Learning Object Permanence from Video

Aviv Shamsian*[1] Ofri Kleinfeld*[1] Amir Globerson[2] Gal Chechik[1,3]

[1] Bar-Ilan University, Ramat-Gan, Israel
[2] Tel Aviv University, Tel Aviv, Israel
[3] NVIDIA Research, Tel-Aviv, Israel

Abstract. *Object Permanence* allows people to reason about the location of non-visible objects, by understanding that they continue to exist even when not perceived directly. Object Permanence is critical for building a model of the world, since objects in natural visual scenes dynamically occlude and contain each-other. Intensive studies in developmental psychology suggest that object permanence is a challenging task that is learned through extensive experience.
Here we introduce the setup of learning Object Permanence from labeled videos. We explain why this learning problem should be dissected into four components, where objects are (1) visible, (2) occluded, (3) contained by another object and (4) carried by a containing object. The fourth subtask, where a target object is carried by a containing object, is particularly challenging because it requires a system to reason about a moving location of an invisible object. We then present a unified deep architecture that learns to predict object location under these four scenarios. We evaluate the architecture and system on a new dataset based on CATER, with per-frame labels, and find that it outperforms previous localization methods and various baselines.

Keywords: Object Permanence, Reasoning, Video Analysis

1 Introduction

Understanding dynamic natural scenes is often challenged by objects that contain or occlude each other. To reason correctly about such visual scenes, systems need to develop a sense of *Object Permanence* (OP) [20]. Namely, the understanding that objects continue to exist and preserve their physical characteristics, even if they are not perceived directly. For example, we want systems to learn that a pedestrian occluded by a truck may emerge from its other side, but that a person entering a car would "disappear" from the scene.

The concept of OP received substantial attention in the cognitive development literature. Piaget hypothesized that infants develop OP relatively late (at two years of age), suggesting that it is a challenging task that requires deep modelling of the world based on sensory-motor interaction with objects. Later evidence showed that children learn OP for occluded targets early [1,2]. Still,

*equal contribution

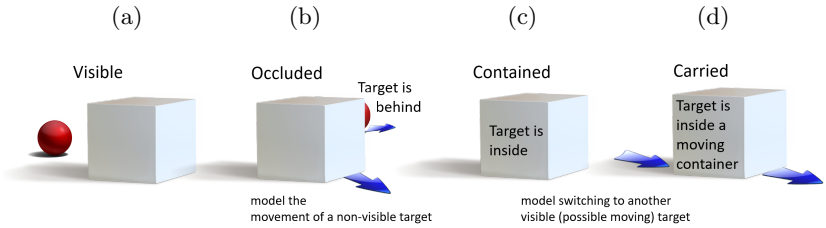

Fig. 1. Inferring object location in rich dynamic scenes involves four different tasks, and two different types of reasoning. (a) The target, a red ball, is fully visible. (b) The target is fully-or partially occluded by the static cube. (c) The target is located inside the cube and fully covered. (d) The non-visible target is located inside another moving object; its location changes even though it is not directly visible.

only at a later age do children develop understanding of objects that are contained by other objects [25]. Based on these experiments we hypothesize that reasoning about the location of non-visible objects may be much harder when they are carried inside other moving objects.

Reasoning about the location of a target object in a video scene involves four different subtasks of increasing complexity (Figure 1). These four tasks are based on the state of the target object, depending if it is (1) visible, (2) occluded, (3) contained or (4) carried. The *visible* case is perhaps the simplest task, and corresponds to object detection, where one aims to localize an object that is visible. Detection was studied extensively and is viewed as a key component in computer vision systems. The second task, *occlusion*, is to detect a target object which becomes transiently invisible by a moving occluding object (e.g., bicycle behind a truck). Tracking objects under occlusion can be very challenging, especially with long-term occlusions [11,9,18,14,4,30].

Third, in a *containment* scenario, a target object may be located inside another container object and become non-visible [28], for example when person enters a store. Finally, the fourth case of a *carried* object is arguably the most challenging task. It requires inferring the location of a non-visible object located inside a moving containing object (e.g., a person enters a taxi that leaves the scene). The task is particularly challenging because one has to keep a representation of which object should be tracked at every time point and to "switch states" dynamically through time. This task received little attention in the computer vision community so far.

We argue that reasoning about the location of a non-visible object should address two distinct and fundamentally different cases: occlusion and containment. First, to *localize an occluded object*, an agent has to build an internal state that models how the object moves. For example, when we observe a person walking in the street, we can predict her ever-changing location even if occluded by a large bus. In this mode, our reasoning mechanism keeps attending to the person and keeps inferring her location from past data. Second, *localizing contained objects*

is fundamentally different. It requires a reasoning mechanism that switches to attend to the containing object, which is visible. Here, even though the object of interest is not-visible, its location can be accurately inferred from the location of the visible containing object. We demonstrate below that incorporating these two reasoning mechanisms leads to more accurate localization in all four subtasks.

