Detect Active Object in a Sequential Voting Process

Qichen Fu

CMU-RI-TR-22-40

July 19, 2022

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Kris M. Kitani, *chair*
Prof. Abhinav Gupta
Yufei Ye

*Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.*

To my two years at CMU.

Abstract

A key component of understanding hand-object interactions is the ability to identify the active object – the object that is being manipulated by the human hand. In order to accurately localize the active object, any method must reason using information encoded by each image pixel, such as whether it belongs to the hand, the object, or the background. To leverage each pixel as evidence to determine the bounding box of the active object, we propose a pixel-wise voting function. Our pixel-wise voting function takes an initial bounding box as input and produces an improved bounding box of the active object as output. The voting function is designed so that each pixel inside of the input bounding box votes for an improved bounding box, and the box with the majority vote is selected as the output. We call the collection of bounding boxes generated inside of the voting function, the Relational Box Field, as it characterizes a field of bounding boxes defined in relationship to the current bounding box. While our voting function is able to improve the bounding box of the active object, one round of voting is typically not enough to accurately localize the active object. Therefore, we repeatedly apply the voting function to sequentially improve the location of the bounding box. However, since it is known that repeatedly applying a one-step predictor (i.e., auto-regressive processing with our voting function) can cause a data distribution shift, we mitigate this issue using reinforcement learning (RL). We adopt standard RL to learn the voting function parameters and show that it provides a meaningful improvement over a standard supervised learning approach. We perform experiments on two large-scale datasets: 100DOH and MECCANO, improving AP50 performance by 8% and 30%, respectively, over the state of the art.

Acknowledgments

I would like to express my sincere appreciation to my advisor, Prof. Kris Kitani, for his enormous advice and support for my Master's study and research. It has been a great pleasure to work in KLab on the EGO4D project and explore new topics related to egocentric videos. I could not have undertaken this wonderful journey without his dedicated guidance and abundant resources offered at KLab.

Besides, I am extremely grateful to my close collaborators: Xingyu, Sean, Take, Jack, Xuhua, and Xindi, for the stimulating discussion, editing help, and late-night feedback. Thanks should also get to my great labmates: all the members of KLab, for their generous help and advice. In KLab, I discovered diverse and exciting ideas from the weekly group meeting, learned valuable lessons from one-to-one chats, and had great fun during retreats and activities. I would also like to thank my thesis committee, Prof. Kris Kitani, Prof. Abhinav Gupta, and Yufei Ye for generously offering their time and feedback on my work.

Finally, I would like to thank my parents and Liyun for their unconditional love and support. Their trust allows me to explore my interests freely throughout my life.

Funding

x

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Finding the active object – the object that is being manipulated by the human hand – is a crucial task towards understanding human-object interactions, especially in egocentric videos where hands are the only visible human parts. It is also an essential step to a variety of downstream tasks including joint hand-object pose estimation [7, 8, 19, 26, 34], reconstruction [6, 15], activity recognition [11, 21, 23], and imitation learning [35]. However, an accurate localization of the active object can be challenging due to natural occlusions caused by the hands during interactions. During a hand-object interaction, it is common for the hand to occlude most of the object in order to grasp the object. On one hand, this makes it hard to detect active objects. On the other hand, the appearance of the hand actually contains important information about the location, shape, size, and pose of the active object. It is important to develop computer vision algorithms that can leverage each pixel of the image, especially the hands and objects, to accurately estimate a bounding box around the active object.

To advance the state of art in active object detection, we introduce a pixel-wise voting function to improve the active object bounding box estimate while being robust to occlusion. The voting function takes as input an initial bounding box estimate of the active object (typically seeded by bounding box around the hand region), and then predicts an improved bounding box, where the improved bounding box

1

is tighter and more centered around the active object. Inside the voting function, we predict a large set of improved active object bounding boxes, by allowing each pixel inside the input box to regress a new bounding box. We call the collection of active object bounding boxes as the *Relational Box Field* (RBF), as they represent a field of bounding boxes related to pixels inside the input bounding box. As we can observe in fig. 3.2, the pixel-wise predictions can be quite diverse because they depend on features from different locations in the input image. Our method overcomes inconsistencies across the RBF by using a technique similar to [10, 27, 33, 38], where our voting function finds the consensus from pixel-wise predictions by selecting the bounding box with a majority vote. Similar to the Hough transform, our voting scheme is able to minimize the influence of outliers and generate stable predictions through the power of aggregation. We also show later in our experiments that our voting scheme is more robust when compared to standard regression methods.

While the voting function can provide a better active object bounding box estimation, one round of voting is typically not enough to accurately localize the active object. Similar to the idea of boosting [13] where one uses a sequence of computational units to iteratively improve prediction performance, we apply multiple rounds of the voting function to progressively obtain a more accurate active object bounding box. However, repeatedly applying a function trained for one-step prediction (*i.e.,* supervised learning or behavior cloning) can result in a data distribution shift, also known as covariate shift [32]. For example, each time a one-step predictor is used in an auto-regressive manner (*i.e.,* the output is passed as input for the next sequence), it can introduce small errors which can compound over the prediction sequence, leading to a data distribution shift, which can lead to bad performance towards the end of a sequence. In order to mitigate this issue, we use reinforcement learning (RL) to learn the proposed voting function. RL is designed to account for distribution shifts in auto-regressive processes by evaluating and optimizing over sequences. Specifically, we use a Markov Decision Process (MDP) to model the sequential decision-making process of obtaining an optimal active object bounding. In each step of the sequential decision-making process, the voting function (the MDP policy) takes an initial estimate of the active object bounding box as input (the MDP

2

state) and predicts a new improved active object bounding box (the MDP action and next state). Interestingly, we find that the first estimate of the bounding box of the active object can be seeded using a hand bounding box since active objects are usually near the hands. We demonstrate that using reinforcement learning provides a meaningful improvement over standard supervised learning models for active object detection.

Our approach is evaluated on two large-scale hand-object interaction datasets: 100DOH [31] and MECCANO [28] datasets. Experiments show that our method achieves new state-of-the-art performance on both hand-object detection and active object detection tasks. We also demonstrate the better generalization ability of our model by evaluating its performance across the datasets. Last, we provide a comprehensive ablation study for our design choices.

Chapter 2

Background

2.1 Human-object Interaction Detection

As shown in fig. 2.1, existing approaches [10, 14, 18, 31] in human(or hand)-object interaction usually detect active objects in two steps: (1) detecting hand and object *independently* with a classical detector [4, 9, 29, 39], followed by (2) an hand-object interaction detection to match hands and objects detected in the first step. One limitation of using classical object detectors is that they are not designed to detect objects under occlusions. Meanwhile, in these methods, the important relationship between hand and object is also not considered when localizing the active object. Instead, our method achieves an accurate localization of the active object in a sequential decision-process dependent on hand, which is robust to occlusions and fully exploits the feature of hand, object, and their inter-dependency like human perception [12, 25].

2.2 Object Detection with Reinforcement Learning

Different from classic object detection methods [9, 29, 39], RL-based detection approaches [3, 16, 22] progressively narrow down the scope from the initial guess to the object bounding box in a top-down sequential process based on the current observation and historical paths. Some RL methods [3, 16, 22] localize the visual objects in scenes using top-down sequential policies. Some methods [20, 36] combine RL with classical detectors to improve object detection performance or efficiency. For instance, [36] uses RL to efficiently generate bounding box proposals to replace the first stage of two-stage detectors like [29].

Similar to [3] training a localization agent that starts with the whole scene and narrows down to the object, we design an agent which starts from the hand box and progressively moves to the object touched by the hand. The key difference is our agent does not directly predict a single action that deforms the bounding box using translation and scaling. Instead, we propose a voting function that allows each pixel to have a different prediction and then finds the consensus from pixel-wise predictions, which essentially enables the agent to ignore noisy observation caused by occlusions.

