Embodied Visual Recognition

Jianwei Yang[1⋆], Zhile Ren[1⋆], Mingze Xu[2],
Xinlei Chen[3], David Crandall[2], Devi Parikh[1,3] Dhruv Batra[1,3]
[1]Georgia Institute of Technology, [2]Indiana University, [3]Facebook AI Research.
https://www.cc.gatech.edu/~jyang375/evr.html

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 Visual Recognition (EVR): 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, 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 the House3D environment. Experimental results show that: 1) agents with embodiment (movement) achieve better visual recognition performance than passive ones; 2) in order to improve visual recognition abilities, agents can learn strategical moving paths that are different from shortest paths.

1. Introduction

Recently, visual recognition tasks such as image classification [30, 32, 41, 59], object detection [23, 24, 50–52] and semantic segmentation [44, 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 [19, 37, 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 estimations of physical dimensions [38]. More fundamentally, it helps agents to understand object permanence, that is, objects have extents and do not cease to exist when they are occluded [6].

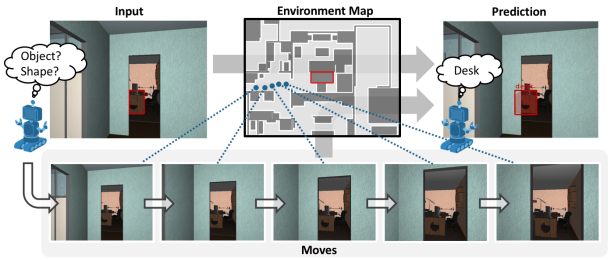

Figure 1: The task of Embodied Visual Recognition: An agent is spawned close to an occluded target object in a 3D environment, and asked for visual recognition, *i.e.*, predicting class label, amodal bounding box and amodal mask of the target object. The agent is free to move around to aggregate information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

Recently, the dominant paradigm for object recognition and amodal perception has been based on single image. Though leveraging the advances of deep learning, visual systems still fail to recognize object and its shape from single 2D image in the presence of heavy occlusions. Consider amodal perception. Existing works ask the model to implicitly learn the 3D shape of the object *and* the projection of that shape back into the image [20, 42, 70]. This is an entangled task, and deep models are thus prone to over-fit to subtle biases in the dataset [22] (*e.g.* learning that beds always extend leftwards into the frame).

Remarkably, humans have the visual recognition ability to infer both semantics and shape for the occluded objects from a single image. On the other hand, humans also have the ability to derive strategical moves to gather more information from new viewpoints to further help the 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 well. As shown in Fig. 1, to recognize the category and whole shape of target object indicated by the red bounding box, agents should learn to actively move

⋆The first two authors contributed equally.

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

In this paper, we introduce a new task called *Embodied Visual Recognition* (EVR) where agents actively move in a 3D environment for visual recognition of a target object. We are aimed at systemically studying whether and how embodiment (movement) helps visual recognition. Though it is a general paradigm, we highlight three design choices for the EVR task in this paper:

Three sub-tasks. In EVR, we aim to recover both semantics and shape for the target object. It consists of three sub-tasks: object recognition, 2D amodal perception (amodal localization and 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, the 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 visual recognition for the target object observed at the spawning point. If the agent does not move, EVR degrades to a passive visual recognition. Both passive and embodied algorithms are trained using the same amount of supervisions 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 a general pipeline for EVR as shown in Fig. 2. Compared with the passive visual recognition model (Fig. 2a), the embodied agent (Fig. 2b) will follow the proposed action from the policy module to move in the environment, and make the predictions on the target object using the visual recognition module. This pipeline introduces several interesting problems: 1) Due to agent's movement, the appearances of observed scene *and* target object change in each step. How should information be aggregated from future frames to the first frame for visual recognition? 2) There is no "expert" that can tell the agent how to move in order to improve its visual recognition performance. How to effectively propose a strategic move without any supervision? 3) In this task, the perception module and action policy are both learned from scratch. Considering the performance of each heavily relies on the competence of the other, how to design proper training regime is also an open question.

To address the above questions, we propose a new model called *Embodied Mask R-CNN*. The perception module extends Mask R-CNN [29] 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 scheme to train these two modules effectively.

