Evidential Active Recognition: Intelligent and Prudent Open-World Embodied Perception

Supplementary Material

Abstract

This document serves as the supplementary material for "Evidential Active Recognition: Intelligent and Prudent Open-World Embodied Perception". Firstly, we offer an in-depth look at the proposed dataset, covering aspects such as the data collection procedure, associated statistics, and specific examples. Subsequently, we describe the training steps and hyperparameters utilized in our proposed method, as well as those applied in the compared baselines. Finally, we provide a comprehensive analysis of the results and statistics pertaining to our proposed method.

1. Difficulty-designated Testing Dataset

Given that the current indoor simulator as in [7, 9] do not provide standard evaluations for active recognition, we choose to collect and organize a novel dataset specifically tailored for this task. We observed that to effectively highlight the necessity of movement during recognition, recognition challenges such as long distances and significant occlusions are best accommodated in testing instances. To this end, we introduce a new dataset in this paper comprising of 13200 testing instances across 27 indoor categories. The proposed dataset will be released to the public.

The contribution of our proposed dataset is threefold. First, to the best of our knowledge, no publicly available dataset exists for evaluating active recognition agents in simulators. Our dataset fills this gap, offering a benchmark for comparing various active recognition methods using widely-accepted simulators under uniform evaluation protocols. Second, we classify each testing instance in our dataset by its recognition difficulty level. This difficulty rating takes into account three factors, which is determined by comparing the asset-only unoccluded mask with the observed target mask and analyzing other target attributes. Third, recognizing that the original semantic annotation [1] contains noise and artifacts, we have refined our testing dataset by eliminating all unqualified testing instances.

1.1. Data generation

Our dataset is derived from the existing MP3D dataset [1], which features 90 distinct buildings. A testing instance is characterized by two components, namely, the agent's starting position and the segmentation map of the target. The starting position of the agent encompasses both the camera's location and orientation, both of which are randomly sampled within a given building. We then verify if there's an object, visible from this location, that satisfies the following criteria: 1. The object belongs to one of the 27 categories of interest. 2. The observable pixel count of the object must be no less than 300. 3. The distance between the object and the agent must range between 3 to 6 meters. If no object at the current location satisfies the established criteria, the random sampling process continues.

We generate 200 testing instances for each building, yielding a total of 18000 instances. It is important to note that the semantic annotations in the MP3D dataset [1] can be noisy and prone to artifacts, necessitating a careful review of these generated testing instances. Upon eliminating instances with incorrect semantic annotations or poor photographic quality, we are left with a final total of 13200 testing instances. Each instance is assigned a difficulty level, which quantifies the overall recognition challenges, factoring in *visibility*, *relative distance*, and *observed pixels*. Here, *visibility* is calculated as the ratio between observed mask and amodal mask of the target, *i.e.*, a heavier occlusion level or out-of-view condition leads to a lower *visibility*. After normalizing each aspect, these three components are linearly combined using respective weights of 0.2, 0.2, and 0.6 to calculate the difficulty score. Instances with a score lower than 0.33 are classified as "hard", those with scores ranging from 0.33 to 0.66 are deemed "moderate", and the remaining instances are labeled as "easy".

To be more specific, we define the *visibility* as a ratio, denoted x_{vis}, the *relative distance* in meters as x_{dist}, and the *observed pixels* as x_{pixel}. Utilizing these parameters, the difficulty level is computed according to the following equation:

$$s_{\text{diff}} = 0.2 * x_{vis} + 0.2 * (1 - \text{norm}(x_{dist})) + 0.6 * \text{clip}(\frac{x_{pixel}}{102400}), \tag{1}$$

where norm(.) represents a normalization function that converts the distance range from 3 to 6 meters into a scale from 0 to 1. The function clip(.) serves as a clipping mechanism, restricting the input value to a maximum of 1. After obtaining s_{diff} for each testing instance, the difficulty level is given by

$$\begin{cases} \text{hard} & \text{if } s_{\text{diff}} < 0.33 \\ \text{moderate} & \text{if } 0.33 \leq s_{\text{diff}} \leq 0.66 \\ \text{easy} & \text{if } s_{\text{diff}} > 0.66 \end{cases}. \tag{2}$$

We employ the HM3D dataset [9], which includes 145 buildings, as the scene dataset for training. The generated training instance begins with the random placement of an agent and the subsequent query of a target object. Recognition difficulty is not supplied during training, as each training instance is generated in real-time, and rendering amodal segmentation maps for randomly selected targets can be time-consuming. During the evaluation phase, we use the proposed dataset. Agents from various methods are deployed at the specified location and tasked with identifying the same object for each testing instance. A recognition method is considered superior if it can yield a more accurate prediction with a fixed number of movements.