2.3 Dense Methods

Most dense methods make a prediction for each pixel or patch in the image, then use voting to aggregate the results into a robust estimation. [27, 38] use pixel-wise predictions with Hough voting to localize keypoint for pose estimation. [33] uses patch-based voting to remove false-positive hypotheses from the background for joint object detection and depth recovery. [10] uses bounding-box-wise voting for human-object interaction detection. Different from the above methods, our proposed method progressively performs pixel-wise voting on the *Relational Box Field* to predict and improve the active object estimation from hand.
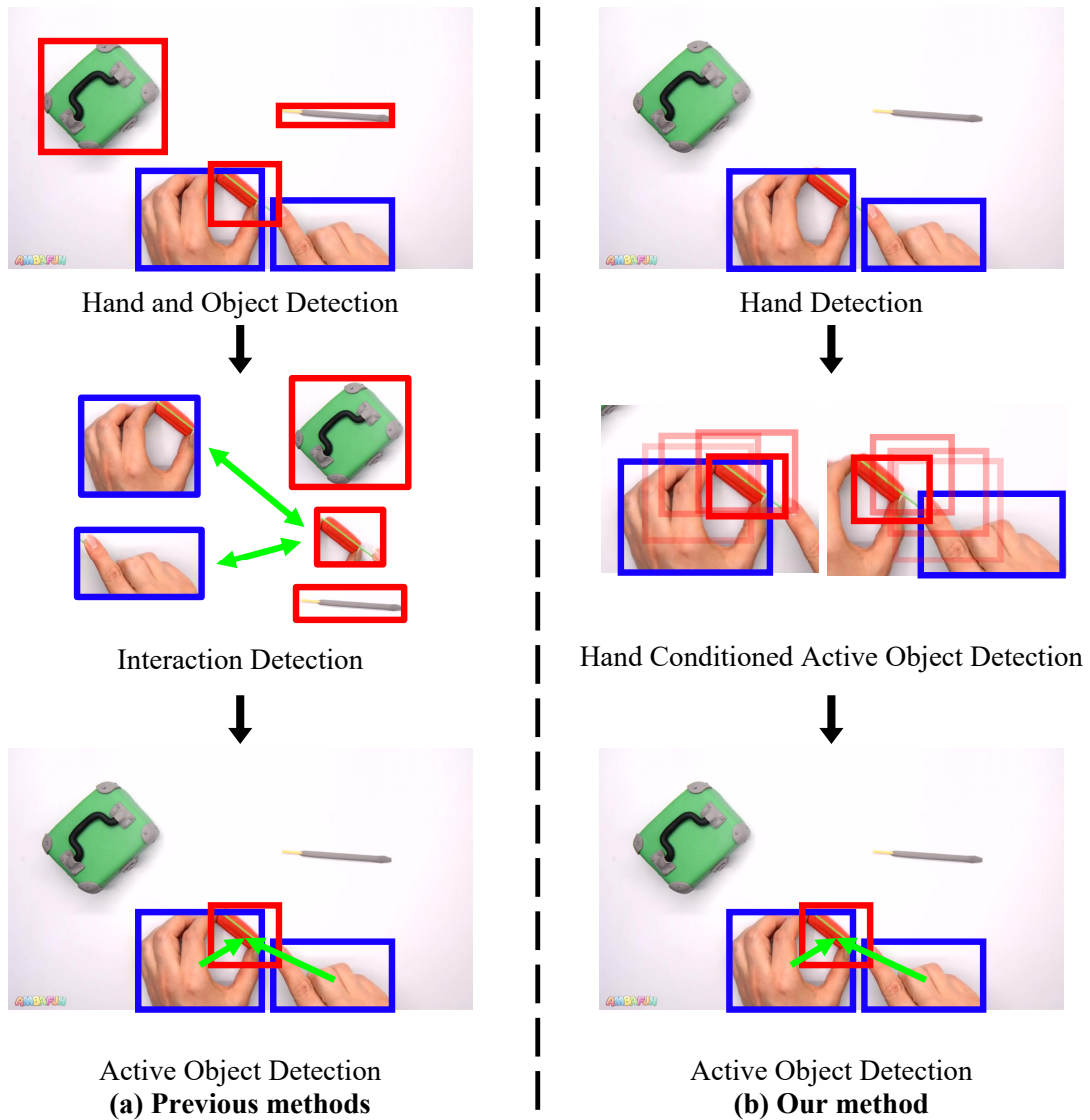
Figure 2.1: Previous methods detect active object by first detecting hands and objects *independently*, followed by an interaction detection to match active objects and hands. Our method directly detects the active object corresponding to each hand a sequential decision-making process dependent on hand.

Chapter 3

Method

The overview structure of our framework is illustrated in fig. 3.1. To achieve robust object localization, especially under occlusion, we propose a voting function with the Relational Box Field that allows each pixel in the image to vote for a bounding box of the active object. To further refine the active bounding box estimation, we repeatedly apply the voting function until the bounding box converges.

In the following subsections, we first explain the Relational Box Field with Pixel-wise Voting for improving the active object bounding box estimation. Then we describe how to use an MDP to model the sequential decision-making process that progressively obtains a more accurate active object bounding box. Finally, we describe the implementation details of the model and the hybrid training strategy consisting of imitation learning and reinforcement learning to learn an optimal policy efficiently.

3.1 Relational Box Field with Pixel-wise Voting

We describe a voting function for estimating an improved bounding box of an object from the input box. Specially, we first predict a *Relational Box Field* which encodes bounding box predictions for every pixel inside the image. Then the predictions from the pixels inside the input box are aggregated into a single improved object box through pixel-wise voting.
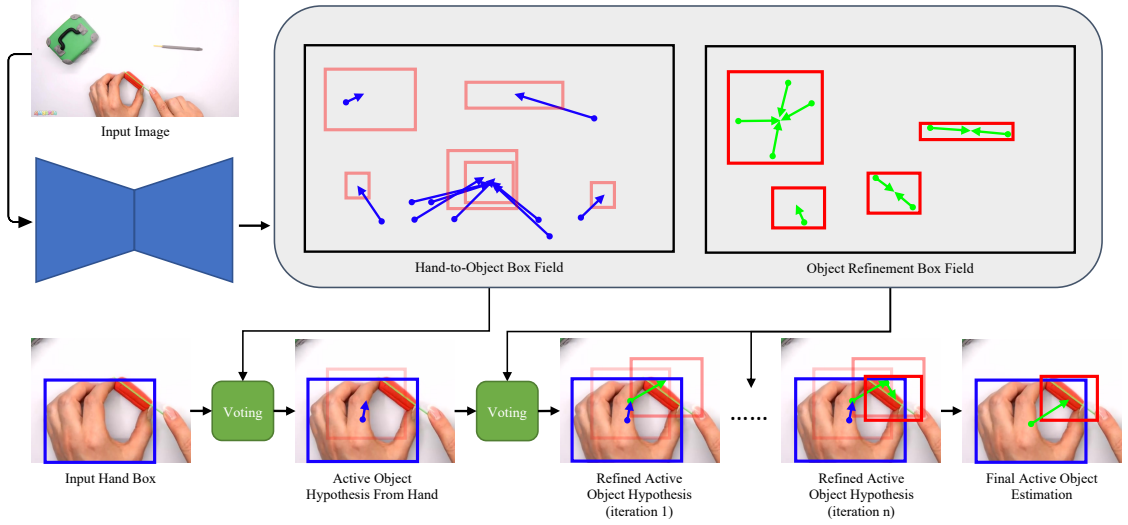
Figure 3.1: Overview of the sequential decision-making process for active object detection. Our approach, seeded using a hand bounding box, progressively refines the current active object bounding box towards an optimal active object estimation. In each step, we use a voting function on the Relational Box Fields to predict a better active object bounding box based on the current input.

Relational Box Field The *Relational Box Field F* is a field of active object bounding boxes in relationship to each pixel inside the image. Specifically, the estimated Relational Box Field $\hat{F} \in \mathbb{R}^{H \times W \times 5}$ has the same spatial resolution as the input image $I \in \mathbb{R}^{H \times W \times 3}$. For a location $(u, v) \in \mathbb{R}^2$, $\hat{F}_{u,v} \in \mathbb{R}^5$ represents its related object box in the form of

$$\hat{F}_{u,v} = [\hat{r}_{u,v}, \hat{\theta}_{u,v}, \hat{h}_{u,v}, \hat{w}_{u,v}, \hat{c}_{u,v}] \tag{3.1}$$

where $\hat{r}_{u,v} \in \mathbb{R}^+$ and $\hat{\theta}_{u,v} \in [0, 2\pi)$ represent the predicted relative displacement from (u, v) to its related bounding box center in polar coordinate system. $\hat{h}_{u,v} \in (0, H]$ and $\hat{w}_{u,v} \in (0, W]$ are the height and width of the bounding box prediction at (u, v). $\hat{c}_{u,v} \in [0, 1]$ is the confidence score of the prediction at (u, v).