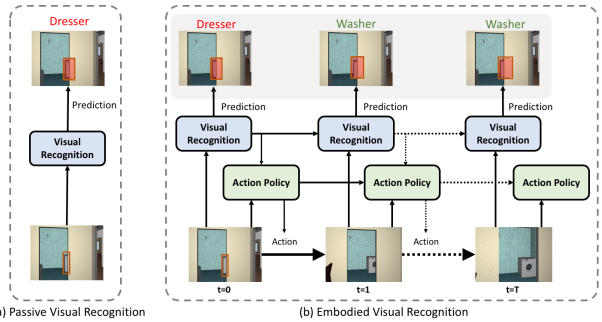

Figure 2: The proposed general pipeline for Embodied Visual Recognition task. To perform visual recognition (object recognition and amodal perception) on the occluded object, the agent *learns to move* (right), rather than standing still and *hallucinating* (left). The visual recognition module focuses on predicting the object class, amodal bounding box and masks for the first frame. The policy module proposes the 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 Visual 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 EVR. Using the House3D simulator [62] on SUNCG [55], 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 EVR and propose a new model, Embodied Mask R-CNN, to learn to move for visual recognition. In this model, the visual recognition and policy module make predictions at each step, and aim to improve the visual recognition performance on the target object in the first frame.
- We evaluate both passive and embodied vision recognition systems, and demonstrate that agents with movements consistently outperform passive ones. Moreover, the learned moves are more effective in improving visual 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

Visual Recognition. Building visual recognition systems is one of the long-term goals of our community. Training on large-scale datasets [43, 53, 68], we have recently

witnessed the versatility and effectiveness of deep neural networks for many tasks, including image classification [30, 32, 41, 59], object detection [23, 24, 50–52], semantic segmentation [44, 66, 67], instance segmentation [29] *etc*. Extending its successful story, similar pipelines have been applied to *amodal perception* as well, notably for amodal segmentation [20, 21, 42, 70].

Despite these advances, visual systems still fail to recognize objects from single 2D images in the presence of significant occlusion and unusual poses. Some work has attempted to overcome this by aggregating multiple frames or views [11, 48, 56, 64], other leveraging CAD models [5, 10, 27, 33, 57]. However, a diverse set of viewpoints or CAD model is not always available *a priori*, and unlikely to hold in practice. We would like to build the capability of agents to move around and change viewing angle in order to perceive. This is the goal of *active vision*.

Active Vision. Active vision has a long history of research [1, 7, 61], and also has connections to developmental psychology [9]. Recent work learns active strategy for object recognition [18, 34–36, 40, 45], object localization/detection [13, 25, 47], object manipulation [14] and instance segmentation [49]. 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*. [2] built a realistic dataset for active object instance classification [28]. Though involving movement, they have a similar setting to the aforementioned works, *i.e*., searching for a good viewpoint for instance classification by assuming the bounding boxes of the target object are known during the whole movement. In contrast, the formulation of EVR is more realistic and challenging – we allow an embodied agent in a 3D simulator to actively move and perform visual recognition. The agent is required to have both a smart moving strategy to control what visual input to receive, 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 [2, 3, 63] for tasks such as robot navigation [3, 65] and scene understanding [4]. Other simulators have been built for synthetic environments [12, 39, 54], such as House3D [62]. They come in handy with accurate labels for 3D objects and programmable interfaces for building various tasks, such as visual navigation [69] and embodied question answering [16, 17, 26]. EVR is a new exploration in the task space on 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;

Figure 3: Example annotations in our dataset. In each column, the top row shows the ground-truth annotation for the image, and the bottom row shows the corresponding agent's location and viewpoint (blue arrow) and target object's location (red bounding box) in top-view map. From left to right, we show an easy, a hard and a partially out-of-view sample, which is not included in previous amodal datasets [42, 70].

Unlike question answering [16, 17, 26], we only focus on visual recognition and is arguably better suited for benchmarking progress and diagnosing vision systems.