Specifically, we develop a unified approach for learning all four object localization subtasks in video. We design a deep architecture that learns to localize objects that may be visible, occluded, contained or carried. Our architecture consists of two reasoning modules designed to reason about (1) carried or contained targets, and (2) occluded or visible targets. The first reasoning component is explicitly designed to answer the question *"Which object should be tracked now?"*. It does so by using an LSTM to weight the perceived locations of the objects in the scene. The second reasoning component leverages the information about which object should be tracked and previous known locations of the target to localize the target, even if it is occluded. Finally, we also introduce a dataset called LA-CATER, based on videos from CATER [8] enriched with new annotations about task type and about ground-truth location of all objects.

Our main novel contributions are: (1) We conceptualize that localizing non-visible objects requires two types of reasoning: about occluded objects and about carried ones. (2) We define four subtypes of localization tasks and introduce annotations for the CATER dataset to facilitate evaluating each of these subtasks. (3) We describe a new unified architecture for all four subtasks, which can capture the two types of reasoning, and we show empirically that it outperforms multiple strong baselines. Our data and code are available for the community at our website [1]

2 Related Work

Relational Reasoning in Synthetic Video Datasets. Recently, several studies provided synthetic datasets to explore object interaction and reasoning. Many of these studies are based on CLEVR [12], a synthetic dataset designed for visual reasoning through visual question answering. CLEVRER [31] extended CLEVR to video, focusing on the causal structures underlying object interactions. It demonstrated that visual reasoning models that thrive on perception based tasks often perform poorly in causal reasoning tasks.

Most relevant for our paper, CATER [8] is a dataset for reasoning about object action and interactions in video. One of the three tasks defined in CATER, the *snitch localization* task, is closely related to the OP problem studied here. It is defined as localizing a target *at the end of a video*, where the target is usually visible. Our work refines their setup, learning to localize the target through the full video, and breaks down prediction into four types of localization tasks. As a result, we provide a fine-grained insight about the architectures and reasoning that is required for solving the complex localization task.

[1] https://chechiklab.biu.ac.il/~avivshamsian/OP/OP_HTML.html

Architectures for Video Reasoning. Several recent papers studied the effectiveness of CNN-based architectures for video action recognition. Many approaches use 3D convolutions for spatiotemporal feature learning [3,27] and separate the spatial and temporal modalities by adding optical flow as a second stream [6,24]. These models are computationally expensive because 3D convolution kernels may be costly to compute. As a result, they may limited the sequence length to 20-30 frames [3,27]. In [34] it was proposed to sparsely sample video frames to capture temporal relations in action recognition datasets. However, sparse sampling may be insufficient for long occlusion and containment sequences, which is the core of our OP focus.

Another strategy for temporal aggregation is to use recurrent architectures like LSTM [10], connecting the underlying CNN output along the temporal dimension [32]. [7,26,23] combined LSTM with spatial attention, learning to attend to those parts of the video frame that are relevant for the task as the video progresses. In Section 6 we experiment with a spatial attention module, which learns to dynamically focus on relevant objects.

Tracking with Object Occlusion. A large body of work has been devoted to tracking objects [18]. For objects under complex occlusion like carrying, early work studied tracking using classical techniques and without deep learning methods. For instance, [11,19] used the idea of object permanence to track objects through long-term occlusions. They located objects using adaptive appearance models, modelling spatial distributions and inter-occlusion relationships. In contrast, the approach presented in this paper focuses on a single deep differentiable model to learn motion reasoning end-to-end. [9] succeeds to track occluded targets by learning how their movement is coupled with the movement of other visible objects. The dataset studied here, CATER [8], has weak object-object motion coupling by design. Specifically, when measuring the correlation between the movement of the target and other object (as in [9]), we found that the correlation in 94% of the videos was not statistically significant.

More recently, models based on Siamese neural network achieved SOTA results in object tracking [5,15,35]. Despite the power of these architectures, tracking highly-occluded objects is still challenging [18]. The tracker of [35], DaSiamRPN, extends the region-proposal sub-network of [15]. It was designed for long-term tracking and handles full occlusion or out-of-view scenarios. DaSiamRPN was used as a baseline for the snitch localization task in CATER [8], and we evaluated its performance for the OP problem in Section 6.

Containment. Few recent studies explored the idea of containment relations. [16] recovered incomplete object trajectories by reasoning about containment relations. [28] proposed an unsupervised model to categorize spatial relations, including containment between objects. The containment setup defined in these studies differs from the one defined here in that the contained object is always at least partially visible [28], or the containment does not involve carrying [16,28].

