Embodied Amodal Recognition: Learning to Move to Perceive Objects

Jianwei Yang[1*] Zhile Ren[1*] Mingze Xu[2]
Xinlei Chen[3] David J. Crandall[2] Devi Parikh[1,3] Dhruv Batra[1,3]
[1]Georgia Institute of Technology [2]Indiana University [3]Facebook AI Research

Abstract

Passive visual systems typically fail to recognize objects in the amodal setting where they are heavily occluded. In contrast, humans and other embodied agents have the ability to move in the environment and actively control the viewing angle to better understand object shapes and semantics. In this work, we introduce the task of Embodied Amodel Recognition (EAR): an agent is instantiated in a 3D environment close to an occluded target object, and is free to move in the environment to perform object classification, amodal object localization, and amodal object segmentation. To address this problem, we develop a new model called Embodied Mask R-CNN for agents to learn to move strategically to improve their visual recognition abilities. We conduct experiments using a simulator for indoor environments. Experimental results show that: 1) agents with embodiment (movement) achieve better visual recognition performance than passive ones and 2) in order to improve visual recognition abilities, agents can learn strategic paths that are different from shortest paths.

1. Introduction

Visual recognition tasks such as image classification [29, 31, 39, 57], object detection [23, 24, 47–49] and semantic segmentation [42, 66, 67] have been widely studied. In addition to recognizing the object's semantics and shape for its visible part, the ability to perceive the whole of an occluded object, known as amodal perception [18, 35, 60], is also important. Take the desk (red bounding box) in the top-left of Fig. 1 as an example. The amodal predictions (top-right of Fig. 1) can tell us about the depth ordering (*i.e.*, desk is behind the wall), the extent and boundaries of occlusions, and even estimaties of physical dimensions [36]. More fundamentally, they help agents understand object permanence that objects have extents and do not cease to exist when they are occluded [6].

*The first two authors contributed equally.

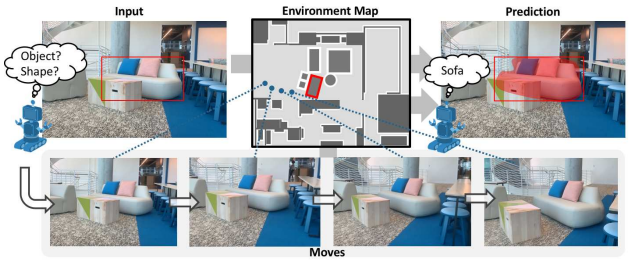

Figure 1: The task of Embodied Amodal Recognition: An agent is spawned close to an occluded target object in a 3D environment, and performs amodal recognition, *i.e.*, predicting the category, amodal bounding box and amodal mask of the target object. The agent is free to move around to aggregate information for better amodal recognition.

Recently, the dominant paradigm for object recognition and amodal perception has been based on single images. Despite leveraging the advances of deep learning, visual systems fail to recognize objects and their shapes from single 2D images in the presence of heavy occlusion. For example in amodal perception, existing work asks the model to implicitly learn the 3D shape of the object *and* the projection of that shape back into the image [19, 40, 70]. This is an entangled task, and deep models are thus prone to overfit to subtle biases in the dataset [22] (*e.g.* learning that beds always extend leftwards into the frame).

Humans have the remarkable recognition ability to infer both semantics and shape for an occluded object from a single image. But humans also have the ability to derive strategic moves to gather information from new viewpoints to help visual recognition. A recent study in [9] shows that toddlers are capable of actively diverting viewpoints to learn about objects, even when they are only 4 – 7 months old.

Inspired by human vision, the key thesis of our work is that in addition to *learning to hallucinate*, agents should *learn to move*. As shown in Fig. 1, to recognize the category and whole shape of a target object indicated by the red bounding box, agents should learn to actively move toward

the target object to unveil the occluded region behind the wall for better recognition.

In this paper, we introduce a new task called *Embodied Amodal Recognition* (EAR) where agents actively move in a 3D environment for amodal recognition of a target object, *i.e.*, predicting its category and amodal shape as well. We aim at systemically studying whether and how embodiment (movement) helps amodal recognition. Below, we highlight three design choices for the EAR task:

Three sub-tasks. In EAR, we aim to recover both semantics and shape for the target object. EAR consists of three sub-tasks: object recognition, 2D amodal localization, and 2D amodal segmentation. With these three sub-tasks, we provide a new test bed for vision systems.

Single target object. When spawned in a 3D environment, an agent may see multiple objects in the field-of-view. We specify one instance as the target, and denote it using a bounding box encompassing its *visible* region. The agent's goal then is to move to perceive this single target object.

Predict for the first frame. The agent performs amodal recognition for the target object observed at the spawning point. If the agent does not move, EAR degrades to passive amodal recognition. Both passive and embodied algorithms are trained using the same amount of supervision and evaluated on the same set of images. As a result, we can create a fair benchmark to evaluate different algorithms.