3. Dataset for EVR

Environment. Although EVR can be set up on any simulation environments [3, 39, 54], in this paper we use House3D [62] as a demonstration. House3D is an open-sourced simulator built on top of SUNCG [55], which contains objects from 80 distinct categories. Similar to the EQA dataset [16], we filter out atypical 3D rooms in House3D that are either too big or have multiple levels, resulting in 550 houses in total. A detailed list of houses can be found in the Appendix. These houses are split to 400, 50, 100 for training, validation and test, respectively.

Rendering. Based on the House3D 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 [20, 42, 70] made a design decision that clips amodal object masks at image borders. We believe 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 the right-most example in 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 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 and not have deformable structures (curtains, towels, *etc*.), ambiguous geometries (toys, paper, *etc*.), or be room components (floors,

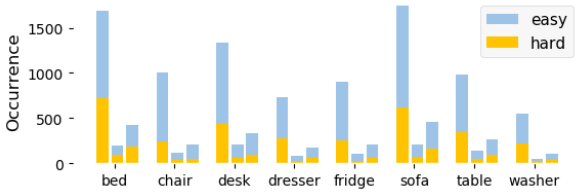

Figure 4: Object occurrences in our dataset. For each category, the three grouped bars represent train/validation/test sets; Upper blue bars represent "easy" instances and bottom orange bars represent "hard" instances.

ceilings, *etc*.), and 3) if the object category label is coarse, we go one level deeper into the label hierarchy in SUNCG, and find a suitable sub-category (such as washer, *etc*.). These criteria lead to 8 categories out of 80, including bed, chair, desk, dresser, fridge, sofa, table and washer.

Initial Location and Viewpoint. We first define the *visibility* of object by the ratio between visible and amodal masks. Then, we randomly sampling spawning locations and viewpoints for the agent by following: 1) The agent should be spawned close to the object, within distances between 3 to 6 meters; 2) The object visibility should be no less than 0.2; 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, each column shows an example ground-truth annotation (top) and the agent's initial location, viewpoint and the target's location (bottom). From left to right, they are easy, hard and partially out-of-view samples. In Fig. 4, we further provide a summary of object occurrences in our dataset. It 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* for 2 degrees. This results in six actions in the action space. Note that we include *move backward* in the action space considering that agent might need to backward to get rid of the occlusions.

Shortest Paths. Since EVR 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 still 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 Visual Recognition. The proposed model consists of two modules, visual recognition module and action module, as outlined in Fig. 2.

Before discussing the detailed designs, we first define the notations. The agent is spawned with 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. Visual Recognition

The visual 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 visual recognition module has a similar goal as Mask R-CNN [29], 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 [30] pre-trianed on ImageNet as the backbone.

Temporal Mask R-CNN. Given the sequential data along agent's trajectory, Temporal Mask R-CNN aims at aggregating temporal information from multiple frames to obtain more accurate predictions. Formally, the prediction of our temporal Mask R-CNN at time step t is:

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

The challenge is how to aggregate information from $\{I_0, I_1, \ldots, I_t\}$ together, especially when the 3D structure of the scene and the locations of the target object in the later frames are not unknown. To address this problem, we use feature-level fusion.

Our perception 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 map 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 model f_{fuse}, we use a single-layer Recurrent Convolution

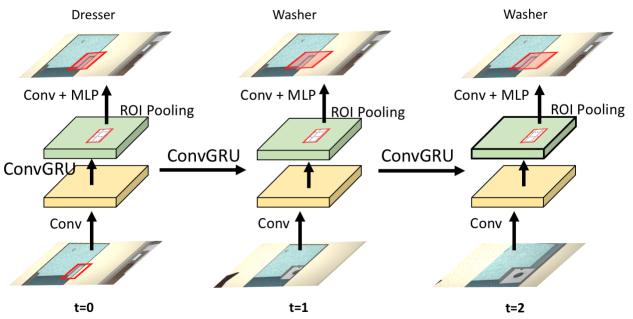

Figure 5: The visual recognition part of Embodied Mask R-CNN. The agent moves in the environment, acquires different views in each step (bottom row), and updates the prediction for the target object of the first frame (top row).