3 The Learning Setup: Reason about Non-visible Objects

We next formally define the OP task and learning setup. We are given a set of videos $v_1, ..., v_N$ where each frame x_t^i in video v_i is accompanied by the bounding box position B_t^i of the target object as its label. The goal is to predict for each frame a bounding box $\hat{B}_t^i$ of the target object that is closest (in terms of L_1 distance) to the ground-truth bounding box B_t^i.

We define four localization tasks: (1) Localizing a visible object, which we define as an object that is at least partially visible. (2) Localizing an occluded object, which we define as an object that is *fully* occluded by another object. (3) Localizing an object contained by another object, thus also completely non visible. (4) Localizing an object that is carried along the surface by a containing object. Thus in this case the target is moving while being completely non-visible. Together, these four tasks form a localization task that we call object-permanence localization task, or OP.

In Section 7.2, we also study a semi-supervised learning setup, where at training time the location B_t^i of the target is provided only in frames when it is visible. This would correspond to the case of a child learning object permanence without explicit feedback about where an object is located when it is hidden.

It is instructive to note how the task we address here differs from the tasks of relation or action recognition [13,17,22]. In these tasks, models are trained to output an explicit label that captures the name of the interaction or relation (e.g., "behind", "carry"). In our task, the model aims to predict the location of the target (a regression problem), but it is not trained to name it explicitly (occluded, contained). While it is possible that the model creates some implicit representation describing the visibility type, this is not mandated by the loss or the architecture.

4 Our Approach

We describe a deep network architecture designed to address the four localization subtasks of the OP task. We refer to the architecture as OPNet. It contains three modules, that account for the perception and inference computations which facilitate OP (see Figure 2).

Perception and detection module (Figure 2a): A perception module, responsible for detecting and tracking visible objects. We incorporated a Faster R-CNN [21] object detection model, fine-tuned on frames from our dataset, as the perception component of our model. After pre-training, we used the detector to output the bounding boxes together with identifiers of all objects in any given frame. Specifically, we represent a frame using a $K \times 5$ matrix. Each row in the matrix represents an object using 5 values: four values of the bounding box (x_1, y_1, x_2, y_2) and one *visibility bit*, which indicates whether the object is visible or not. As the video progresses, we assign a unique row to each *newly identified* object. If an object is not detected in a given frame, its corresponding information (assigned row) is set to zero. In practice, $K = 15$ was the maximal number

of objects in a single video in our dataset. Notably, the videos in the dataset we used do not contain two identical objects, but we found that the detector sometimes mistakes one object for another. Objects in a video form an unordered collection [33]. To increase learning efficiency in this settings, we canonicalize the representation and keep the target as the first item of the set.

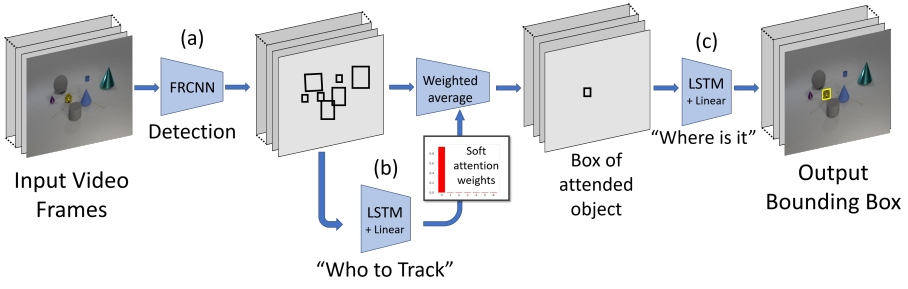

Fig. 2. The architecture of the *Object-Permanence network* (OPNet) consists of three components. (a) A perception module for detection. (b) A reasoning module for inferring which object to track when the target is carried or contained. (c) A second reasoning module for occluded or visible targets, to refine the location of the predicted target.

"Who to track?" module (Figure 2c): responsible for understanding which object is currently covering the target. This component consists of a single LSTM layer with a hidden dimension of 256 neurons and a linear projection matrix. After applying the LSTM to the object bounding boxes, we project its output to K neurons, each representing a distinct object in the frame. Finally, we apply a softmax layer, resulting in a distribution over the objects in the frame. This distribution can be viewed as an attention mask focusing on the object that covers the target in this frame. Importantly, we do not provide explicit supervision to this attention mask (e.g., by explicitly "telling the model" during training what is the correct attention mask). Rather, our only objective is the location of the target. The output of this module is 5 numbers per frame. It is computed as the the weighted average over the $K \times 5$ outputs of the previous stage, weighted by the attention mask.

"Where is it" module (Figure 2b): learns to predict the location of occluded targets. This final component consists of a second LSTM and a projection matrix. Using the output of the previous component, this component is responsible for predicting the target localization. It takes the output of the previous step (5 values per frame), feeds it into the LSTM and projects its output to four units, representing the predicted bounding box of the target for each frame.