1.2. Statistics of the Dataset

Figure 1 presents the statistics of the generated dataset. We begin by illustrating the category distribution in Figure 1a. Given that the data collection process is random, the category distribution effectively mirrors the occurrence of different classes across all buildings. Figure 1c displays the number of different difficulty levels associated with each category. Additionally, Figure 1 also provides the *relative distance* distribution, *visibility* distribution, and occlusion distribution for each category. *Visibility* is determined by the ratio of the observed object mask to the amodal segmentation mask of the target, thereby taking into account out-of-view cases. In contrast, occlusion is calculated solely based on the agent's actual viewing window.

1.3. Dataset Visualization

Additional visualizations of testing instances can be found in Figure 2. Each example features an image showing the target as well as a full semantic segmentation. The caption for each example includes the target category, difficulty level, *visibility* (expressed as a percentage), and *relative distance*. These examples demonstrate that the proposed dataset encompasses a wide range of viewing conditions that an agent might encounter in real-world deployments.

2. Implementation Details

For the sake of reproducibility, this section details the implementation of our method alongside the compared baselines. Initially, we clarify the network architectures employed in this paper. Subsequently, we describe the training strategy for various agents and the hyperparameters utilized during the training phase.

2.1. Network Architecture

The recognition model in our proposed method is based on Faster-RCNN [5], albeit with the region proposal network removed as the query box is directly provided by the ground truth. The query box could be further obtained by an class-agnostic visual tracker [10]. The backbone of the recognition model is ResNet-50 [3], pretrained on ImageNet [6], with the first three residual blocks remaining fixed during subsequent training. The ROI feature used for classification is derived from the C4 head, as per [5]. The class prediction for a single-step observation, namely, α in our main paper, is ensured to be non-negative by applying the exponential function. Other non-negative functions, such as the sigmoid function, could also be used. The final opinion regarding the target category is obtained by employing the combination approach outlined in the main paper.

The policy component developed in our proposed method comprises a visual encoder, an embedding layer to encode the last action, a linear layer to predict action distribution (actor), and a linear layer to predict the action's value (critic). The visual encoder receives inputs comprised of v^0, v^t, and q^t, with q^t represented as a binary mask resized to match the resolution of the visual observations. These inputs are concatenated and passed through the visual encoder, which consists of four convolution blocks. Each block includes a 5×5 convolutional layer, a batch normalization layer, a ReLU, and a 5×5 max pooling layer, aligning with the structure described in [11] for a fair comparison. The last action at the previous step $t - 1$ is encoded with the embedding layer and concatenated with the image feature following the visual encoder. The aggregated feature is subsequently used to predict both the action to be taken and its value.

The baselines compared in this paper exhibit a structure similar to that of our proposed method. However, as outlined in `Amodal-Rec` [11], we re-implement their method using a single-layer conv-GRU to sequentially aggregate the features from ResNet-50. The features from the conv-GRU is then utilized to predict the class.

2.2. Training

As discussed in [2, 11], joint training of the recognition model and the policy could result in sub-optimal outcomes. This is primarily because the recognition model may not provide accurate reward feedback for policy learning, particularly in the early training stage. Consequently, our proposed method employs a staged training strategy. Initially, we train the recognition model using frames collected by a heuristic policy; specifically, a fixation policy that centers the target within the observation. To further increase the randomness, we implement the fixation policy with a 20% probability assigned to stochastic `move_forward` and 80% to other fixation adjustments. This fixation policy also serves as a baseline in experiments. Once trained, this recognition model is then fixed and integrated into the proposed agent to train the policy component. Unlike `Amodal-Rec` [11], our method does not require retraining of our recognition model to adapt to the learned policy. This is because our approach generates predictions using only the current observation, without relying on sequentially aggregated features.