Based on the above choices, we propose the general pipeline for EAR shown in Fig. 2. Compared with the passive recognition model (Fig. 2a), the embodied agent (Fig. 2b) follows the proposed action from the policy module to move in the environment, and makes predictions on the target object using the amodal recognition module. This pipeline introduces several interesting problems: 1) Due to the agent's movement, the appearances of the observed scene *and* target object change in each step. How should information be aggregated from future frames to the first frame for amodal recognition? 2) There is no "expert" that can tell the agent how to move in order to improve its amodal recognition performance. How do we effectively propose a strategic move without any supervision? 3) In this task, the perception module and action policy are both learned from scratch. Considering that the performance of each heavily relies on the competence of the other, how do we design proper training regime?

To address the above questions, we propose a new model called *Embodied Mask R-CNN*. The perception module extends Mask R-CNN [28] by adding a recurrent network to aggregate temporal features. The policy module takes the current observation and features from the past frames to predict the action. We use a staged training approach to train these two modules effectively.

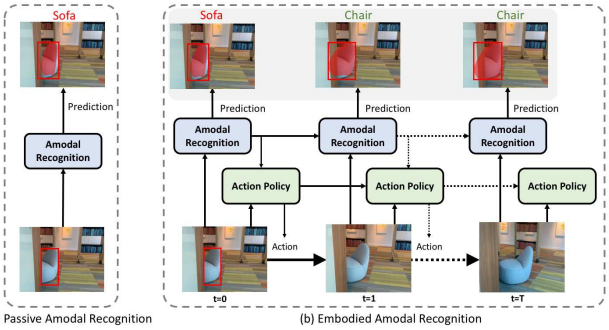

Figure 2: The proposed general pipeline for Embodied Amodal Recognition. To perform amodal recognition (object recognition and amodal perception) on the occluded object, the agent *learns to move* (right), rather than standing still and *hallucinating* (left). The amodal recognition module focuses on predicting the object class, amodal bounding box, and shapes for the first frame. The policy module proposes a next move for the agent to acquire useful information about the object.

Contributions. The main contributions of this paper are:

- We introduce a new task, Embodied Amodal Recognition, where an agent can move in a 3D environment to perform 2D object recognition and amodal perception, including amodal localization and segmentation.
- We build a new dataset for EAR. Using an simulator on an indoor environment, we collect viewpoints for agents so that the target object is partially visible. We also provide precise ground-truth annotations of object classes, amodal bounding boxes, and masks.
- We present a general pipeline for EAR and propose a new model, Embodied Mask R-CNN, to learn to move for amodal recognition. In this model, the amodal recognition and policy modules make predictions at each step, and aim to improve the amodal recognition performance on the target object in the first frame.
- We evaluate both passive and embodied amodal recognition systems, and demonstrate that agents with movements consistently outperform passive ones. Moreover, the learned moves are more effective in improving amodal recognition performance, as opposed to random or shortest-path moves.
- We observe the emergence of interesting agent behaviors: the learned moves are different from shortest-path moves and generalize well to unseen environments (*i.e.*, new houses and new instances of objects).

2. Related Work

Object Recognition. Building object recognition systems is one of the long-term goals of our community. Train-

ing on large-scale datasets [41, 51, 68], we have witnessed the versatility and effectiveness of deep neural networks for many tasks, including image classification [29, 39, 57], object detection [23, 24, 47, 49], and semantic segmentation [28, 42, 66, 67].

Amodal Perception. The *amodal perception* task is the perception of the whole shape of an occluded physical structure. In contrast to classical object recognition, examples of the representations for amodal perception systems are amodal bounding boxes [50, 54], and 3D volumetric [13, 63] or layered reconstructions [53, 58]. In this paper, we focus on amodal segmentation, for both visible and occlude object parts. Recent work learns to hallucinate the full segmentation using labeled datasets [19, 21, 40, 70]. We would like to build the capability of agents to move around and change viewing angles in order to perceive the occluded objects. This is the goal of *active vision*.

Active Vision. Active vision has a long history of research [2, 7, 62], and also has connections to developmental psychology [9]. Recent work learns active strategies for object recognition [17, 32–34, 38, 43], object localization/detection [11, 25, 45], object manipulation [12], instance segmentation [46], feature learning [1], and scene synthesis [20, 59]. However, all of them assume a constrained scenario where either a single image is provided or the target object is localized in different views. Moreover, the agent is not embodied in a 3D environment, and thus no movement is required. Ammirato *et al*. [3] built a realistic dataset for active object instance classification [27]. Though involving movement, they have a similar setting to the aforementioned works. Our EAR task is more realistic and challenging – the agent is required to have both a smart movement strategy to control what to see, and a good visual recognition system to aggregate temporal information from multiple viewpoints.

Embodiment. Recently, a number of 3D simulators have been introduced to model virtual embodiment. Several of them are based on real-world environments [3, 4, 61, 64] for tasks such as robot navigation [4, 65] and scene understanding [5]. Other simulators have been built for synthetic environments [10, 37, 52]. They provide accurate labels for 3D objects and programmable interfaces for building various tasks, such as visual navigation [69] and embodied question answering [15, 16, 26]. EAR is a new task for these environments: unlike visual navigation, where the goal is to find objects or locations, our task assumes the target object is already (partially) observed at the beginning, and unlike question answering [15, 16, 26], we only focus on amodal recognition, which is arguably suited for benchmarking progress and diagnosing vision systems.