5 The LA-CATER Dataset

To train models and evaluate their performance on the four OP subtasks defined above, we introduce a new set of annotations to the CATER dataset [8]. We refer to these as *Localization Annotations* (LA-CATER).

The CATER dataset consists of 5,500 videos generated programmatically using the Blender 3D engine. Each video is 10-second long (300 frames) and contains 5 to 10 objects. Each object is characterized by its shape (cube, sphere, cylinder and inverted cone), size (small, medium, large), material (shiny metal and matte rubber) and color (eight colors). Every video contains a golden small sphere referred to as "the snitch", that is used as the target object which needs to be localized.

For the purpose of this study, we generated videos following a similar configuration to the one used by CATER, but we computed additional annotations during video generation. Specifically, we augmented the CATER dataset with ground-truth bounding boxes locations of all objects. These annotations were programmatically extracted from the Blender engine, by projecting the internal 3D coordinates of objects are to the 2D pixel space.

We further annotated videos with detailed frame-level annotations. Each frame was labeled with one of four classes: visible, fully occluded, contained (i.e., covered, static and non-visible) and carried (i.e., covered, moving and non-visible). This classification of frames matches the four localization subtasks of the OP problem.

LA-CATER includes a total number of 14K videos split into train, dev and test datasets. See Table 1 for a classification of video frames to each one of the localization subtasks across the dataset splits. Further details about dataset preparation are provided in the supplementary.

Table 1. Fraction of frames per type in the train, dev and test sets of LA-CATER. Occluded and carried target frames make up less than 8% of the frames, but they present the most challenging prediction tasks.

	Number of Samples	Visible	Occluded	Contained	Carried
Train	9,300	63.00%	3.03%	29.43%	4.54%
Dev	3,327	63.27%	2.89%	29.19%	4.65%
Test	1,371	64.13%	3.07%	28.56%	4.24%

6 Experiments

We describe our experimental setup, compared methods and evaluation metrics. Implementation details are given in the supplementary

6.1 Baselines and Model Variants

We compare our proposed OPNet with six other architectures designed to solve the OP tasks. Since we are not aware of previously published unified architectures designed to solve all OP tasks at once, we used existing models as components in our baselines. All baseline models receive the predictions of the object detector (perception) component as their input.

(A) Programmed Models. We evaluated two models that are "hard-coded" rather than learned. They are designed to evaluate the power of models that programmatically solve the reasoning task.

- (1) *Detector + Tracker.* Using the detected location of the target, this method initiates a DaSiamRPN tracker [35] to track the target. Whenever the target is no longer visible, the tracker is re-initiated to track the object located in the last known location of the target.
- (2) *Detector + Heuristic.* When the target is not detected, the model switches from tracking the target to tracking the object located closest to last known location of the target. The model also employs an heuristic logic to adjust between the sizes of the current tracked object and the original target.

(B) Learned Models. We evaluated four learned baselines with an increasing level of representation complexity.

- (3) *OPNet.* The proposed model, as presented in Section 4.
- (4) *Baseline LSTM.* This model uses a single unidirectional LSTM layer with a hidden state of 512 neurons, operating on the temporal (frames) dimension. The input to the LSTM is the concatenation of the objects input representations. It is the simplest learned baseline as the input representation is not transformed non-linearly before being fed to the LSTM.
- (5) *Non-Linear + LSTM.* This model augments the previous model and increases the complexity of the scene representation. The input representations are upsampled using a linear layer followed by a ReLU activation, resulting in a 256-dimensional vector representation for each object in the frame. These high-dimensional objects representations are concatenated and fed into the LSTM.
- (6) *Transformer + LSTM.* This model augments the previous baselines by introducing a much complex representations for objects in frame. We utilized a transformer encoder [29] after up-sampling the input representations, employing self attention between all the objects in a frame. We used a transformer encoder with 2 layers and 2 attention heads, yielding a single vector containing the target attended values. These attended values, which corresponds to each other object in the frame, are then fed into the LSTM.
- (7) *LSTM + MLP.* This model (*Figure 2*) ablates the second LSTM module (c) in the model presented in Section 4.

6.2 Evaluation Metric

We evaluate model performance at a given frame t by comparing the predicted target localization and the ground truth (GT) target localization. We use two metrics as follows. First, intersection over union (IoU),

$$IoU_t = \frac{B_t^{GT} \cap B_t^p}{B_t^{GT} \cup B_t^p} \quad , \tag{1}$$

where B_t^p denotes the predicted bounding box for frame t and B_t^{GT} denotes the ground truth bounding box for frame t. Second, we evaluate models using their mean average precision (MAP). MAP is computed by employing an indicator function to each frame, determining whether the IoU value is greater than a predefined threshold, then averaging across frames in a single video and all the videos in the dataset.

$$AP = \frac{1}{n} \sum_{t=1}^{n} \mathbf{1_t} \text{ , where } \mathbf{1_t} = \begin{cases} 1 & IoU_t > IoU\ threshold \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$MAP = \frac{1}{N} \sum_{v=1}^{N} AP_v \quad . \tag{3}$$

These per-frame metrics allow us to quantify the performance on each of the four OP subtasks separately.