The proposed method and other baselines are implemented using PyTorch and trained with two NVIDIA 3090 GPUs. The batch size is set to 30 for both the recognition and policy of our training. For the recognition model, we employ stochastic gradient descent (SGD) with a learning rate of 0.005, momentum of 0.9, and weight decay of 0.0005. A learning rate scheduler is utilized with a step size of 5000 and a gamma of 0.9. For the policy, Proximal Policy Optimization (PPO) [8] is used, applying the Adam optimizer with a learning rate of 2.5×10^{-5} and epsilon of 5×10^{-5}. We set $\gamma = 0.99$ and $\tau = 0.95$ during the computation of returns. The policy training loss is thus a combination of the action loss, value loss, and an action entropy loss to promote exploratory behavior, with weighting coefficients of 1, 0.5, and 0.01, respectively. Moreover, the training of the recognition part and the policy takes approximately 14 and 24 hours to converge, respectively.

3. Quantitative Results and Analysis

3.1. Categorical Accuracy

Figure 3 illustrates the testing success rate for different object categories in the proposed dataset. As was detailed in the main paper, the success rate is computed based on the prediction made after the final movement. Our method exhibits notable improvement on certain categories, including windows, sofas, curtains, bed and towels.

We observe an imbalance in recognition accuracy across different categories. This discrepancy may stem from the training approach of the recognition module, which utilizes randomly sampled observations from training scenes. Some object categories, such as gym equipment, are less prevalent in training indoor environments, or are often positioned in locations that are challenging to observe, like sinks. Consequently, the training data for these categories is inadequate, leading to sub-optimal results. A potentially more effective approach could be to integrate a pre-trained vision recognition module, such as CLIP [4]. This alternative, however, is left for future investigation.

3.2. Action Distribution

In Figure 4, we present the action distributions of our proposed method at various steps ($t = 1, \ldots, 9$), in comparison with `Amodal-Rec` [11] and the established fixation policy. A key observation is the relative infrequency of `look_up` and `look_down` actions in both learned policies. This trend can be attributed to the agent's floor-level operation, where `look_up` and `look_down` actions generally yield minimal benefits for the recognition process. Additionally, our proposed agent demonstrates a tendency to rotate or tilt its camera during the initial four steps, followed by forward movement in subsequent steps. This strategy suggests that the learned policy prioritizes centering the target within the viewing window, subsequently approaching it for enhanced clarity and proximity in recognition. This approach mirrors human recognition tactics, wherein the target is initially fixated upon before moving closer for better examination. Lastly, our method exhibits a more varied action distribution compared to `Amodal-Rec`[11], indicative of a broader range of exploratory behaviors in diverse recognition scenarios.

3.3. Distance Change

In this study, we report the average relative distance between the agent and the target across various steps, as illustrated in Figure 5. Intuitively, being closer to the target enhances recognition accuracy. With our proposed method, the agent is able to reduce the average distance to the target by 0.85m at the final step, given a forward movement of 0.25m per step. It is important to highlight that navigating towards the target in our experimental setup presents significant challenges. The agent operates in an unseen environment and relies solely on RGB observations, requiring to avoid blocking obstacles for effective movements.

3.4. Statistical Analysis

We also present standard errors over five runs for two learning-based methods in Table 1. Due to page constraints, please refer to our main paper for the mean performance metrics.

4. Qualitative Results

We have also included additional qualitative results of the proposed method on the dataset in the form of a demonstrative video. In each episode, the visual observation is displayed on the left, with the predicted action and class for the current observation noted in the respective left and right corners. On the right side of the video, we provide both the single-frame belief and the combined belief up to the current timestep. Please refer to the combined belief for our final prediction. We can observe the trend of the combined opinion as more steps are taken.

5. Failure Cases

Our method's failures are mainly of two types: (a) Navigation failures: These occur due to the complex navigation skills required in the simulator. Failures often arise when better viewpoints for recognition are inaccessible from the starting location, owing to obstacles, varying floor levels, or movement constraints. (b) Hard-level failures: As the testing environments are completely novel, failures occur when objects are distant, or have ambiguous appearances, making acquiring new observations ineffective.

Table 1. The standard error of methods with 5 random seeds. Please refer to our main paper for their mean performance.