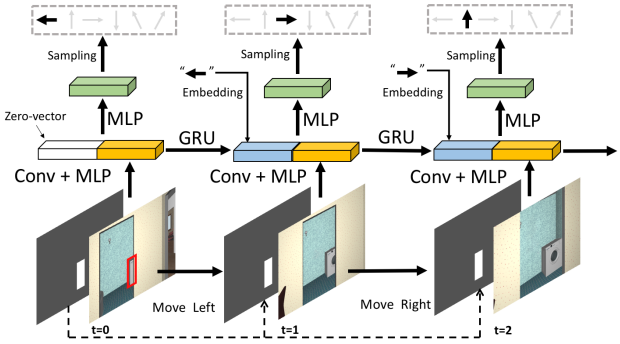

Figure 6: The action policy part of Embodied Mask R-CNN. At each step, the agent takes the current visual observation, last action and initial visible bounding box of target object as input, and predicts which action to take.

Network with Gated Recurrent Unit (GRU-RCN) [8, 15] to fuse temporal 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 visual 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 [29].

4.2. Learning to Move

The goal of the policy network is to propose the next moves in order to acquire useful information for visual 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:

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

The policy network has three components $\{\pi_{\text{imgEnc}}, \pi_{\text{actEnc}}, \pi_{\text{act}}\}$. π_{imgEnc} is an encoder for image features. The inputs 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 π_{imgEnc}, which consists of four $\{5\times5$ Conv, BatchNorm, ReLU, 2×2 MaxPool$\}$ blocks [16], producing an encoded image feature: $\boldsymbol{z}_t^{\text{img}} = \pi_{\text{imgEnc}}\Big([I^b, I_0, I_t]\Big)$.

Besides image features, we also encode the last action in each step t. We use a multi-layer embedding network π_{actEnc}, which consists of an embedding layer, producing an encoded action feature $\boldsymbol{z}_t^{\text{act}} = \pi_{\text{actEnc}}(a_{t-1})$. We concatenate $\boldsymbol{z}_t^{\text{act}}$ and $\boldsymbol{z}_t^{\text{img}}$, and pass it to a single-layer GRU network π_{act}, whose output is sent to a linear layer with SoftMax to obtain a probability distribution over the action space:

$$a_t \sim \pi_{\text{act}}\Big([\boldsymbol{z}_t^{\text{img}}, \boldsymbol{z}_t^{\text{act}}]\Big). \tag{5}$$

We learn $\{\pi_{\text{imgEnc}}, \pi_{\text{actEnc}}, \pi_{\text{act}}\}$ 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 visual 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 visual 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 perform 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 π, we use policy gradient with REINFORCE [58].

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 an 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 visual recognition performance on the first frame in the moving path. We report object classification accuracy (Clss-Acc), and 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 conventional passive visual recognition setting, where the agent does not move during training and testing. The comparison to this baseline establishes the benefit of embodiment.
- *ShortestPath/Passive (SP/PP)*: The agent moves along the shortest path for training visual 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 replicate the initial frame to create fake observations along the moving path to feed 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 move is required for embodied visual recognition. We report the performance by taking the average of 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 visual recognition model using frames in shortest path trajectories, and then train our own action policy. For *ActivePath/ActivePath*, we further fine-tune our visual recognition model using rendered images generated from the learned action policy.

Noticeably, all the above models use the same temporal Mask R-CNN architecture for visual 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. For simplicity, we use the short name to represent each method in the figures.

5.2. Implementation Details

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

Stage 1: training visual recognition. We implement our visual recognition model, Temporal Mask R-CNN, based on the PyTorch implementation of Mask R-CNN [46]. We use ResNet50 [30] pre-trained from ImageNet [53] as the backbone and crop RoI features with a C4 head [52]. 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 visual recognition model, and train the action policy *from scratch*. We used RMSProp [31] 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 visual recognition. Based on the learned action policy, we fine-tune the visual 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 visual 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 visual 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 does not appear to serve as effective data augmentation for training visual recognition in passive vision systems.

Embodiment helps visual recognition. In Table 1, we can 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 passive one. This provides evidence for this embod-

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 visual 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 last (10-th) action step for embodied agents.