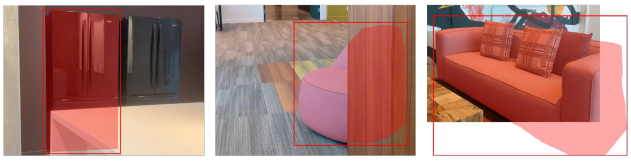

Figure 3: From left to right, we show the ground-truth annotation on RGB images. We show partially occluded objects and out-of-view object.

3. Embodied Amodal Recognition Dataset

Environment. Although EAR can be set up on any simulation environments [4, 37, 52], in this paper we use an indoor simulator as a demonstration. In those synthetically-generated indoor environments, there are objects in distinct categories. Similar to the EQA dataset [15], we filter out atypical 3D rooms that are either too big or have multiple levels, resulting in 550 houses in total. These houses are split to 400, 50, 100 for training, validation and test, respectively.

Rendering. Based on the indoor simulator, we render 640×800 images, and generate ground truth annotations for object category, amodal bounding boxes, and amodal masks. Previous work on amodal segmentation [19, 40, 70] made a design decision that clips amodal masks at image borders. This undermines the definition of amodal masks and was a limitation of using static images. Our work relies on a simulator, and thus we can easily generate amodal masks that extend beyond the image borders (see Fig. 3). In practice, we extend borders of rendered images by 80 pixels on each side (resulting in 800×960 images).

Objects. We select a subset of object categories that are suitable for us to study an agent's understanding for occluded objects. Our selection criteria are: 1) objects should have a sufficient number of appearances in the training data, 2) objects should have relatively rigid shapes without deformable structures (curtains, towels, *etc*.), ambiguous geometries (toys, paper, *etc*.), or being room components (floors, ceilings, *etc*.), and 3) if the object category label is coarse, we go one level deeper into the label hierarchy, and find a suitable sub-category (such as washer, *etc*.). As a result, there are 8 categories out of 80, including bed, chair, desk, dresser, fridge, sofa, table, and washer. In our dataset, there are 859/123/349 unique object instances (*i.e*., shapes) in the train/val/test set respectively, and 235 are shared by train and test sets.

Initial Location and Viewpoint. We first define the *visibility* of an object by the ratio between visible and amodal masks. Then, we randomly sample spawning locations and viewpoints for the agent as follows: 1) The agent should be spawned close to the object, between 3 to 6 meters away;

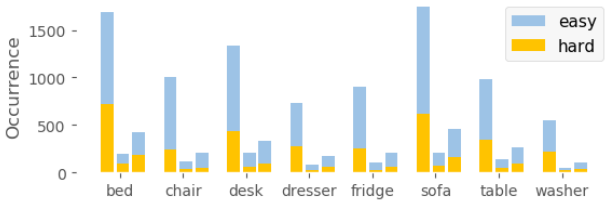

Figure 4: Object occurrences in our dataset. For each category, the three grouped bars represent train/validation/test sets; upper blue bars represent "easy" instances and bottom orange bars represent "hard" instances.

2) The object visibility should be no less than 0.2; and 3) At most 6 instances are sampled for each object category in one house. Finally, we obtain 8940 instances in training set, 1113 in validation set, and 2170 in test set. We also categorize spawning locations into "hard" instances if the object visibility is less than 0.5; otherwise "easy". In Fig. 3, from left to right, we visualize easy, hard, and partially out-of-view samples. The summary of object occurrences in Fig. 4 shows that our dataset is relatively balanced across different categories and difficulties.

Action Space. We configure our agent with two sets of primitive actions: moving and turning. For moving, we allow agents to *move forward, backward, left*, and *right* without changing viewing angle. For turning, we allow agents to *turn left* or *right* 2 degrees. This results in six actions in the action space. Note that we include *move backwards* in the action space because that agent might need to back track to remove occlusions.

Shortest Paths. Since EAR aims to learn to move around to recognize occluded objects better, it is *not* immediately clear what the "ground-truth" moving path is. This is different from other tasks, *e.g.* point navigation, where the shortest path can serve as an "oracle" proxy. Nevertheless, as shortest-path navigation allows the agent to move closer to the target object and likely gain a better view, we provide shortest-paths as part of our dataset, hoping it can provide both imitation supervision and a strong baseline.

4. Embodied Mask R-CNN

In this section, we propose a model called *Embodied Mask R-CNN* to address the Embodied Amodal Recognition. The proposed model consists of two modules, amodal recognition and action policy, as outlined in Fig. 2.

Before discussing the detailed designs, we first define the notation. The agent is spawned with an initial location and gaze described in the previous section. Its initial observation of the environment is denoted by I_0, and the task specifies a target object with a bounding box $\boldsymbol{b}_0$ encompassing the visible region. Given the target object, the agent moves in the 3D environment following an action policy π. At each step 0 to T, the agent takes action a_t based on π and observes an image I_t from a view angle $\boldsymbol{v}_t$. The agent outputs its prediction of the object category, amodal bounding box and mask, denoted by $\boldsymbol{y}_t = \{c_t, \boldsymbol{b}_t, \boldsymbol{m}_t\}$, for the target object in the first frame. The goal is to recover the true object category, amodal bounding box, and amodal segmentation mask, $\boldsymbol{y}_* = \{c^*, \boldsymbol{b}^*, \boldsymbol{m}^*\}$, at time step 0.