Loss for Relational Box Field During training, we supervise the Relational Box Field with respect to the set of ground truth bounding boxes $\mathcal{A}$. Note that bounding

10

boxes may overlap with each other. To alleviate the confusion due to overlapping, we remove the overlapped regions from each bounding box $a \in \mathcal{A}$ during training as clipped bounding boxes $a^\star$:

$$a^\star = a \cap \left(\coprod \mathcal{A}\right) \qquad (3.2)$$

where $\coprod$ denotes the disjoint union, *i.e.,* the union of non-overlapped areas in $\mathcal{A}$.

For each pixel (u, v) inside the clipped bounding box $a^\star$, we apply localization loss $L_{u,v}^{loc}$ – smooth-L_1 regression loss on the predicted box parameters $(\hat{r}_{u,v}, \hat{\theta}_{u,v}, \hat{h}_{u,v}, \hat{w}_{u,v})$, and Focal Loss $L_{u,v}^C$ on the predicted confidence score $\hat{c}_{u,v}$. Both $L_{u,v}^{loc}$ and $L_{u,v}^C$ are only applied in $a^\star$. Given predicted $\hat{F}$, the overall loss applied to a bounding box a is the average of pixel-wise losses:

$$L_{\hat{F}}(a) = \frac{\sum_{u,v \in a^\star} L_{u,v}^{loc} + L_{u,v}^C}{|a^\star|} \qquad (3.3)$$

To reduce the bias towards large bounding boxes, we compute the total loss of Relational Box Field by averaging the loss of each bounding box

$$L_{\hat{F}}(\mathcal{A}) = \frac{\sum_{a \in \mathcal{A}} L_{\hat{F}}(a)}{|\mathcal{A}|} \qquad (3.4)$$

Weighted Pixel-wise Voting To improve the robustness against noise in the Relational Box Field prediction, we use pixel-wise voting to aggregate the predicted bounding box results. The voting function summarizes the bounding box predictions from all pixels belonging to the input box a into an improved bounding box estimation $(\hat{x}^a, \hat{y}^a, \hat{h}^a, \hat{w}^a)$ as

$$\hat{x}^a, \hat{y}^a, \hat{h}^a, \hat{w}^a = \text{Vote}_{\hat{F}}(a) \qquad (3.5)$$

To identify the best location of the improved bounding box, we create a histogram of votes over bounding box parameters, where the predictions with larger confidence scores $\hat{c}_{u,v}$ contributes with larger weights Specifically, we compute the voting scores of the target object box center $S_{\text{center}}^a \in \mathbb{R}^{H \times W}$, width $S_{\text{width}}^a \in \mathbb{R}^W$, and height

$S^a_{\text{height}} \in \mathbb{R}^H$ for the improved bounding box estimation from the input box a as

$$S^a_{\text{center}}(x,y) = \sum_{u,v \in a} \hat{c}_{u,v} \cdot \mathbb{1}(\lfloor u + \hat{r}_{u,v} \sin(\hat{\theta}_{u,v}) \rceil = y) \cdot$$

$$\mathbb{1}(\lfloor v + \hat{r}_{u,v} \cos(\hat{\theta}_{u,v}) \rceil = x)$$

$$S^a_{\text{width}}(w) = \sum_{u,v \in a} \hat{c}_{u,v} \cdot \mathbb{1}(\lfloor \hat{w}_{u,v} \rceil = w)$$

$$S^a_{\text{height}}(h) = \sum_{u,v \in a} \hat{c}_{u,v} \cdot \mathbb{1}(\lfloor \hat{h}_{u,v} \rceil = h)$$

$$(3.6)$$

where $\mathbb{1}(\cdot)$ is the indicator function and $\lfloor \cdot \rceil$ is the function of rounding to the nearest integer. The optimal bounding box parameters $(\hat{x}_a, \hat{y}_a, \hat{h}_a, \hat{w}_a)$ are then obtained by retrieving the candidate with maximum voting score in $S^a_{\text{center}}, S^a_{\text{width}}, S^a_{\text{height}}$ respectively.

As shown in fig. 4.2, the informative patterns (such as fingers and objects) produce more consistent predictions compared to irrelevant information such as background and occlusions. Since the voting function picks the box with the most votes as output, it allows the model to explicitly focus on informative patterns instead of irrelevant information.

3.2 MDP

Recall the goal of our method is to localize the active object bounding box $b^o = [x^o, y^o, w^o, h^o] \in \mathbb{R}^4$ corresponding to a hand bounding box $b^h = [x^h, y^h, w^h, h^h] \in \mathbb{R}^4$ in an image I, where the (x^o, y^o) and (x^h, y^h) are the centers of the active object and hand box, and w^o, h^o, w^h, h^h stand for their widths and heights. Considering the remarkable success in hand detection [2, 24, 31], we assume the hand boxes is given. As the grasp appearance of human hands is highly indicative of the location and shape of the object being manipulated, we leverage this important clue by first generating an active object bounding box hypothesis using visual clues inside the

hand bounding box. However, the object hypothesis based on only hand appearance could be ambiguous because a similar hand grasp could hold various objects. Thus, it is important to further refine the object estimation by incorporating object patterns from the estimated object bounding box as direct evidence.

We find only one step of pixel-wise voting from the Relational Box Field could be insufficient to localize the active object accurately. Therefore, we propose to further refine the active object localization based on the updated estimation of the bounding box in a sequential decision-making process, which is modeled by an MDP. The state, action, dynamics, policy, and reward of the MDP are defined as follows.

State   The state s_t at each timestamp t consists of two parts: a local state and a global state feature which is consistent throughout the sequence. We use the current active object bounding box estimation $\hat{b}_t^o = [\hat{x}_t^o, \hat{y}_t^o, \hat{w}_t^o, \hat{h}_t^o] \in \mathbb{R}^4$ as the local state, where $(\hat{x}_t^o, \hat{y}_t^o)$ represents the center of estimated bounding box and $\hat{w}_t^o, \hat{h}_t^o$ stand for its width and height. The global state representation is a combination of the detected hand bounding box b^h and the image feature $\mathcal{F} \in \mathbb{R}^{H \times W \times 256}$ extracted from image I.

$$s_t = (\hat{b}_t^o, b^h, \mathcal{F}) \tag{3.7}$$

Our method does not require an initial active object bounding box guess. Instead, we exploit the visual clue inside the hand bounding box to generate the initial active object bounding box hypothesis. In other words, the local state $\hat{b}_0^o$ at timestamp $t = 0$ can be arbitrary initialized because it is ignored by the policy.

Action   Each action at timestamp t is in the format of $a_t = [\Delta x_t, \Delta y_t, \Delta w_t, \Delta h_t] \in \mathbb{R}^4$, which are the offsets to update the center coordinate, width and height of the active bounding box estimation.

Actions that do not significantly change the bounding box lead to the *terminal* state. We use the relative changes of the center, height, and width of the bounding

box as the criterion for termination. Specifically, we check:

$$\frac{|\Delta x_t|}{\hat{w}_t^o}, \frac{|\Delta y_t|}{\hat{h}_t^o}, \frac{|\Delta w_t|}{\hat{w}_t^o}, \frac{|\Delta h_t|}{\hat{h}_t^o} \qquad (3.8)$$

The *terminal* state will be triggered if all the relative changes are below a threshold. We set the threshold value to 0.05 in all of our experiments.

Dynamics Our state transition dynamics $h : (s_t, a_t) \mapsto s_{t+1}$ is a deterministic function, which leads to exactly one next state from one state-action pair. It updates the current state s_t by adding the action a_t (the offsets for bounding box parameters) to the current active object bounding box estimation $\hat{b}_t^o$. Except when timestamp $t = 0$, the action is applied to the hand bounding box b^h since our method is conditioned on the hand location and appearance for the initial active object hypothesis.