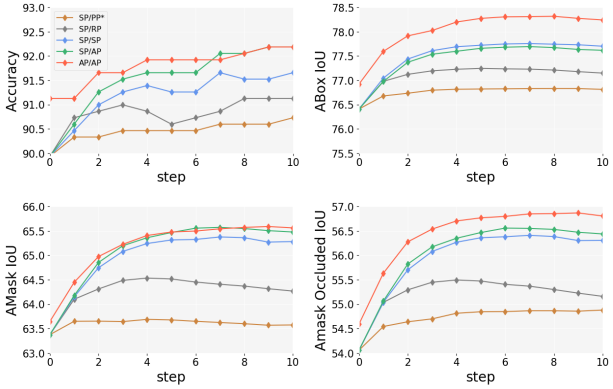

Figure 7: Performance of different models on hard samples over action step on four metrics.

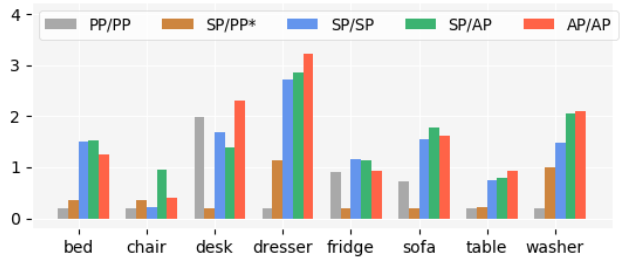

Figure 8: Relative comparison on different object categories for different methods. The numbers are obtained by taking the average of the first three metrics.

ied paradigm helps visual recognition and the proposed Embodied Mask R-CNN model is effective for EVR.

Our model learns a better moving strategy. In Table 1, we compare the models with embodiment (bottom four rows). The shortest path is derived to guide the agent move *close* to the target object. It may not be the optimal moving strategy for EVR, since the task does not necessarily require the agent to get close to the target object. In contrast, our model learns a moving strategy to improve the agent's visual recognition ability. Though using the same visual recognition model, *ShortestPath/ActivePath* finds a better moving strategy, and the performance is on par or slightly better than *ShortestPath/ShortestPath*. After fine-tuning the visual recognition model using the learned path, (ActivePath/ActivePath) achieves further improvement by adapting the visual recognition model to the learned paths.

5.4. Analysis on Visual 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 single viewpoint, and embodiment helps because it can recover the occluded object portions.

Improvements over action step. We show visual 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 much after a number of steps, it becomes more challenging for it 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.

5.5. Analysis on the Learned Policy

We visualize some example moving paths executed by learned policy (*ActivePath/ActivePath*) in Fig. 9. Using