4.1. Amodal Recognition

The amodal recognition module is responsible for predicting the object category, amodal bounding box, and amodal mask at each navigational time step.

Mask R-CNN w/ Target Object. Our amodal recognition module has a similar goal as Mask R-CNN [28], so we followed the architecture design. In our task, since the agent is already provided with the visible location of target object in the first frame, we remove the region proposal network from Mask R-CNN and directly use the location box to feed into the second stage. In our implementation, we use ResNet-50 [29] pre-trained on ImageNet as the backbone.

Temporal Mask R-CNN. Given the sequential data $\{I_0, I_1, \ldots, I_t\}$ along the agent's trajectory, aggregating the information is challenging, especially when the 3D structure of the scene and the locations of the target object in the later frames are not known. To address this, we propose a model called Temporal Mask R-CNN to aggregate the temporal information from multiple frames, as shown in Fig. 5. Formally, the prediction of our Temporal Mask R-CNN at time step t is:

$$\boldsymbol{y}_t = f(\boldsymbol{b}_0, I_0, I_1, ..., I_t). \tag{1}$$

Our amodal recognition model has three components: $\{f_{\text{base}}, f_{\text{fuse}}, f_{\text{head}}\}$. For each frame I_t, we first use a convolutional neural network to extract a feature map $\boldsymbol{x}_t = f_{\text{base}}(I_t)$. Then, a feature aggregation function combines all the feature maps up to t, resulting in a fused feature map $\hat{\boldsymbol{x}}_t = f_{\text{fuse}}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t)$. For the feature aggregation f_{fuse}, we use a single-layer Convolutional Gated Recurrent Unit (Conv-GRU) [8, 14] to fuse temporal features. Besides Conv-GRU, we can also use simple temporal average or maximal pooling to fuse the features. These features are then sent to a Region-of-Interest (RoI) [23] head layer f_{head} to make predictions for the first frame:

$$\boldsymbol{y}_t = f_{\text{head}}(\boldsymbol{b}_0, \hat{\boldsymbol{x}}_t). \tag{2}$$

To train the model, we use image sequences generated from the shortest-path trajectory. The overall loss for our

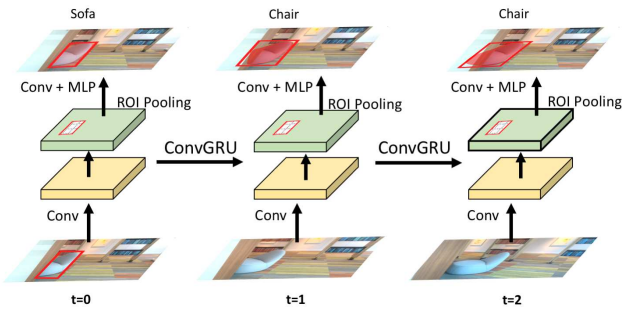

Figure 5: The amodal recognition part of Embodied Mask R-CNN. The agent moves in the environment, acquires different views in each step (bottom row), and updates the prediction for the target object of the first frame (top row).

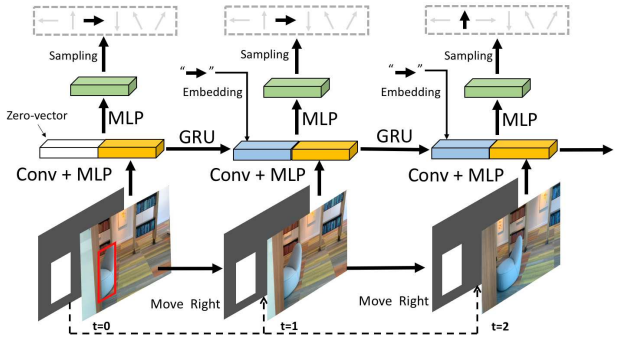

Figure 6: The action policy part of Embodied Mask R-CNN. At each step, the agent takes the current visual observation, last action, and initial visible bounding box of the target object as input, and predicts which action to take.

amodal recognition is defined as:

$$L^p = \frac{1}{T}\sum_{t=1}^{T}\Big[L_c^p(c_t, c^*) + L_b^p(\boldsymbol{b}_t, \boldsymbol{b}^*) + L_m^p(\boldsymbol{m}_t, \boldsymbol{m}^*)\Big], \tag{3}$$

where L_c^p is the cross-entropy loss, L_b^p is the smooth L1 regression loss, and L_m^p is the binary cross-entropy loss [28].

4.2. Learning to Move

The goal of the policy network is to propose the next moves in order to acquire useful information for amodal recognition. We disentangle it with the perception network, so that the learned policy will not over-fit to a specific perception model. We elaborate our design as follows.

Policy Network. Similar to the perception network, the policy network receives a visible bounding box of target object $\boldsymbol{b}_0$ and the raw images as inputs, and outputs probabilities over the action space. We sample actions at step t using:

$$a_t \sim \pi(\boldsymbol{b}_0, I_0, I_1, ...I_t). \tag{4}$$

As shown in Fig. 6, the policy network has three components $\{f_{\text{imgEnc}}, f_{\text{actEnc}}, f_{\text{act}}\}$. f_{imgEnc} is an encoder for image features. At step t, its inputs consist of I_0, I_t, as well as a mask I^b representing the visible bounding box of the target object $\boldsymbol{b}_0$ in the initial view. We concatenate those inputs, resize them to 320×384, and pass them to f_{imgEnc}, which consists of four $\{5\times5$ Conv, BatchNorm, ReLU, 2×2 MaxPool$\}$ blocks [15], producing an encoded image feature: $\boldsymbol{z}_t^{\text{img}} = f_{\text{imgEnc}}\Big([I^b, I_0, I_t]\Big)$.