7 Results

We start with comparing OPNet with the baselines presented in Section 6.1. We then provide more insights into the performance of the models by repeating the evaluations with *"Perfect Perception"* in Section 7.1. Section 7.3 describes a semi-supervised setting of training with visible frames only. Finally, in Section 7.3 we compare OPNet with the models presented in the CATER paper on the original CATER data. We first compare OPNet and the baselines presented in Section 6.1. Table 2 shows IoU for all models in all four sub-tasks and Figure 3 presents the MAP accuracy of the models across different IoU thresholds.

It can be seen in Table 2 that OPNet performs consistently well across all subtasks and outperforms all other models overall. On the visible and occluded frames performance is similar to other baselines. But on the contained and carried frames, OPNet is significantly better than the other methods. This is likely due to OPNet's explicit modeling of the object to be tracked.

Table 2 also reports results for two variants of OPNet: OPNet (LSTM+MLP) and OPNet (LSTM+LSTM). The former is missing the second module ("Where is it" in Figure 3) which is meant to handle occlusion, and indeed under-performs for occlusion frames (the "occluded" and "contained" subtasks). This highlights the importance of using the two LSTM modules in Figure 3.

Table 2. Mean IoU on LA-CATER test data. "±" denotes the standard error of the mean (SEM). OPNet performs consistently well across all subtasks, and is significantly better on *contained* and *carried*

Mean IoU± SEM	Visible	Occluded	Contained	Carried	Overall
Detector + Tracker	90.27 ±0.13	53.62 ±0.58	39.98 ±0.38	34.45 ±0.40	71.23 ±0.51
Detector + Heuristic	90.06 ±0.14	47.03 ±0.73	55.36 ±0.53	55.87 ±0.59	76.91 ±0.43
Baseline LSTM	81.60 ±0.19	59.80 ±0.61	49.18 ±0.64	21.53 ±0.40	67.20 ±0.53
Non-Linear + LSTM	88.25 ±0.14	70.14 ±0.62	55.66 ±0.67	24.58 ±0.44	73.53 ±0.51
Transformer + LSTM	**90.82** ±0.14	**80.40** ±0.61	70.71 ±0.78	28.25 ±0.45	80.27 ±0.50
OPNet (LSTM + MLP)	88.11 ±0.16	55.32 ±0.85	65.18 ±0.89	**57.59** ±0.85	78.85 ±0.52
OPNet (LSTM + LSTM)	88.89 ±0.16	78.83 ±0.56	**76.79** ±0.62	56.04 ±0.77	**81.94** ±0.41

Figure 3 provides interesting insight into the behavior of the programmed models, namely, Detector + Tracker and Detector + Heuristic. These models perform well when the IoU threshold is low. This reflects the fact that they have a good coarse estimate of where the target is, but fail to provide more accurate localization. At the same time, OPNet performs well for accurate localization, presumably due to its learned "Where is it" module.

7.1 Reasoning with Perfect Perception

The OPNet model contains an initial "Perception" module that analyzes the frame pixels to get bounding boxes. Errors in this component will naturally propagate to the rest of the model and adversely affect results. Here we analyze the effect of the perception module by replacing it with ground truth bounding boxes and visibility bits. See supplementary for details on extracting ground-truth annotations. In this setup all errors reflect failure in the reasoning components of the models.

Table 3 provides the IoU performance and Figure 7 the MAP for all compared methods on all four subtasks. The results are similar to the previous results. When compared to the previous section (imperfect, detector-based, perception), the overall trend is the same, but all models improve when using the ground truth perception information. Interestingly, the subtask that improves the most from using ground truth boxes is the carried task. This makes sense, since it is the hardest subtask and the one that most relies on having the correct object locations per frame.

7.2 Learning only from Visible Frames

We now examine a learning setup in which localization supervision is available only for frames where the target object is visible. This setup corresponds more naturally to the process by which people learn object permanence. For instance, imagine a child learning to track a carried (non visible) object for the first time and receiving a surprising feedback only when the object reappears in the scene.