Policy The policy generates an action a_t by applying the voting function on two relational box fields: hand-to-object box field $\hat{F}^{ho}$ and object refinement box field $\hat{F}^{oo}$, which are predicted based on the image feature $\mathcal{F}$ in the current state s^t.

The hand-to-object box field F^{ho} addresses the *contact* relation, which is used to generate an initial hypothesis of the active object box $\hat{b}_1^o$. Specifically, F^{ho} encodes the mapping from the hand pixels to the corresponding active object bounding box

$$F^{ho} : (u, v) \mapsto (x^o, y^o, w^o, h^o) \qquad \text{for } u, v \in b^h \qquad (3.9)$$

In F^{ho}, the ground truth confidence score of contact relation $c_{u,v}^{ho} = 1$ if (u, v) lies in a hand which is touching an object, while $c_{u,v}^{ho} = 0$ otherwise.

The object refinement relational box field F^{oo} exploits object local patterns inside the current object box estimation to further improve it. Specifically, F^{oo} encodes the mapping from the pixels of active objects to their box center, width, and height.

$$F^{oo} : (u, v) \mapsto (x^o, y^o, w^o, h^o) \qquad \text{for } u, v \in b^o \qquad (3.10)$$

To summarize, at $t = 0$, the policy generates an initial active object hypothesis by

14

applying the voting in eq. (3.5) to bounding box predictions within $\hat{F}^{ho}$ from pixels belonging to the hand bounding box h^t. When $t > 0$, the policy applies the voting in eq. (3.5) to predictions within $\hat{F}^{oo}$ from pixels belonging to the current active object estimation $\hat{b}_t^o$ to vote for a refined active object bounding box. Specifically, the policy finds a local optimal active object bounding box estimation $\hat{b}_{t+1}^o$ as:

$$\hat{b}_{t+1}^o = \begin{cases} \text{Vote}_{\hat{F}^{ho}}(b^h) & \text{if } t = 0 \\ \text{Vote}_{\hat{F}^{oo}}(\hat{b}_t^o) & \text{otherwise} \end{cases} \tag{3.11}$$

where $\hat{b}_t^o$ is the local state and b^h belongs to the global state representation in s_t defined in eq. (3.7).

The output action of the policy $a_t = \pi(s_t)$ is defined as the displacement towards the local optimal active object bounding box estimation $\hat{b}_{t+1}^o$ from either current estimation $\hat{b}_t^o$ when $t > 0$, or the hand bounding box b^h when $t = 0$.

Reward We design the reward r as the accuracy of the active object bounding box estimation. As the action a_t updates the active object bounding box from $\hat{b}_t^o$ to $\hat{b}_{t+1}^o$, we use the GIoU [30] between b_{t+1}^o and the ground truth b_o as the reward at timestamp t

$$r(s_t, a_t) = \text{GIoU}(\hat{b}_{t+1}^o, b^o) \tag{3.12}$$

3.3 Implementation Details

In this subsection, we describe the implementation details of our image feature extractor and the policy model.

Image Feature Extractor Our image feature extractor f is in an encoder-decoder style with self-attention. The network takes an input image $I \in \mathbb{R}^{H \times W \times 3}$ and outputs a feature map $\mathcal{F} \in \mathbb{R}^{H \times W \times 256}$. For the encoder, we use a pre-trained ResNet101 on ImageNet and modify it by changing the last two layers into dilated convolution to increase the spatial size of the feature map. To further exploit the synergy between

hand and object, we apply one layer of image-wide self-attention (with details in the supplementary) on the deep feature before forwarding it to the decoder. In the decoder, the Atrous Spatial Pyramid Pooling from DeepLabV3+ [5] and bi-linear up-sampling are repeatedly performed to efficiently expand the feature map until its size matches the input size.

Policy Model   Our light-weighed policy model is a 1×1 convolution layer with an input channel of size 256 and an output channel of size 14. The policy model takes the image feature $\mathcal{F}$ and outputs the prediction of hand-to-object relational box field F^{ho} and object refinement relational box field F^{oo}.

3.4   Finding a Policy with Hybrid Training

It is challenging to learn the image feature extractor and policy model from scratch: a massive amount of random actions, in the beginning, provide weak training signals, and optimizing the image feature extractor for each state is slow.

While our policy uses a deterministic voting scheme to predict the action, a set of optimal hand-to-object relational box field F^{ho} and object refinement relational box field F^{oo} would generate the same desired behavior as learning by the reward function. This design enables us to first use imitation learning to learn these two relational box fields in a fully supervised mode. Specifically, we perform multi-task training with an overall objective which is a linear combination of losses from hand-to-object and object refinement box fields following eq. (3.4) to jointly pretrain the image feature extractor and policy model

$$L_{IL} = L_{F^{ho}}(\mathcal{H}) + L_{F^{oo}}(\mathcal{O}) \tag{3.13}$$

where $\mathcal{H}$ and $\mathcal{O}$ are the set of hand bounding boxes and active object bounding boxes in the image. In imitation learning, we use a batch size of 28 and an initial learning rate of 10^{-4} for 100 epochs with $10\times$ learning rate drop every 30 epochs.

After pretraining, we freeze the weights in the image feature extractor and further

train the policy model using reinforcement learning (RL) on the sequential predictions for the active object b_o. We follow the standard policy gradient on the cumulative reward. For computational efficiency, we set the horizon $T = 5$. In consistent with the ultimate objective of getting a final accurate active object, we do not discount the future reward by setting a discount factor $\gamma = 1$. To improve the training stability, we add the losses of relational box fields as an auxiliary loss

$$L_{RL} = \sum_{t=0}^{T}(1 - r(s^t, a^t)) + L_{F^{ho}}(b^h) + L_{F^{oo}}(b^o) \tag{3.14}$$

For reinforcement learning, we use a batch size of 48 and a learning rate of 10^{-5} for 5 epochs. We show that RL boosts the precision of localization in the supplementary.
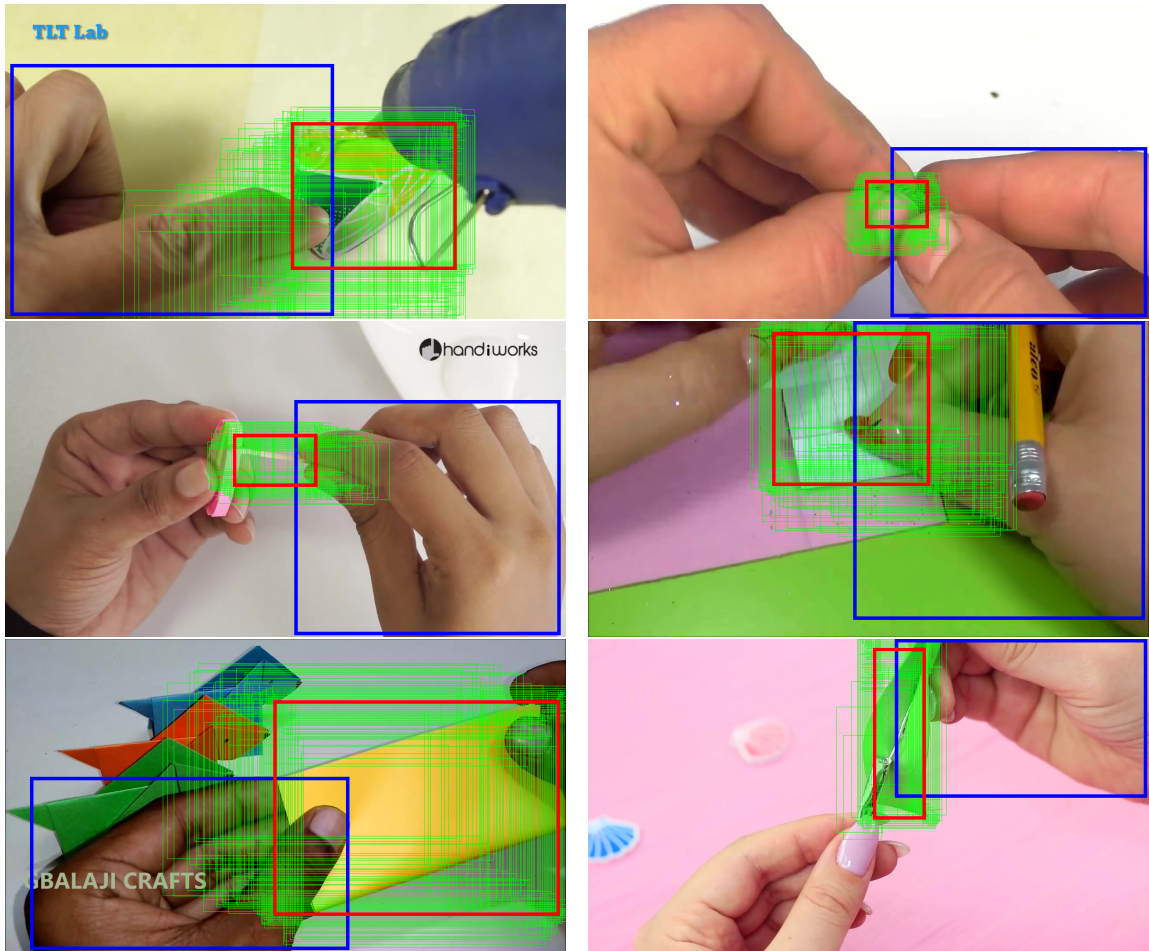
Figure 3.2: Relational Box Field and Pixel-wise Voting visualization. Each green bounding box is an estimated active object bounding box for a pixel inside the blue input bounding box (initialized with a detected hand box). The voting function selects the majority vote prediction (red) as the improved active object bounding box estimate. To ensure visibility, we only show 200 sampled predictions.