Besides image features, we also encode the last action in each step t. We use a multi-layer perceptron (MLP) network f_{actEnc} to get the action feature $\boldsymbol{z}_t^{\text{act}} = f_{\text{actEnc}}(a_{t-1})$. We then concatenate $\boldsymbol{z}_t^{\text{act}}$ and $\boldsymbol{z}_t^{\text{img}}$, and pass the result to a single-layer GRU network f_{act} for integrating history information:

$$z_t = f_{\text{act}}\Big([\boldsymbol{z}_t^{\text{img}}, \boldsymbol{z}_t^{\text{act}}], h_{t-1}\Big), \tag{5}$$

where h_{t-1} is the hidden state from last step. z_t is then sent to a linear layer with softmax to derive the probability distribution over the action space, from which the action a_t is sampled. We learn $\{f_{\text{imgEnc}}, f_{\text{actEnc}}, f_{\text{act}}\}$ via reinforcement learning. We now describe how we design the reward.

Rewards. Our goal is to find a good strategy for the agent to move to improve its amodal recognition performance. We directly use the classification accuracy and Intersection-over-Union (IoU) to measure the advantages of candidate agent moves. Specifically, at each step t, we obtain the prediction of amodal recognition $\boldsymbol{y}_t$, and then compute the classification accuracy Acc_t^c (1 if correct, otherwise 0) and IoU between the amodal bounding box IoU_t^b and mask IoU_t^m. Due to the different scales of these three rewards, we compute a weighted sum and then use reward shaping:

$$r_t = \lambda_c Acc_t^c + \lambda_b IoU_t^b + \lambda_b IoU_t^m, \tag{6}$$

$$R_t = r_t - r_{t-1}, \tag{7}$$

where λ_c=0.1, λ_b=10 and λ_m=20. To learn the policy network, we use policy gradient with REINFORCE [56].

4.3. Staged Training

We observe that joint training of the perception and policy networks from scratch struggles because the perception model cannot provide a correct reward to the policy network, and the policy network cannot take reasonable actions in turn. We thus resort to a staged training strategy. Namely, we first train the perception network with frames collected from the shortest path. Then, we plug in the pre-trained perception network to train the policy network with the perception part fixed. Finally, we retrain the perception network so that it can adapt to the learned action policy.

5. Experiments

5.1. Metrics and Baselines

Metrics. Recall that we evaluate the amodal recognition performance on the first frame in the moving path. We report object classification accuracy (Clss-Acc), the IoU scores for amodal box (ABox-IoU) and amodal mask (AMask-IoU). We additionally evaluate the performance of amodal segmentation *only* on the occluded region of the target object (AMask-Occ-IoU).

Baselines. We conduct extensive comparisons against a number of baselines. We use the format Training/Testing moving paths to characterize the baselines:

- *Passive/Passive (PP/PP)*: This is the passive amodal recognition setting, where the agent does not move during training and testing. Comparisons to this baseline establishes the benefit of embodiment.
- *ShortestPath/Passive (SP/PP)*: The agent moves along the shortest path for training amodal recognition, but the agent does not move during testing. We use this baseline to understand how much improvement is due to additional unlabeled data.
- *ShortestPath/Passive* (SP/PP*)*: Training is the same as above; In testing, the agent does not move, but we create a sequence of static observations by replicating the initial frame and feed them to the model. This baseline determines whether the improvement is due to the effectiveness of the recurrent network.
- *ShortestPath/RandomPath (SP/RP)*: The agent moves randomly during test. This baseline establishes whether strategic moves are required for embodied amodal recognition. We report the performance by taking the average scores of five random tests.
- *ShortestPath/ShortestPath (SP/SP)*: The agent moves along the shortest path during both training and testing. This is an "oracle-like" baseline, because in order to construct shortest-path, the agent need to know the entire 3D structure of the scene. However, there is no guarantee that this is an optimal path for recognition.

We compare these baselines with our two final models: *ShortestPath/ActivePath (SP/AP)* and *ActivePath/ActivePath (AP/AP)*. For *ShortestPath/ActivePath*, we train the amodal recognition model using frames in shortest path trajectories, and then train our action policy. For *ActivePath/ActivePath*, we further fine-tune our amodal recognition model based on the learned action policy.

Noticeably, all the above models use the same temporal Mask R-CNN architecture for amodal recognition. For single-frame prediction, the GRU module is also present. Moreover, all of those models are trained using the same amount of supervision and then evaluated on the same test set for fair comparison.

5.2. Implementation Details

Here we provide the implementation details of our full system *ActivePath/ActivePath*. There are three stages:

Stage 1: training amodal recognition. We implement our amodal recognition model, Temporal Mask R-CNN, based on the PyTorch implementation of Mask R-CNN [44]. We use ResNet50 [29] pre-trained from ImageNet [51] as the backbone and crop RoI features with a C4 head [49]. The first three residual blocks in the backbone are fixed during training. We use stochastic gradient descent (SGD) with learning rate 0.0025, batch size 8, momentum 0.99, and weight decay 0.0005.

