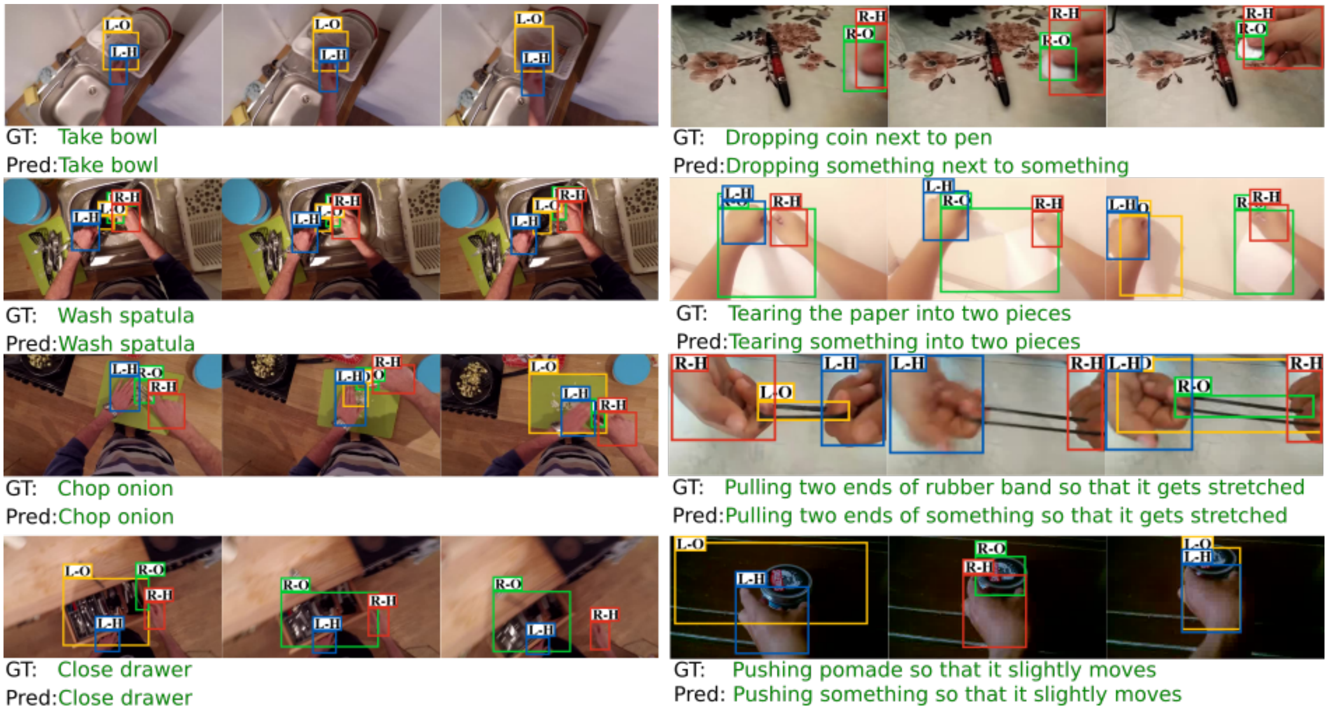

GT: Take bowl
Pred:Take bowl
GT: Dropping coin next to pen
Pred:Dropping something next to something
GT: Wash spatula
Pred:Wash spatula
GT: Tearing the paper into two pieces
Pred:Tearing something into two pieces
GT: Chop onion
Pred:Chop onion
GT: Pulling two ends of rubber band so that it gets stretched
Pred:Pulling two ends of something so that it gets stretched
GT: Close drawer
Pred:Close drawer
GT: Pushing pomade so that it slightly moves
Pred: Pushing something so that it slightly moves

Figure 6. Qualitative results of correctly-recognised interactions from EPIC-KITCHENS-100 (col 1) and Something-Else (col 2). L/R indicate the side and H/O denote hands and objects. GT and $Pred$ are Ground Truth and Prediction.