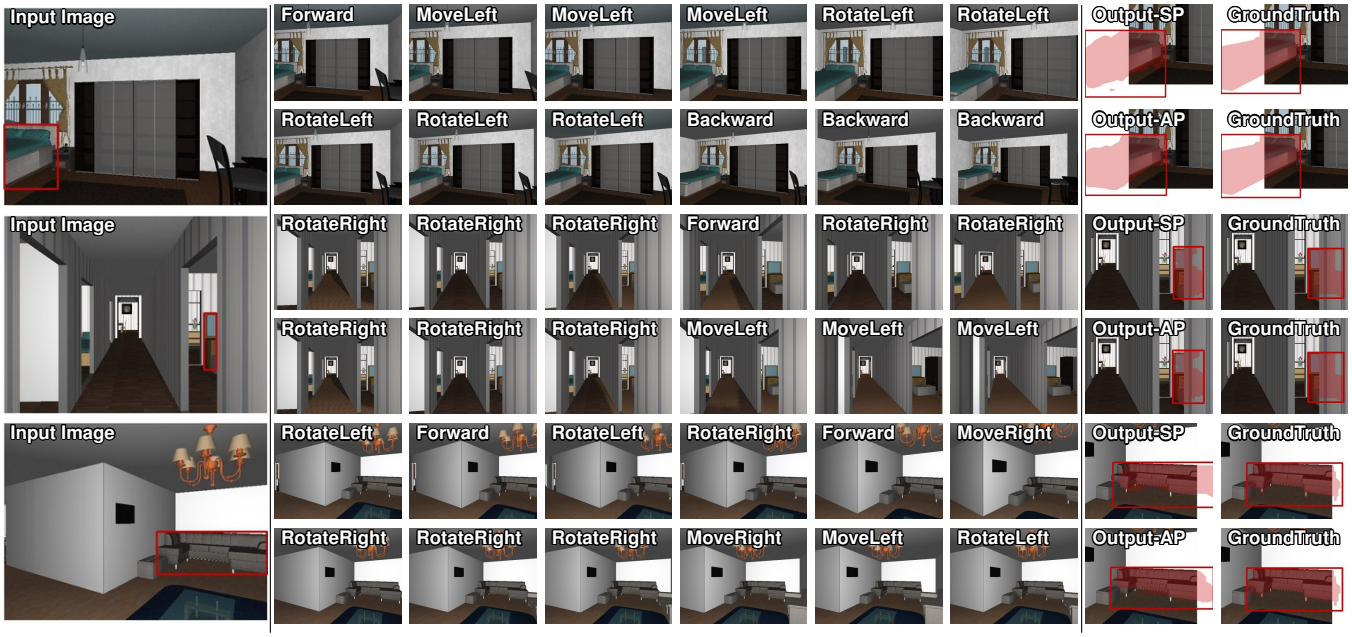

Figure 9: For each image, we visualize the shortest-path trajectory (top) and the learned active perception trajectory (bottom) at step 1, 3, 4, 6, 8, 10. Different from agents using the shortest-path move, our agents actively perceive the target object, and achieve better visual recognition performance.

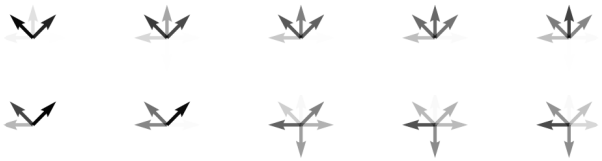

Figure 10: Distribution of actions at step 1, 3, 5, 7, 10 on test set. ↑: Forward, ↓: Backward, ←: Move left, →: Move right, ↖: Rotate left, ↗: Rotate right. Top row: shortest-path movement. Bottom row: our learned policy. Darker color denotes more frequent actions.

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 different moving strategies are learned from our model compared with shortest path even though the visual recognition model is shared by two models. Specifically, our agent rarely moves forward. Instead, it learns to occasionally move backward. This comparison indicates the shortest path may not be the optimal path for EVR.

Distance to the target object. We further investigate the moving strategy in terms of the distance to the target object. As shown in Fig. 11, under the shortest path, the agent gets closer to the object. However, our learned moves keep

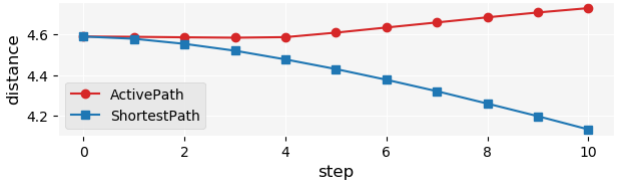

Figure 11: Distance to target objects at each step averaged on test set for shortest path and our learned path.

the distance nearly constant to the target object. Under this moving strategy, the viewed-size of the target object at each step does not change too drastically. 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.

6. Conclusion

In this work, we introduced a new task called *Embodied Visual 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 strategical movements that were different from shortest path, to recover the semantics and shape of occluded objects.

Acknowledgments. We thank Manolis Savva, Marcus Rohrbach, and Lixing Liu for the helpful discussions about task formulation, and Dipendra Misra for helpful RL training tips. This work was supported in part by NSF (CAREER IIS-1253549), AFRL, DARPA, Siemens, Samsung, Google, Amazon, ONR YIPs, ONR Grants N00014-16-1-{2713,2793}, the IU Office of the Vice Provost for Research, and the College of Arts and Sciences, the 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] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision (IJCV)*, 1988. 3

[2] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg. A dataset for developing and benchmarking active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 3

[3] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. 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 (CVPR)*, 2018. 3

[4] 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

[5] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

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

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

[8] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. *International Conference on Lefvarning Representations (ICLR)*, 2016. 5

[9] S. Bambach, D. J. Crandall, L. B. Smith, and C. Yu. Toddler-inspired visual object learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1, 3

[10] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[11] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[12] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville. HoME: A household multimodal environment. *arXiv preprint arXiv:1711.11017*, 2017. 3