Chapter 4

Experiments

In this section, we demonstrate the performance of our approach on hand-object detection and active object detection tasks. In order to show our method is capable of tackling both third-person and egocentric real-life applications, we evaluate on two challenging datasets: 100DOH [31] and MECCANO [28].

4.1 Experiments on 100DOH

100DOH [31] is a large-scale benchmark for hand-object interaction. It has 99,899 frames (79,921 for training, 9,995 for validation and 9,983 for testing). Among the dataset, there are 189.6K annotated hands with 110.1K objects. For each hand, it is either paired with an active object bounding box or has no object.

We use the same hand-object detection evaluation metrics used in [31], which calculates the average precision (AP) of the tuple $(hand, object)$. Specifically, a tuple is considered as true positive if and only if: 1) the IoU between the predicted hand bounding box and ground truth is greater than or equal to the IoU threshold; and 2) the IoU between the predicted object bounding box and ground truth is greater than or equal to the IoU threshold.

Since active objects are usually close to hands, we construct a *Simple Baseline* by first detecting hands and objects independently using Faster-CNN [29], and then

Methods	Backbone	# of Params	Hand Source	AP^{50}_{hand}	AP^{75}	AP^{50}	AP^{25}
Simple Baseline	R101	47M	FasterRCNN	89.59	28.15	44.73	47.57
100DOH Detector	R101	47M	FasterRCNN	89.59	28.50	46.95	51.80
PPDM	DLA34	21M	CenterNet	89.64	26.89	45.80	53.04
HOTR	R50	51M	DETR	90.26	29.30	49.27	**57.80**
Ours	R101	48M	FasterRCNN	89.59	**29.90**	**53.02**	57.15
Simple Baseline	R101	47M	Ground Truth	100	34.51	44.68	52.35
Ours	R101	48M	Ground Truth	100	**40.05**	**54.82**	**64.86**

Table 4.1: Results of hand-object interaction detection on 100DOH. The Simple Baseline method is described in section 4.1.

assigning the closest object (in terms of box center Euclidean distance) to each hand as the active object. Besides *Simple Baseline* and the baseline in the dataset paper, we also adapt two recent methods on human-object interaction detection, PPDM [18] and HOTR [17], to our task by changing the subject from human to hand. We assign the object with the highest score in detected interaction tuple ($hand, object$) to the corresponding hand for evaluation.

We report the AP at IoU thresholds of 0.75, 0.5, and 0.25. As a fair comparison against the baselines, we use detected hand bounding boxes from Faster-RCNN [29] as the input of the proposed method. As illustrated in table 4.1, our method outperforms all baseline with similar hand detection as input. We also exhibit the maximum performance when ground truth hand bounding boxes are available. Some qualitative results with comparisons are displayed in fig. 4.1, which shows the proposed method not only provides more precise object localization but also is more robust to occlusions. More qualitative results and visualization of object refinement are shown in the supplementary.

4.2 Experiments on MECCANO

MECCANO [28] is an egocentric dataset for human-object interaction understanding. It contains 64,349 frames which are annotated with active object boxes. Since there is no hand bounding box and hand-object correspondence annotation in the MECCANO dataset, we perform Pseudo labeling in two steps to train our model. First, we adapt
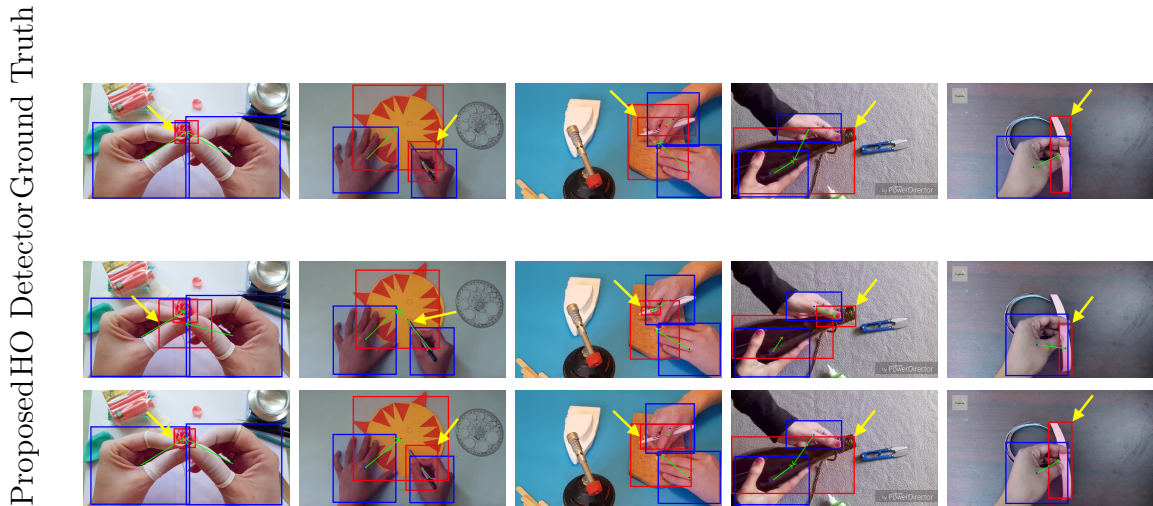
Figure 4.1: Comparison of qualitative hand-object interaction detection results on the 100DOH dataset, where HO Detector denotes 100DOH Detector [31]. The differences are highlighted using bold yellow arrows. It demonstrates our method not only provides more precise object localization (col. 1) but also is more robust to the scenes where objects are overlapped (col. 2, 3) or occluded by hands (col. 2, 3, 4, 5).

the pre-trained hand detector [31] to detect hands in all frames. Second, we assign the closest annotated active object to each hand box in terms of box centers. There is no human annotation involved in the Pseudo labeling, which leads to a fair comparison.

Since MECCANO does not have the interaction annotation to train PPDM [18] and HOTR [17], we compare our methods against 100DOH Detector [31] in terms of standard object detection metrics: average precision (AP) at IoU thresholds of 0.75, 0.5, and 0.25. We first compare the generalization ability by directly evaluating models trained on 100DOH. Then we compare the performance by retraining the proposed method using automatically generated pseudo labels of MECCANO. table 4.2 shows that our method has a better generalization ability when adapted to a new dataset without retraining, and better performance after retraining on the MECCANO dataset. More qualitative results and visualization of object refinement are shown in the supplementary.

Method	Backbone	Finetune	AP^{75}	AP^{50}	AP^{25}
100DOH Detector	R101	✗	-	11.17	-
Ours	R101	✗	**9.09**	**16.61**	**23.97**
100DOH Detector	R101	✓	-	20.18	-
Ours	R101	✓	**12.99**	**26.25**	**34.88**

Table 4.2: Results of active object detection on MECCANO. We compare our method against 100DOH Detector.