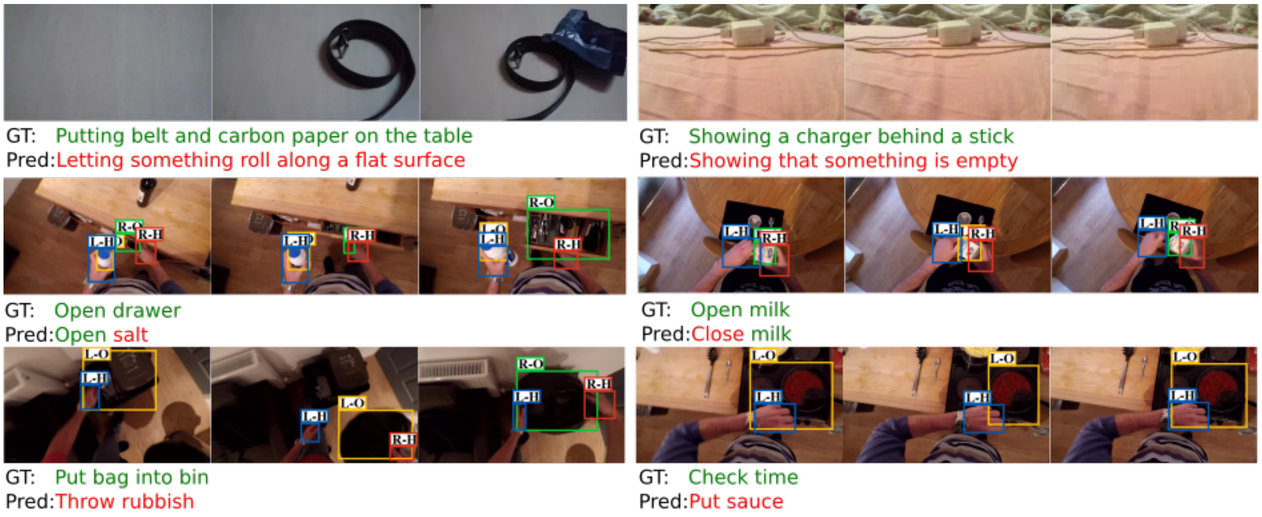

GT: Putting belt and carbon paper on the table
Pred:Letting something roll along a flat surface
GT: Showing a charger behind a stick
Pred:Showing that something is empty
GT: Open drawer
Pred:Open salt
GT: Open milk
Pred:Close milk
GT: Put bag into bin
Pred:Throw rubbish
GT: Check time
Pred:Put sauce

Figure 7. Failure cases of Something-Else (row 1) and EPIC-KITCHENS-100 (rows 2, 3).

STD harms the performance of IRN.

4.4. Qualitative Results

We demonstrate the results on clips containing only one hand or both hands in Fig. 6. It illustrates correct action recognition examples from both datasets. It is also important to highlight that our method recovers from partial failures in automatic detections of hands and objects, as in rows 3 and 4. IRN is robust to two types of detection errors: (a) incorrect active objects detection; (b) incorrect side of hand and objects. Specifically, row 3 illustrates “chop onion” and “pull two ends of something so that it gets stretched”. The onion is not detected in several frames due to occlusion. Similarly. the knife and rubber are missed in several frames along the trajectory. For row 4 (col 1) the drawer switches between being a left-object (O_L) and a right object (O_R), while in row 4 (col 2) the hand sides are swapped in various frames. Both errors (a) and (b) have little impact on our method, due to our usage of trajectories and the attention mechanism that selects the relevant hand

and object representations to reason about interactions.

Moreover, we show failure cases in Fig. 7. The main reason for failure is undetected or unobserved hands throughout the videos (row 1). In both examples, the hands are not visible throughout. Row 2 (col 1) shows that our focus on both hands might result in detecting another concurrent action, paying more attention to the object in the other hand (salt). In row 2 (col 2) the method fails to recognise the interaction with small movement. In row 3, the less frequent action of replacing the bin bag is incorrectly mistaken as the frequent throwing action. A clear limitation of our approach is evident in row 3 (col 2) where the person is checking the time on their hand watch. Evidently, the watch is never recognised as the interacting object, as it is always part of the hand detection.