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

[14] R. Cheng, A. Agarwal, and K. Fragkiadaki. Reinforcement learning of active vision for manipulating objects under occlusions. In *Conference on Robot Learning (CoRL)*, 2018. 3

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 5

[16] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 5

[17] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Neural Modular Control for Embodied Question Answering. In *Conference on Robot Learning (CoRL)*, 2018. 3

[18] J. Denzler and C. 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

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

[20] K. Ehsani, R. Mottaghi, and A. Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3

[21] P. Follmann, R. König, P. Härtinger, and M. Klostermann. Learning to see the invisible: End-to-end trainable amodal instance segmentation. *arXiv preprint arXiv:1804.08864*, 2018. 3

[22] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1

[23] R. Girshick. Fast R-CNN. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 3, 5

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 3

[25] A. Gonzalez-Garcia, A. Vezhnevets, and V. 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] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[27] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[28] X. Han, H. Liu, F. Sun, and X. Zhang. Active object detection with multi-step action prediction using deep q-network. *IEEE Transactions on Industrial Informatics*, 2019. 3

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 3, 4, 5

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3, 4, 6

[31] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012. 6

[32] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3

[33] H. Izadinia, Q. Shan, and S. M. Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[34] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

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

[36] E. Johns, S. Leutenegger, and A. J. Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

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

[38] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Amodal completion and size constancy in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[39] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017. 3

[40] D. Kragic, M. Björkman, H. I. Christensen, and J.-O. Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and autonomous Systems*, 2005. 3

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 1, 3

[42] K. Li and J. Malik. Amodal instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 3

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2

[44] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 3

[45] M. Malmir, K. Sikka, D. Forster, J. Movellan, and G. 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

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

[47] S. Mathe, A. Pirinen, and C. Sminchisescu. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[48] D. Novotny, D. Larlus, and A. Vedaldi. Learning 3D object categories by looking around them. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[49] D. Pathak, Y. Shentu, D. Chen, P. Agrawal, T. Darrell, S. Levine, and J. Malik. Learning instance segmentation by interaction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 3

[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[51] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 3

[52] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 1, 3, 6

[53] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. 2, 6

[54] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 3

[55] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[56] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[57] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[58] R. S. Sutton, A. G. Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998. 5

[59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 3

[60] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin*, 2012. 1

[61] D. Wilkes and J. K. Tsotsos. Active object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1992. 3

[62] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian. Building generalizable agents with a realistic and rich 3D environment. *arXiv preprint arXiv:1801.02209*, 2018. 2, 3

[63] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson env: real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[64] F. Xiao and Y. J. Lee. Video object detection with an aligned spatial-temporal memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[65] X. Ye, Z. Lin, H. Li, S. Zheng, and Y. 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] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *International Conference on Lefvarning Representations (ICLR)*, 2015. 1, 3

[67] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

[68] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 2

[69] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 3

[70] Y. Zhu, Y. Tian, D. Mexatas, and P. Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

Appendix

In the Appendix, we will provide more information about our dataset.

A. Object Category

In our dataset, there are eight object categories, including bed, chair, desk, dresser, fridge, sofa, table, washer. In addition to Fig.4 in the main paper, we show the number of instances for each object category in Table 2. As can be seen, the distribution over eight categories is fairly balanced. In Fig. 12, we show some examples for each object category.

	bed	chair	desk	dresser	fridge	sofa	table	washer	total
Train	1687	1009	1333	737	900	1742	981	551	8940
Val	197	122	207	82	103	206	144	52	1113
Test	427	210	330	172	207	456	264	104	2170

Table 2: Number of instances for each object category.

B. Shortest Path

For each spawned location in the environment, we compute a shortest path from the initial location to the target object. In Fig. 13, we show four examples. In the 2D top-down maps, the blue dot denote the target object; the red regions represent potential spawning locations of the agent; the green dots denote the selected spawning location; the blue curves are the shortest-path trajectories. The bottom five rows are agents' observations in each step.

C. House Names in SUNCG

In the text files (train.txt, val.txt, test.txt), we list all the house names for training, validation and test in https://www.cc.gatech.edu/~jyang375/evr.html.