4.3   Ablation Studies

We perform ablation studies on both 100DOH and MECCANO datasets. Specifically, we examine the effects of pixel-wise voting, the sequential decision-making process, reinforcement learning. More runtime analysis, and discussion about limitations are included in the supplementary.

Effect of Pixel-wise Voting   The pixel-wise weighted voting is the key component of the proposed method to retrieve a single bounding box estimation from dense predictions while being robust to occlusions. To understand the effect of pixel-wise weighted voting, we visualize the heatmap of the IoU between pixel-wise bounding box predictions and the final predicted bounding box after voting aggregation in fig. 4.2. As expected, voting picks the final estimated bounding box mostly based on the predictions in the regions of informative patterns such as fingers and objects as opposed to irrelevant information such as the background. We further quantitatively examine the effectiveness of pixel-wise voting by comparing it with two aggregation methods: 1) using the prediction from the central pixel of the input box, and 2) averaging the bounding box parameters of all predictions inside the input box. Table 4.3 reveals the superiority of using voting to aggregate dense predictions.

Effect of Sequential Decision-Making Process   We analyze whether applying the voting function multiple times could sequentially improve the active object bounding box estimation. Specifically, we report the performance after applying different numbers of the voting function. The results are shown in table 4.4. For the

Dataset	Aggregation	AP^{75}	AP^{50}	AP^{25}
100DOH	Center	25.00	52.31	56.88
100DOH	Average	25.59	51.77	56.41
100DOH	Vote	**29.90**	**53.02**	**57.15**
MECCANO	Center	9.59	22.85	32.78
MECCANO	Average	12.83	25.76	34.66
MECCANO	Vote	**12.99**	**26.25**	**34.88**

Table 4.3: Ablation studies on different aggregation methods to retrieve estimation from dense predictions.
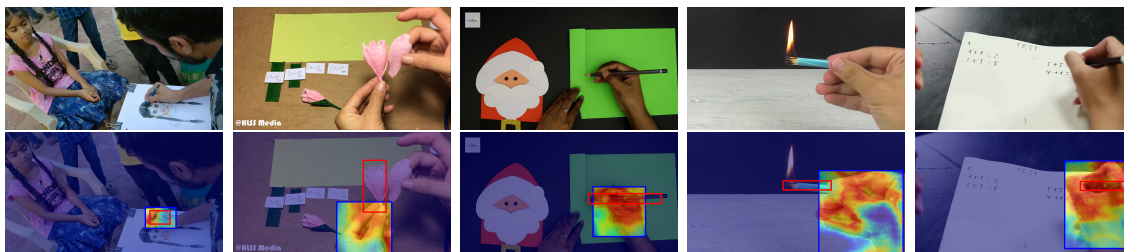

Figure 4.2: We visualize the IoU (red indicates higher IoU) between the final active object box estimation (red) and the pixel-wise predictions inside the hand bounding box (blue). This figure shows voting is able to adapt predictions from informative hand parts like fingers as opposed to irrelevant parts like wrist and background.

100DOH dataset, the performance converges after applying the voting function two times. On the MECCANO dataset, the performance is progressively improved with more iterations of voting.

Effect of Reinforcement Learning Repeatedly applying the voting function trained for one-step prediction (supervised learning) could result in a data distribution shift issue. Specifically, the small error at each step could compound the sequential predictions, which leads to a bad performance towards the final prediction. The application of RL is to mitigate this issue by optimizing over the sequence with an accumulative loss for the sequential predictions. We examine the effect of RL by comparing the performance with and without using RL. The results are shown in table 4.5, which demonstrate that RL gives significant improvements for AP^{75} and AP^{50} on 100DOH dataset.

Dataset	# of Voting	AP^{75}	AP^{50}	AP^{25}
100DOH	1	29.18	49.58	53.11
100DOH	2	29.80	**53.11**	**57.47**
100DOH	3	**29.92**	53.10	57.23
100DOH	∞	29.90	53.02	57.15
MECCANO	1	7.33	22.61	32.94
MECCANO	2	8.97	22.56	32.46
MECCANO	3	10.38	23.91	33.03
MECCANO	∞	**12.99**	**26.25**	**34.88**

Table 4.4: Ablation studies on the number of voting of the sequential decision-making process.

Dataset	RL	AP^{75}	AP^{50}	AP^{25}
100DOH	✗	23.64	46.84	**57.44**
100DOH	✓	**29.90**	**53.02**	57.15
MECCANO	✗	**13.13**	26.21	34.88
MECCANO	✓	12.99	**26.25**	**34.88**

Table 4.5: Ablation studies on reinforcement learning (RL) on 100DOH and MEC-CANO datasets.

Chapter 5

Conclusions

In this paper, we propose a voting function with Relational Box Field to leverage each pixel as evidence to robustly predict the bounding box of the active object, despite under occlusions. The voting function is applied repeatedly to improve the active object estimation. We use an MDP to model the sequential decision-making process and apply reinforcement learning to learn an optimal policy. Our method achieves state-of-the-art performance on both hand-object detection and active object detection tasks, as well as better generalization ability across datasets. In the future, we will try to develop a stochastic MDP policy to further explore the state space.

Appendix A

Supplementary

A.1 Overview

In this supplementary, we provide additional implementation and experimental details, as well as qualitative results and analysis. We present the details of the self-attention layer in appendix A.2.1 and the confidence calculation in appendix A.2.2. We validate that the proposed method is more robust to detect active objects under occlusion in appendix A.3. We illustrate additional qualitative results and visualizations in appendix A.4. The inference running time is presented in appendix A.5.

A.2 Implementation Details

A.2.1 Self-attention Layer

As described in section 3.3, inside the image feature extractor, we use a self-attention layer between the encoder and the decoder to further exploit the synergy between hands and objects. The architecture of the self-attention layer is illustrated in fig. A.1. The self-attention layer takes the image feature $\mathcal{F}_{\text{deep}}$ from the encoder as input and computes the query, key, and value embeddings (Q, K, and V) from $\mathcal{F}_{\text{deep}}$ using learnable embedding matrices W_q, W_k and W_v. Then the relationships between every
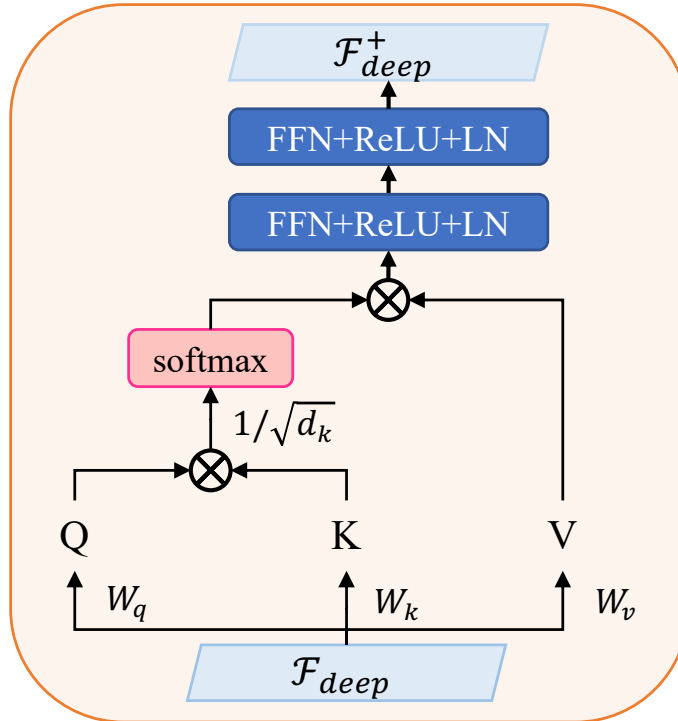
Figure A.1: The architecture of the self-attention layer

spatial location in the feature map are computed using query Q and key K, which is used as the weight to average V. Finally, a two-layer MLP with layer normalization [1] is applied as

$$Q = W_q \mathcal{F}_{\text{deep}}, K = W_k \mathcal{F}_{\text{deep}}, V = W_v \mathcal{F}_{\text{deep}}$$
$$\mathcal{F}_{\text{deep}}^+ = \text{MLP}(\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V) \tag{A.1}$$

where d_k is the feature dimension of the key and $\mathcal{F}_{\text{deep}}^+$ is the feature map after applying self-attention, which is forwarded to the decoder. Empirically, we find marginal improvement from positional encoding, so we omit it for simplicity.