5. Conclusion

In this paper, we present a framework for hand-object interaction reasoning that separately attends to actors (hands) and interacting objects, through encoders, as well as action representation which includes contextual information through a decoder. We present results on two hand-object interaction datasets, demonstrating generality and competitive performance, with an ablation study.

Acknowledgements. This work used public datasets and is supported by EPSRC UMPIRE (EP/T004991/1).

References

[1] F. Baradel et al. Object level visual reasoning in videos. In *ECCV*, 2018. 1, 2, 3

[2] G. Bertasius et al. Is space-time attention all you need for video understanding? In *ICML*, 2021. 2

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1, 2

[4] Y. Chao et al. Learning to detect human-object interactions. In *WACV*, 2018. 2

[5] D. Damen et al. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *ECCV*, 2018. 1

[6] D. Damen et al. Rescaling egocentric vision. *IJCV*, 2021. 1, 2, 3, 5, 6

[7] J. Donahue et al. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015. 2

[8] C. Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, June 2020. 2

[9] C. Feichtenhofer et al. Convolutional two-stream network fusion for video action recog. In *CVPR*, 2016. 2

[10] C. Feichtenhofer et al. Slowfast networks for video recognition. In *ICCV*, 2019. 2, 5

[11] C. Gao et al. ICAN: Instance-centric attention network for human-object interaction. In *BMVC*, 2018. 2

[12] C. Gao et al. DRG: Dual relation graph for human-object interaction detection. In *ECCV*, 2020. 2

[13] R. Girdhar et al. Video action transformer network. In *CVPR*, 2019. 1, 2, 3, 4, 6

[14] G. Gkioxari et al. Detecting and recognizing human-object interactions. In *CVPR*, 2018. 2, 3

[15] R. Goyal et al. The something something video database. In *ICCV*, 2017. 1, 5

[16] C. Gu et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 1, 2

[17] M. Islam et al. How much position information do convolutional neural networks encode? In *ICLR*, 2020. 3

[18] S. Ji et al. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2012. 2

[19] W. Kay et al. The kinetics human action video dataset. *arXiv*, 2017. 5

[20] C. Li et al. Collaborative spatiotemporal feature learning for video action recognition. In *CVPR*, 2019. 2

[21] Y. Li et al. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019. 2

[22] J. Materzynska et al. Something-Else: Compositional action recognition. In *CVPR*, 2020. 2, 5, 6

[23] D. Neimark et al. Video transformer network. *arXiv*, 2021. 2

[24] J. Pan et al. Actor-context-actor relation network for spatio-temporal action localization. 2020. 1, 2, 6

[25] G. Radevski et al. Revisiting spatio-temporal layouts for compositional action recognition. *arXiv*, 2021. 2, 6

[26] S. Ren et al. Faster R-CNN. In *NeurIPS*, 2015. 1, 2

[27] D. Shan et al. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 1, 2, 3, 5

[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recog. In *NeurIPS*, 2014. 2

[29] K. Soomro and A. Zamir. Action recognition in realistic sports videos. In *the Computer Vision in Sports*. 2014. 2

[30] C. Sun et al. Actor-centric relation network. In *ECCV*, 2018. 1

[31] D. Tran et al. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 2

[32] A. Vaswani et al. Attention is all you need. In *NeurIPS*, 2017. 4

[33] B. Wan et al. Pose-aware multi-level feature network for human object interaction detection. In *ICCV*, 2019. 2

[34] L. Wang et al. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016. 2

[35] X. Wang et al. Non-local neural networks. In *CVPR*, 2018. 2

[36] X. Wang and A. Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 2, 6

[37] C. Wu et al. Long-Term Feature Banks for Detailed Video Understanding. In *CVPR*, 2019. 2

[38] J. Yue-Hei Ng et al. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015. 2

[39] B. Zhou et al. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018. 2

[40] P. Zhou and M. Chi. Relation parsing neural network for human-object interaction detection. In *ICCV*, 2019. 2