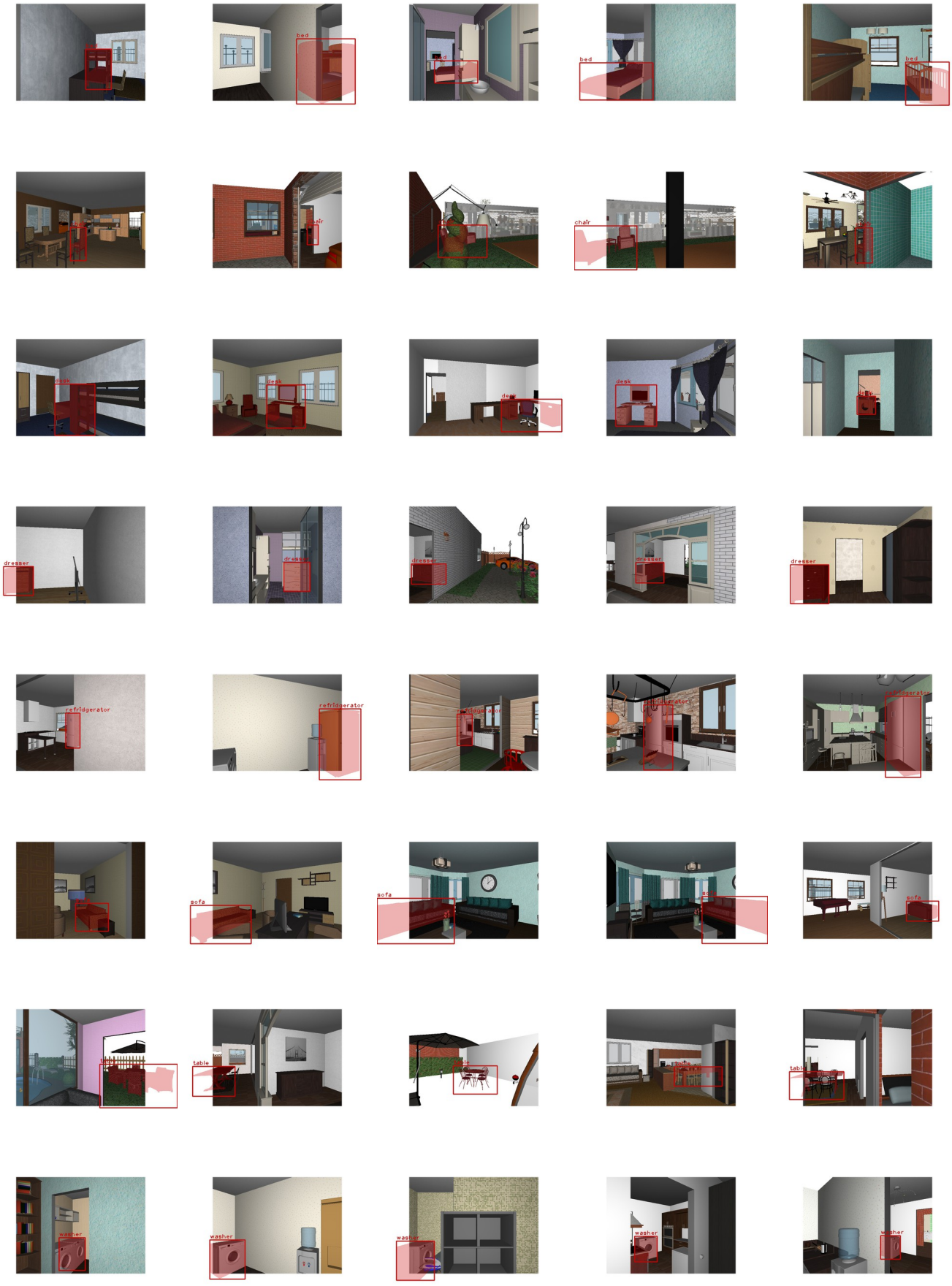

Figure 12: Visualizing examples of our dataset. In each row, we visualize ground-truth annotations for bed, chair, desk, dresser, fridge, sofa, table, washer.

Figure 13: Exemplar top-down maps of shortest paths (top row) and the corresponding observations at step 0, 3, 6, 9 and 12 (bottom five rows).