Stage 2: training action policy. We fix the amodal recognition model, and train the action policy *from scratch*. We used RMSProp [30] as the optimizer with initial learning rate 0.00004, and set ϵ=0.00005. In all our experiments, the agent moves 10 steps in total.

Stage 3: fine-tuning amodal recognition. Based on the learned action policy, we fine-tune the amodal recognition model, so that it can adapt to the learned moving path. We use SGD, with learning rate 0.0005.

5.3. General Analysis on Experimental Results

In Table 1, we show the quantitative comparison of amodal recognition performance for different models. We report the numbers on all examples from the test set ('all'), the easy examples (visibility > 0.5), and hard examples (visibility $\leqslant 0.5$). We have the following observations.

Shortest path move does not help passive amodal recognition. As shown in Table 1, both *ShortestPath/Passive* and *ShortestPath/Passive** are slightly inferior to *Passive/Passive*. Due to the movement, the visual appearance of additional images might change a lot compared with the first frame. As such, these extra inputs do not appear to serve as effective data augmentation for training amodal recognition in passive vision systems.

Embodiment helps amodal recognition. In Table 1, we find that agents that move at test time (bottom four rows) consistently outperform agents that stay still (first three rows). Interestingly, *even moving randomly* at test time (*ShortestPath/RandomPath*), the agent still outperforms the passive one. This provides evidence that this embodied paradigm helps amodal recognition and the proposed Embodied Mask R-CNN model is effective for EAR.

Our model learns a better moving policy. In Table 1, we compare the models with embodiment (bottom four rows). The shortest path is derived to guide the agent to move *close* to the target object. It may not be the optimal moving policy for EAR, since the task does not necessarily require the agent to get close to the target object. In

Moving Path		Clss-Acc			ABox-IoU			AMask-IoU			AMask-Occ-IoU		
Train	Test	all	easy	hard	all	easy	hard	all	easy	hard	all	easy	hard
Passive	Passive	92.9	94.1	90.9	81.3	83.9	76.5	67.6	69.6	63.9	49.0	46.0	54.6
ShortestPath	Passive	92.8	94.3	89.9	81.2	83.8	76.4	67.4	69.6	63.4	48.6	45.8	54.1
ShortestPath	Passive*	93.0	94.3	90.7	80.9	83.1	76.8	66.7	68.4	63.6	48.4	44.9	54.9
ShortestPath	RandomPath	93.1	94.1	91.1	81.6	83.9	77.1	67.8	69.7	64.3	49.0	45.8	55.2
ShortestPath	ShortestPath	93.2	94.1	91.7	82.0	84.3	77.7	68.6	70.4	65.3	**50.2**	**46.9**	56.3
ShortestPath	ActivePath	93.3	93.9	**92.2**	82.0	**84.4**	77.6	**68.8**	**70.5**	65.5	**50.2**	**46.9**	56.4
ActivePath	ActivePath	**93.7**	**94.6**	**92.2**	**82.2**	84.3	**78.2**	68.7	70.3	**65.6**	**50.2**	46.8	**56.7**

Table 1: Quantitative comparison of amodal recognition using different models. "Train" denotes the source of moving path used to train the perception model; "Test" denotes the moving path in the testing stage. We report the performance at the last (10-th) action step for embodied agents.

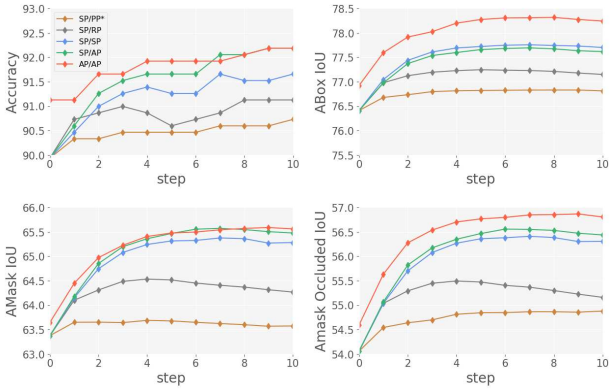

Figure 7: Performance of different models on hard samples over action step on four metrics.

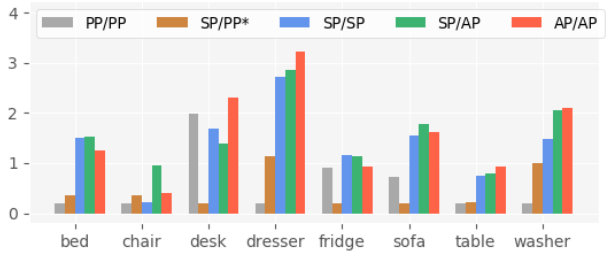

Figure 8: Performance on each object category for all methods. For each category, we take average over the first three metrics for each method and truncate them by the number from worst method to show the relative improvements.

contrast, our model learns a moving policy to improve the agent's amodal recognition ability. Though using the same amodal recognition model, *ShortestPath/ActivePath* finds a better moving policy, and the performance is on par or slightly better than *ShortestPath/ShortestPath*. After fine-