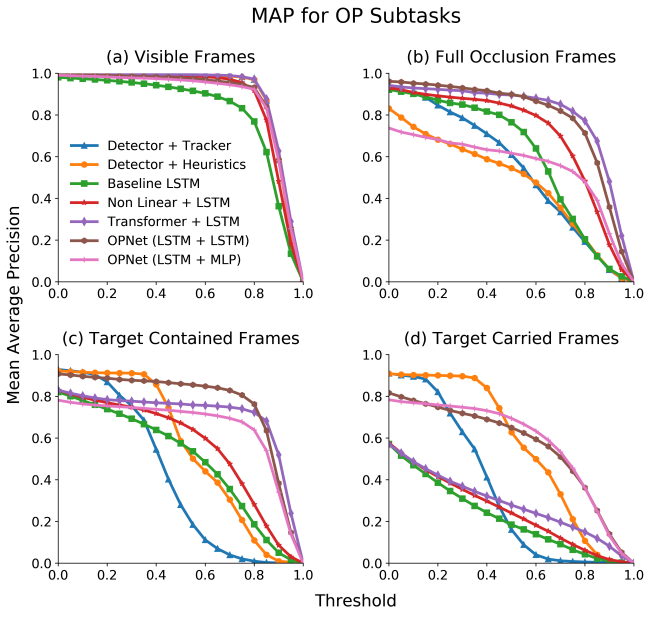

Fig. 3. Mean average precision (MAP) as a function of IoU thresholds. The two programmed models, Detector+Tracker (blue) and Detector+Heuristic (orange) perform well when the IoU threshold is low, providing a good coarse estimate of target location. OPNet performs well on all subtasks.

Table 3. Mean IoU with *Perfect Perception*. "±" denotes the standard error of the mean (S.E.M.). Results are similar in nature to those with imperfect, detector-based, perception (Table 2). All models improve when using ground-truth perception. The largest improvement due to OPNet is observed in the carried task.

Mean IoU ± SEM	Visible	Occluded	Contained	Carried	Overall
DETECTOR + TRACKER	90.27 ±0.13	53.62 ±0.58	39.98 ±0.38	34.45 ±0.40	71.23 ±0.51
DETECTOR + HEURISTIC	**95.59** ±0.34	30.40 ±0.81	59.81 ±0.47	59.33 ±0.50	81.24 ±0.49
BASELINE LSTM	75.22 ±0.31	50.52 ±0.75	45.10 ±0.62	19.12 ±0.36	61.41 ±0.53
NON-LINEAR + LSTM	88.63 ±0.25	65.73 ±0.82	58.77 ±0.70	23.89 ±0.41	74.53 ±0.54
TRANSFORMER + LSTM	93.99 ±0.24	**81.31** ±0.88	75.75 ±0.85	28.01 ±0.44	83.78 ±0.55
OPNET (LSTM + MLP)	88.11 ±0.16	19.39 ±0.60	77.40 ±0.68	**78.25** ±0.65	83.84 ±0.48
OPNET (LSTM + LSTM)	88.78 ±0.25	67.79 ±0.69	**83.47** ±0.47	76.42 ±0.66	**85.44** ±0.38

In absence of any supervision when the target is non-visible, incorporating an extra auxiliary loss is needed to account for these frames. Towards this end, we incorporated an auxiliary *consistency loss* that minimizes the change between predictions in consecutive frames. $\mathcal{L}_{consistency} = \frac{1}{n}\sum_{t=1}^{n} \|b_t - b_{t-1}\|^2$. The total

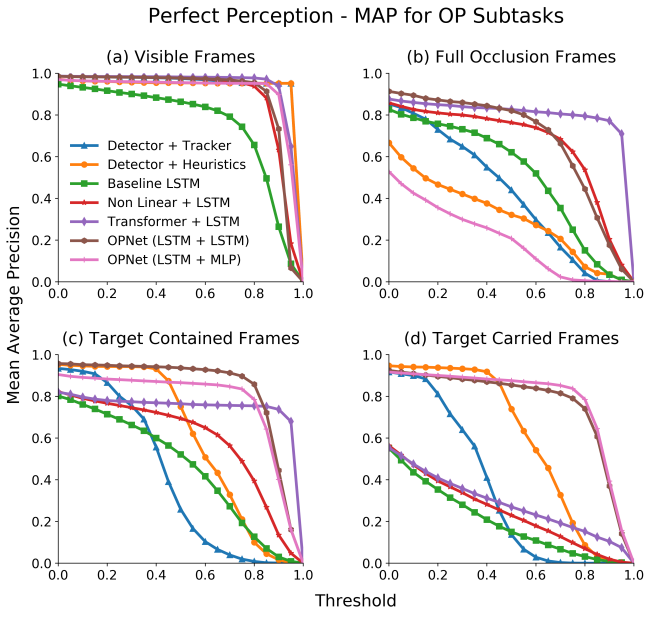

Fig. 4. Mean average precision (MAP) as a function of IoU thresholds for reasoning with Perfect Perception (Section 7.1). The most notable performance gain of OPNet (pink and brown curves) was with carried targets (subtask d).

loss is defined as an interpolation between the localization loss and the consistency loss, balancing their different scales: $\mathcal{L} = \alpha \cdot \mathcal{L}_{localization} + \beta \cdot \mathcal{L}_{consistency}$ Details on choosing the values of α and β are provided in the supplementary.