In order to avoid exhaustive computation, we set $d_k = 256$, and reduce the feature dimension of $\mathcal{F}_{\text{deep}}$ from 2048 to 512 by a convolutional layer with a kernel size of 1×1. Following [37] , we use 8 attention heads to address multiple relations between hands and objects.

A.2.2   Confidence Calculation

By definition, an active object must be manipulated by a human hand. We first predict a contact score $s_{b^h}^{\text{contact}}$ representing the probability that a given hand b^h is manipulating an object. Besides, we predict a object probability score $s_{\hat{b}^o}^{\text{obj}}$ for the final object estimation $\hat{b}^o$ of the given hand. To compute $s_{b^h}^{\text{contact}}$ and $s_{\hat{b}^o}^{\text{obj}}$, we use the average of confidence scores inside predicted hand-to-object $\hat{F}^{ho}$ and object refinement $\hat{F}^{oo}$ box fields as

$$s_{b^h}^{\text{contact}} = \frac{\sum_{u,v \in b^h} \hat{c}_{u,v}^{ho}}{|b^h|}, s_{\hat{b}^o}^{\text{obj}} = \frac{\sum_{u,v \in \hat{b}^o} \hat{c}_{u,v}^{oo}}{|\hat{b}^o|} \tag{A.2}$$

We suppress the object detection by a object probability threshold t_{obj}. The final confidence $\hat{c}_{b^h}$ of the hand b^h is the fusion of the hand contact score and the object probability score defined as

$$\hat{c}_{b^h} = \begin{cases} 1 - s_{b^h}^{\text{contact}} & \text{if } s_{b^h}^{\text{contact}} < t_{\text{contact}} \\ s_{b^h}^{\text{contact}} \cdot s_{\hat{b}^o}^{obj} & \text{otherwise} \end{cases} \tag{A.3}$$

We use $t_{\text{obj}} = 0.2$, $t_{\text{contact}} = 0.1$ in all our experiments.

A.3   Analysis: Robustness to Occlusions

To analyze the robustness of our method to occlusions, we compute the recall on the hand-object pairs with three different occlusion levels on 100DOH dataset. The occlusion level of a hand-object pair is measured by the IoU between their bounding boxes. In 100DOH dataset, there are 2222 hand-object pairs with an IoU $\in [.25, .5)$, 348 pairs with an IoU $\in [.5, .75)$, and 14 pairs with an IoU $\in [.75, .1]$. The quantitative comparison in table A.1 shows that our method is more robust in detecting active objects under occlusions over all baselines.

Method	Recall(IoU$\in$[.25, .5))	Recall (IoU$\in$[.5, .75))	Recall(IoU$\in$[.75, 1])
100DOH Detector	68.68	63.22	78.57
PPDM	53.24	53.45	64.29
HOTR	71.69	68.10	71.43
Ours	**77.22**	**78.45**	**100**

Table A.1: Results of hand-object interaction detection for hand-object pairs with different occlusion levels on 100DOH dataset.

A.4 Visualizations

Qualitative Results The qualitative results on 100DOH dataset [31] and MEECANO dataset [28] are presented in fig. A.2 and fig. A.3 respectively. Each green arrow points from a hand bounding box (blue) to the corresponding active object bounding box (red). The visualization shows that our method is able to robustly detect the active object under scenes with overlapping objects and severe occlusions. Most failure cases are due to wrong hand detection, motion blur, and insufficient feature from tiny hands and objects.

Visualization of Iterative Refinement We further visualize the effect of iterative refinement. In this visualization, we show the initial active object hypothesis (yellow bounding box) and the refined active object estimation (red bounding box) on 100DOH dataset (in fig. A.4) and MEECANO dataset (in fig. A.5). All the examples show that the iterative refinement by applying the voting function multiple times could improve the active object bounding box estimation. For better visibility, every sample only shows one pair of hands and objects.

Visualization of Pixel-wise Voting To validate the design of pixel-wise voting, we visualize more examples about the heatmap of the IoU between pixel-wise bounding box predictions and the final predicted bounding boxes after voting in fig. A.6. In this visualization, we clearly observe that the final estimated bounding boxes picked by the voting are related more closely to the predictions in the regions of informative patterns such as fingers and objects as opposed to irrelevant information such as the
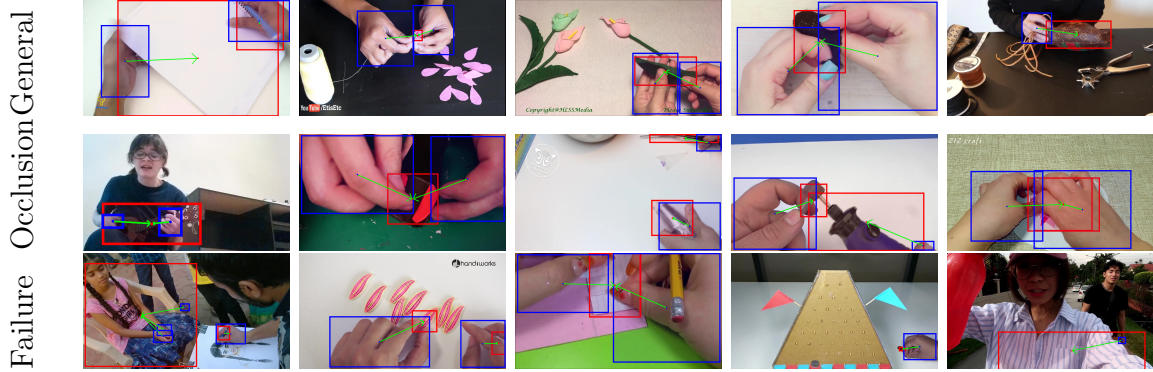
Figure A.2: Qualitative Results on the 100DOH dataset. Each green arrow points from a hand bounding box (blue) to the corresponding active object bounding box (red).
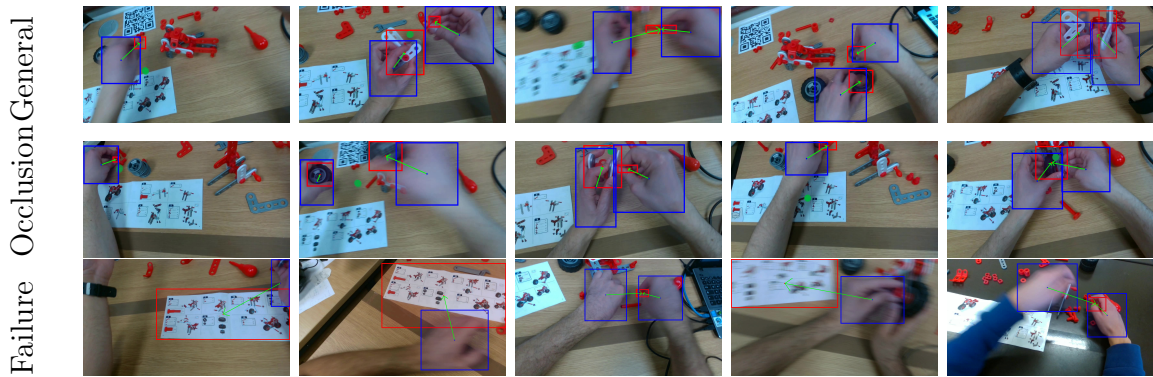

Figure A.3: Qualitative Results on the MECCANO dataset. Each green arrow points from a hand bounding box (blue) to the corresponding active object bounding box (red).

background. For better visibility, every sample only shows one pair of hands and objects.

A.5 Running Time

We report the runtime on a desktop with a Ryzen 3900X CPU and an RTX 2080Ti GPU. For a 512×512 input image with 2 hands on average, the proposed method runs at 18 frames/second, with 13 ms for network forward inference, and 42 ms for active object localization with voting.