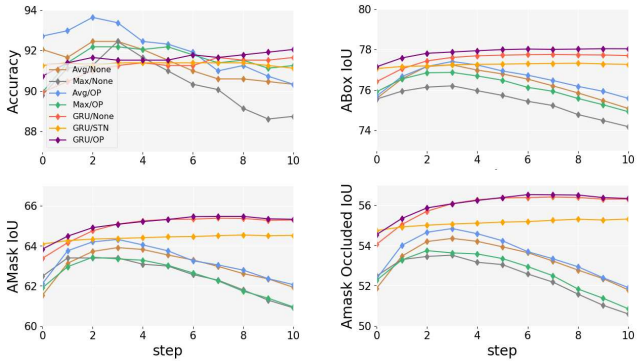

Figure 9: Performance for different feature aggregation/warping methods on hard samples over action step.

tuning the amodal recognition model using the learned path, (ActivePath/ActivePath) achieves further improvement by adapting the amodal recognition model to the learned paths.

5.4. Analysis on Amodal Recognition

Objects with different occlusions. In Table 1, we observe that agents with embodiment in general achieve more improvement on "hard" samples compared with "easy" samples. For example, the object classification accuracy of *ActivePath/ActivePath* is 0.5% higher than *Passive/Passive* for "easy" samples, while 1.3% higher for "hard" samples. In general, objects with heavy occlusions are more difficult to recognize from a single viewpoint, and embodiment helps because it can recover the occluded object portions.

Improvements over action step. We show amodal recognition performance along the action step in Fig. 7 on hard samples. In general, the performance improves as more steps are taken and more information aggregated, but eventually saturates. We suspect that the agent's location and viewpoint might change a lot after a number of steps, it be-

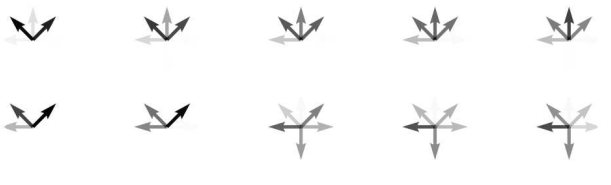

Figure 10: Distribution of actions at step 1, 3, 5, 7, 10 on test set. ↑: Forward, ↓: Backward, ←: Move left, →: Move right, ↖: Rotate left, ↗: Rotate right. Top row: shortest-path movement. Bottom row: our learned policy. Darker color denotes more frequent actions.

comes more challenging to aggregate information.

Performances on different object categories. In Fig. 8, we plot the relative improvements for different models on different object categories (we add a small constant value to each in the visualization for clarity). For comparison, we compute the average of the first three metrics for each category and all samples. The improvement is more significant on categories such as bed, dresser, sofa, table, and washer.

Other feature aggregation and warping methods. We investigate other feature aggregation and warping methods here. For feature aggregation in the amodal recognition module, we replace GRU with simple Max/Average pooling. To warp the features, we extract optical flow using [55] (*OP*) and then warp the features from future frames to the first frame. Moreover, we use a Spatial Transformer Network (*STN*) to learn to warp the features. The comparisons are shown in Fig. 9. As we can see, Max/Average pooling methods cannot further aggregate useful information after three steps; merely warping features also does not work well. However, combining GRU with feature warping (*GRU/OP*) does further improve the performance.

5.5. Analysis on the Learned Policy

Using the learned moving paths, agents can predict better amodal masks compared with shortest path, and their moving patterns are also different.

Comparing moving strategies. Fig. 10 shows the distribution of actions at steps 1, 3, 5, 7 and 10 for the shortest path and our learned path. We can observe that different moving strategies are learned from our model compared with shortest path even though the amodal recognition model is shared by two models. Specifically, our agent rarely moves forward. Instead, it learns to occasionally move backward. This can be beneficial in cases where the agent is spawned close to the target, and moving backward can reveal more content of the object. This comparison indicates the shortest path may not be the optimal path for EAR. As shown in Fig. 11, under the shortest path, the agent gets closer to the object. However, our learned moves keep the distance nearly constant to the target object. Under this moving policy, the viewed-size of the target object at each step does not change too drastically.

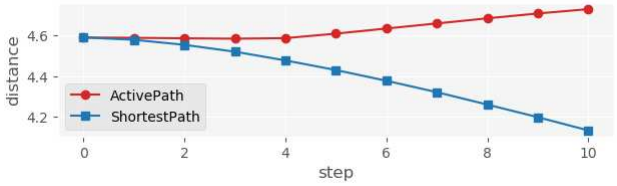

Figure 11: Distance to target objects at each step averaged over the test set for shortest path and our learned path.

6. Conclusion

In this work, we introduced a new task called *Embodied Amodal Recognition* — an agent is spawned in a 3D environment, and is free to move in order to perform object classification, amodal localization and segmentation of a target occluded object. As a first step toward solving this task, we proposed an *Embodied Mask R-CNN* model that learned to move strategically to improve the visual recognition performance. Through comparisons with various baselines, we demonstrated the importance of embodiment for visual recognition. We also show that our agents developed strategic movements that were different from shortest path, to recover the semantics and shape of occluded objects.