Table 4 shows the mean IoU for this setup (compare with Table 2). The baselines perform well when the target is visible, fully occluded or contained without movement. This phenomenon goes hand-in-hand with the inductive bias of the *consistency loss*. Usually, to solve these subtasks, a model only needs to predict

Table 4. IoU for learning with *only visible supervision*. "±" denotes the standard error of the mean (S.E.M.). The models do not perform well when the target is carried.

Mean IoU	Visible	Occluded	Contained	Carried	Overall
Baseline LSTM	88.61 ±0.16	80.39 ±0.54	68.35 ±0.76	27.39 ±0.45	78.09 ±0.49
Non Linear + LSTM	89.30 ±0.15	82.49 ±0.45	67.25 ±0.75	27.34 ±0.45	78.15 ±0.49
Transformer + LSTM	88.33 ±0.15	**83.74** ±0.44	**69.93** ±0.77	**27.65** ±0.54	78.43 ±0.49
OPNet (LSTM + MLP)	88.45 ±0.17	48.03 ±0.82	10.95 ±0.51	7.28 ±0.30	61.18 ±0.69
OPNet (LSTM + LSTM)	**88.95** ±0.16	81.84 ±0.48	69.01 ±0.76	27.50 ±0.45	**78.50** ±0.49

the last known target location. This explains why the OPNet (LSTM+MLP) model performs so poorly in this setup.

We note that the performance of non-OPNet models on the carried task is similar to that obtained using full supervision (see Table 2, Section 7) . This suggests that these models fail to use the supervision for the "carried" task, and further reinforces the observation that localizing carried object is highly challenging.

7.3 Comparison with CATER Data

The original CATER paper [8] considered the "snitch localization" task, aiming to localize the snitch at the last frame of the video, and formalized as a classification problem. The x-y plane was divided with a 6-by-6 grid, and the goal was to predict the correct cell of that grid.

Here we report the performance of OPNet and relevant baselines evaluated with the exact setup of [8], to facilitate comparison with the results reported there. Table 5 shows the accuracy and L_1-distance metrics for this evaluation. OPNet significantly improves over all baselines from [8]. It reduces classification error from 40% to 24%, and the L_1 distance from 1.2 to 0.54.

7.4 Qualitative Examples

To gain insight into the behaviour and limitations of the OPNet model, we provide examples of its successes and failures. The first video [1] provides a "win" example, demonstrating the power of the *"who-to-track"* reasoning component. In this example, the model correctly handles recursive containment that involve "carrying". See Figure 5 (top row). The second video [2] illustrates a failure, where OPNet fails to switch between tracked objects when the target is "carried". The model accidentally switches to an incorrect cone object (the yellow cone) that already contains another object, not the target. Interestingly, OPNet operates properly when the yellow cone is picked up and switches to track the blue ball that was contained by the yellow cone. It suggests that OPNet learns an implicit representation of the object actions (pick-up, slide, contain etc.) even though it was not explicitly trained to do so. See Figure 5 (bottom row).

Table 5. Classification accuracy on CATER using the metrics of [8].

Model	Accuracy (higher is better)	L_1 Distance (lower is better)
DaSiamRPN	33.9	2.4
TSN-RGB + LSTM	25.6	2.6
R3D + LSTM	60.2	1.2
OPNet (Ours)	**74.8**	**0.54**

[1] *https://youtu.be/FnturB2Blw8*
[2] *https://youtu.be/qkdQSHLrGqI*

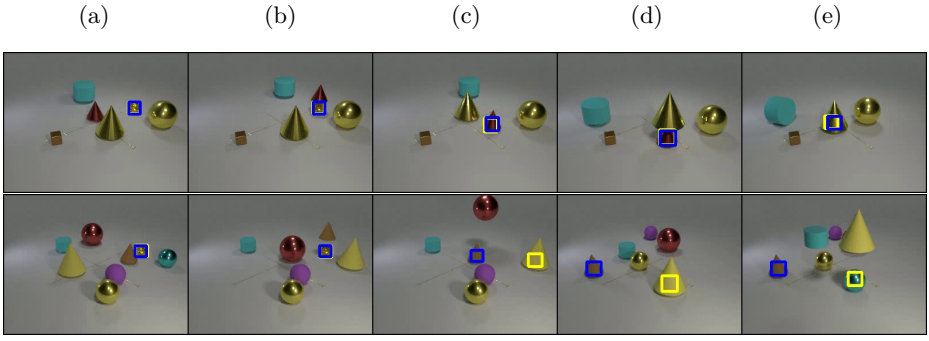

Fig. 5. Examples of a success case (top row) and a failure case (bottom row) for localizing a carried object. The blue box marks the ground-truth location. The yellow box marks the predicted location. *Top* (a) The target object is visible; (b-c) The target becomes covered and carried by the orange cone; (d-e) The big golden cone covers and carries the orange cone, illustrating recursive containment. The target object is not visible, but OPNet successfully tracks it. *Bottom* (c-d) OPNet accidentally switches to the wrong cone object (the yellow cone instead of the brown cone); (e) OPNet correctly finds when the yellow cone is picked up and switches to track the blue ball underneath.