Figure A.4: Visualization of iterative refinement on the 100DOH dataset. We show the initial active object hypothesis (yellow bounding box) and the refined active object estimation (red bounding box) corresponding to the hand (blue bounding box).
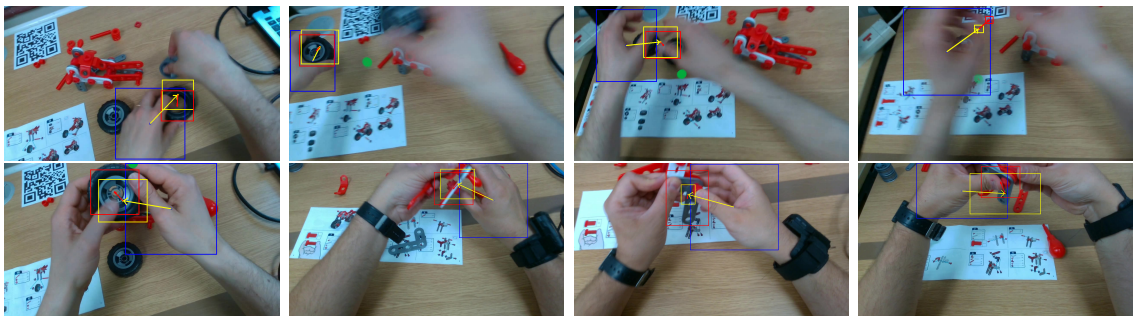

Figure A.5: Visualization of iterative refinement on the MECCANO dataset. We show the initial active object hypothesis (yellow bounding box) and the refined active object estimation (red bounding box) corresponding to the hand (blue bounding box).
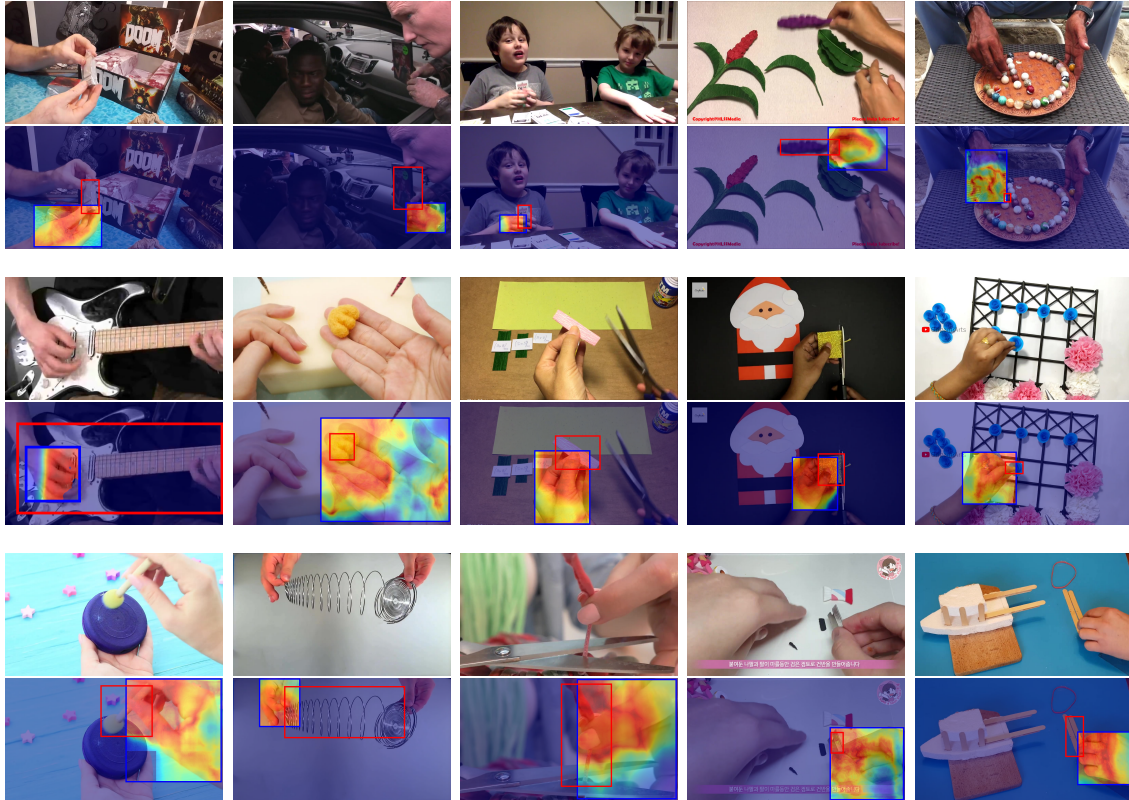
Figure A.6: We show more examples by visualizing the IoU (red indicates higher IoU) between the final active object box estimation (red) and the pixel-wise predictions inside the hand bounding box (blue). The final estimated bounding boxes picked by the voting are more closely related to the predictions in the regions of informative patterns such as fingers and objects as opposed to irrelevant information such as the background.

Bibliography

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. A.2.1

[2] Sven Bambach, Stefan Lee, David J Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the IEEE international conference on computer vision*, pages 1949–1957, 2015. 3.2

[3] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2488–2496, 2015. 2.2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2.1

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 3.3

[6] Yujin Chen, Zhigang Tu, Di Kang, Ruizhi Chen, Linchao Bao, Zhengyou Zhang, and Junsong Yuan. Joint hand-object 3d reconstruction from a single image with cross-branch feature fusion. *IEEE Transactions on Image Processing*, 30: 4008–4021, 2021. 1

[7] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. Robust hand pose estimation during the interaction with an unknown object. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3123–3132, 2017. 1

[8] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6608–6617, 2020. 1

[9] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and

Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019. 2.1, 2.2

[10] Hao-Shu Fang, Yichen Xie, Dian Shao, and Cewu Lu. Dirv: Dense interaction region voting for end-to-end human-object interaction detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1291–1299, 2021. 1, 2.1, 2.3

[11] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *2011 international conference on computer vision*, pages 407–414. IEEE, 2011. 1

[12] J Randall Flanagan, Gerben Rotman, Andreas F Reichelt, and Roland S Johansson. The role of observers' gaze behaviour when watching object manipulation tasks: predicting and evaluating the consequences of action. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368(1628):20130063, 2013. 2.1

[13] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999. 1

[14] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8359–8367, 2018. 2.1

[15] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 571–580, 2020. 1

[16] Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Lu, and Shuicheng Yan. Tree-structured reinforcement learning for sequential object localization. *Advances in Neural Information Processing Systems*, 29, 2016. 2.2

[17] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 74–83, 2021. 4.1, 4.2

[18] Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 482–490, 2020. 2.1, 4.1, 4.2

[19] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,

pages 14687–14697, 2021. 1

[20] Songtao Liu, Di Huang, and Yunhong Wang. Pay attention to them: deep reinforcement learning-based cascade object detection. *IEEE transactions on neural networks and learning systems*, 31(7):2544–2556, 2019. 2.2

[21] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1903, 2016. 1

[22] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. Reinforcement learning for visual object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2894–2902, 2016. 2.2

[23] Kenji Matsuo, Kentaro Yamada, Satoshi Ueno, and Sei Naito. An attention-based activity recognition for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 551–556, 2014. 1

[24] Arpit Mittal, Andrew Zisserman, and Philip HS Torr. Hand detection using multiple proposals. In *Bmvc*, volume 2, page 5. Citeseer, 2011. 3.2

[25] Jiri Najemnik and Wilson S Geisler. Optimal eye movement strategies in visual search. *Nature*, 434(7031):387–391, 2005. 2.1

[26] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *2011 International Conference on Computer Vision*, pages 2088–2095. IEEE, 2011. 1

[27] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 1, 2.3

[28] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1569–1578, 2021. 1, 4, 4.2, A.4

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2.1, 2.2, 4.1

[30] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 3.2

[31] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020. (document), 1, 2.1, 3.2, 4, 4.1, 4.1, 4.2, A.4

[32] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. 1

[33] Min Sun, Gary Bradski, Bing-Xin Xu, and Silvio Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *European Conference on Computer Vision*, pages 658–671. Springer, 2010. 1, 2.3

[34] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4511–4520, 2019. 1

[35] Anand Thobbi and Weihua Sheng. Imitation learning of hand gestures and its evaluation for humanoid robots. In *The 2010 IEEE International Conference on Information and Automation*, pages 60–65. IEEE, 2010. 1

[36] Burak Uzkent, Christopher Yeh, and Stefano Ermon. Efficient object detection in large images using deep reinforcement learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1824–1833, 2020. 2.2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. A.2.1

[38] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 2.3

[39] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2.1, 2.2