Acknowledgments. We thank Lixing Liu, Manolis Savva, Marcus Rohrbach and Dipendra Misra for helpful discussions. Georgia Institute of Technology and Indiana University's efforts in this research was supported in part by NSF, AFRL, DARPA, ONR YIPs, ARO PECASE, the IU Office of the Vice Provost for Research, the IU College of Arts and Sciences, and the IU School of Informatics, Computing, and Engineering through the Emerging Areas of Research Project "Learning: Brains, Machines, and Children." The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 37–45, 2015. 3

[2] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision (IJCV)*, 1988. 3

[3] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg. A dataset for devel-

oping and benchmarking active vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 3

[4] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[5] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 3

[6] Renee Baillargeon, Elizabeth S Spelke, and Stanley Wasserman. Object permanence in five-month-old infants. *Cognition*, 20(3):191–208, 1985. 1

[7] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 1988. 3

[8] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *International Conference on Learning Representations (ICLR)*, 2016. 4

[9] Sven Bambach, David J. Crandall, Linda B. Smith, and Chen Yu. Toddler-inspired visual object learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 3

[10] Simon Brodeur, Ethan Perez, Ankesh Anand, Florian Golemo, Luca Celotti, Florian Strub, Jean Rouat, Hugo Larochelle, and Aaron Courville. HoME: A household multimodal environment. *arXiv preprint arXiv:1711.11017*, 2017. 3

[11] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 3

[12] Ricson Cheng, Arpit Agarwal, and Katerina Fragkiadaki. Reinforcement learning of active vision for manipulating objects under occlusions. In *Conference on Robot Learning (CoRL)*, 2018. 3

[13] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision*, pages 628–644. Springer, 2016. 3

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[15] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3, 5

[16] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural Modular Control for Embodied Question Answering. In *Conference on Robot Learning (CoRL)*, 2018. 3

[17] Joachim Denzler and Christopher M Brown. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2002. 3

[18] Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. The MIT Press, 1999. 1

[19] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3

[20] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 3

[21] Patrick Follmann, Rebecca König, Philipp Härtinger, and Michael Klostermann. Learning to see the invisible: End-to-end trainable amodal instance segmentation. 2019. 3

[22] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations (ICLR)*, 2019. 1

[23] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1, 3, 4

[24] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1, 3

[25] Abel Gonzalez-Garcia, Alexander Vezhnevets, and Vittorio Ferrari. An active search strategy for efficient object class detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 3

[26] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali

Farhadi. IQA: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[27] Xiaoning Han, Huaping Liu, Fuchun Sun, and Xinyu Zhang. Active object detection with multi-step action prediction using deep q-network. *IEEE Transactions on Industrial Informatics*, 2019. 3

[28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2, 3, 4, 5

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 3, 4, 6

[30] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012. 6

[31] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[32] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 3

[33] Dinesh Jayaraman and Kristen Grauman. End-to-end policy learning for active visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 3

[34] Edward Johns, Stefan Leutenegger, and Andrew J Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[35] Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*. Praeger, 1979. 1

[36] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Amodal completion and size constancy in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[37] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017. 3

[38] Danica Kragic, Mårten Björkman, Henrik I Christensen, and Jan-Olof Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and autonomous Systems*, 2005. 3

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 1, 3

[40] Ke Li and Jitendra Malik. Amodal instance segmentation. In *Proceedings of the European Conference on Computer Vision*, 2016. 1, 3

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 3

[42] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 3

[43] Mohsen Malmir, Karan Sikka, Deborah Forster, Javier Movellan, and Garrison W Cottrell. Deep q-learning for active recognition of germs: Baseline performance on a standardized dataset for active learning. *Proceedings of the British Machine Vision Conference (BMVC)*, 2015. 3

[44] Francisco Massa and Ross Girshick. maskrnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. 6

[45] 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*, 2016. 3

[46] Deepak Pathak, Yide Shentu, Dian Chen, Pulkit Agrawal, Trevor Darrell, Sergey Levine, and Jitendra Malik. Learning instance segmentation by interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 3

[47] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 3

[48] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 1, 3, 6

[50] Zhile Ren and Erik B Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1525–1533, 2016. 3

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 3, 6

[52] Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 3

[53] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. Multi-layer depth and epipolar feature transformers for 3D scene reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 3

[54] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. 3

[55] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 8

[56] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998. 5

[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 3

[58] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3D scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[59] Hsiao-Yu Fish Tung, Ricson Cheng, and Katerina Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2595–2603, 2019. 3

[60] Johan Wagemans, James H Elder, Michael Kubovy, Stephen E Palmer, Mary A Peterson, Manish Singh, and Rüdiger von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin*, 2012. 1

[61] Marcus Wallenberg and Per-Erik Forssén. A research platform for embodied visual object recognition. In *SSBA 2010, Uppsala, Sweden, 11-12 March 2010*, pages 137–140, 2010. 3

[62] David Wilkes and John K Tsotsos. Active object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1992. 3

[63] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 3

[64] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[65] Xin Ye, Zhe Lin, Haoxiang Li, Shibin Zheng, and Yezhou Yang. Active object perceiver: Recognition-guided policy learning for object searching on mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 3

[66] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Learning Representations (ICLR)*, 2015. 1, 3

[67] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 3

[68] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 3

[69] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 3

[70] Yan Zhu, Yuandong Tian, Dimitris Mexatas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 3