8 Conclusion

We considered the problem of localizing one target object in highly dynamic scenes, where the target can be occluded, contained or even carried away, concealed by another object. We name this task *object permanence*, following the cognitive concept of an object that is physically present in a scene but is occluded or carried. We presented an architecture called OPNet, whose components naturally correspond to the perceptual and the reasoning stages of solving OP. Specifically, it has a module that learns to switch attention to an object if it infers that the object contains or carries the target. Our empirical evaluation shows that these components are needed for improving accuracy in this task.

Our results highlight a remaining gap between perfect perception and a pixel-based detector. It is expected that this gap may be even wider when applying OP to more complex natural videos in an open-world setting. It will be interesting to further improve detection architectures to reduce this gap

Acknowledgments

This study was funded by grants to GC from the Israel Science Foundation and Bar-Ilan University (ISF 737/2018, ISF 2332/18). AS is funded by the Israeli innovation authority through the AVATAR consortium. AG received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (grant ERC HOLI 819080).

References

1. Aguiar, A., Baillargeon, R.: 2.5-month-old infants' reasoning about when objects should and should not be occluded. Cognitive psychology **39**(2), 116–157 (1999)
2. Baillargeon, R., DeVos, J.: Object permanence in young infants: Further evidence. Child development **62**(6), 1227–1246 (1991)
3. 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)
4. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5374–5383 (2019)
5. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2019)
6. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal residual networks for video action recognition. corr abs/1611.02155 (2016). arXiv preprint arXiv:1611.02155 (2016)
7. Gao, L., Guo, Z., Zhang, H., Xu, X., Shen, H.T.: Video captioning with attention-based lstm and semantic consistency. IEEE Transactions on Multimedia **19**(9), 2045–2055 (2017)
8. Girdhar, R., Ramanan, D.: Cater: A diagnostic dataset for compositional actions and temporal reasoning. arXiv preprint arXiv:1910.04744 (2019)
9. Grabner, H., Matas, J., Van Gool, L., Cattin, P.: Tracking the invisible: Learning where the object might be. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1285–1292. IEEE (2010)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
11. Huang, Y., Essa, I.: Tracking multiple objects through occlusions. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 1051–1058. IEEE (2005)
12. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2901–2910 (2017)
13. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1), 32–73 (2017)
14. Kristan, M., Leonardis, A., et al., J.M.: The sixth visual object tracking vot2018 challenge results. In: ECCV Workshops (2018)
15. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8971–8980 (2018)
16. Liang, W., Zhu, Y., Zhu, S.C.: Tracking occluded objects and recovering incomplete trajectories by reasoning about containment relations and human actions. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
17. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: European conference on computer vision. pp. 852–869. Springer (2016)

18. Mojtaba Marvasti-Zadeh, S., Cheng, L., Ghanei-Yakhdan, H., Kasaei, S.: Deep learning for visual tracking: A comprehensive survey. arXiv pp. arXiv–1912 (2019)
19. Papadourakis, V., Argyros, A.: Multiple objects tracking in the presence of long-term occlusions. Computer Vision and Image Understanding **114**(7), 835–846 (2010)
20. Piaget, J.: The construction of reality in the child (1954)
21. 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)
22. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: CVPR 2011. pp. 1745–1752. IEEE (2011)
23. Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. arXiv preprint arXiv:1511.04119 (2015)
24. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
25. Smitsman, A.W., Dejonckheere, P.J., De Wit, T.C.: The significance of event information for 6-to 16-month-old infants' perception of containment. Developmental psychology **45**(1), 207 (2009)
26. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: Thirty-first AAAI conference on artificial intelligence (2017)
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. Ullman, S., Dorfman, N., Harari, D.: A model for discovering 'containment'relations. Cognition **183**, 67–81 (2019)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
30. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015)
31. Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., Tenenbaum, J.B.: Clevrer: Collision events for video representation and reasoning (2019)
32. 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)
33. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: Advances in neural information processing systems. pp. 3391–3401 (2017)
34. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. European Conference on Computer Vision (2018)
35. Zhu, Z., Wang, Q., Bo, L., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: European Conference on Computer Vision (2018)