Method	**Easy**		**Moderate**		**Hard**		**All**	
	top-1	*top-3*	top-1	top-3	top-1	top-3	top-1	top-3
Amodal-Rec [11]	±0.3	±0.0	±0.4	±0.1	±0.5	±0.1	±0.4	±0.1
Ours	±0.3	±0.0	±0.4	±0.0	±0.4	±0.1	±0.3	±0.1

References

[1] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 1

[2] Lei Fan and Ying Wu. Avoiding lingering in learning active recognition by adversarial disturbance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4612–4621, 2023. 2

[3] 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*, pages 770–778, 2016. 2

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 2

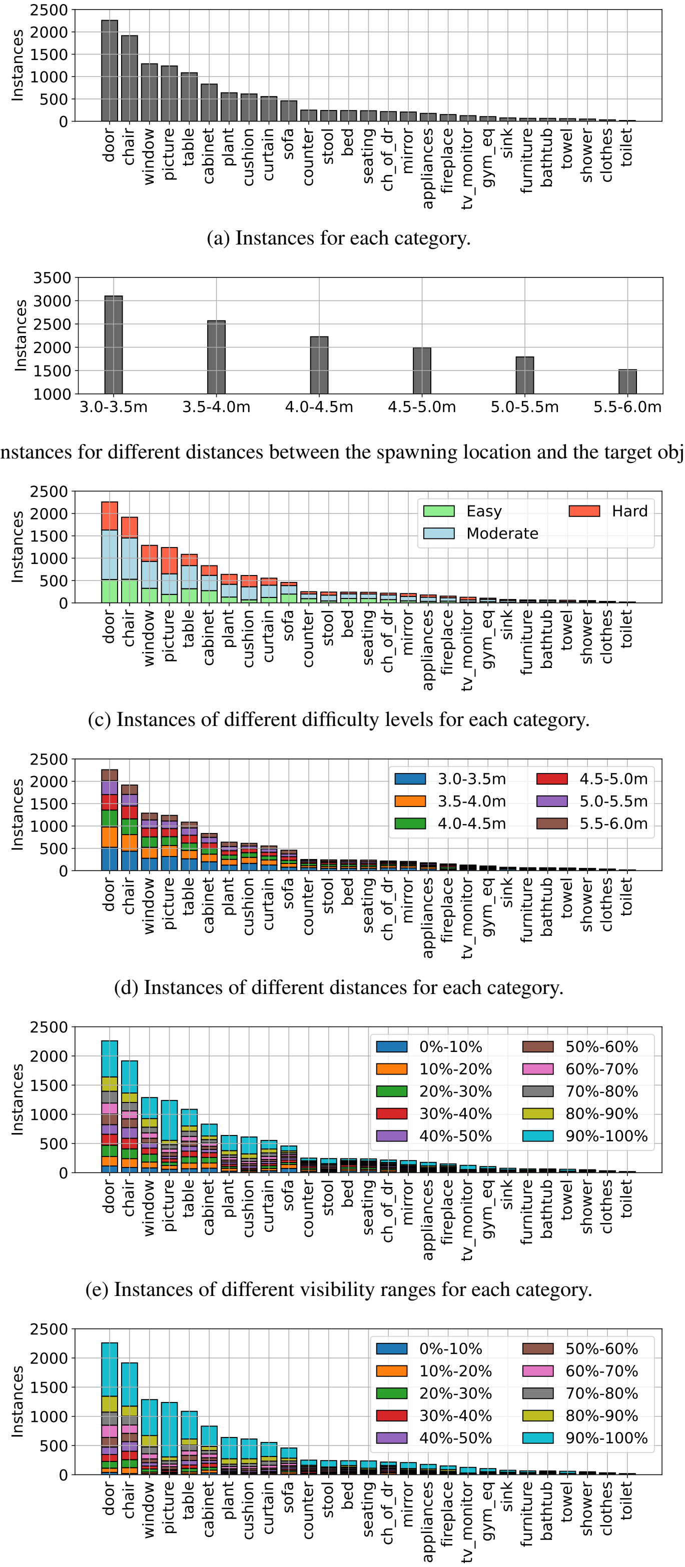
[7] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 1

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[9] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 34:251–266, 2021. 1

[10] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019. 2

[11] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2040–2050, 2019. 2, 3, 4



(a) Instances for each category.

(b) Instances for different distances between the spawning location and the target object.

(c) Instances of different difficulty levels for each category.

(d) Instances of different distances for each category.

(e) Instances of different visibility ranges for each category.

(f) Instances of different occlusion ranges for each category.

Figure 1. Additional statistics related to the proposed dataset, examining aspects such as category, distance, difficulty level, visibility ranges, and occlusion ranges. The `ch_of_dr` and `gym_eq` are short for ”chest of drawers” and ”gym equipment”, respectively.

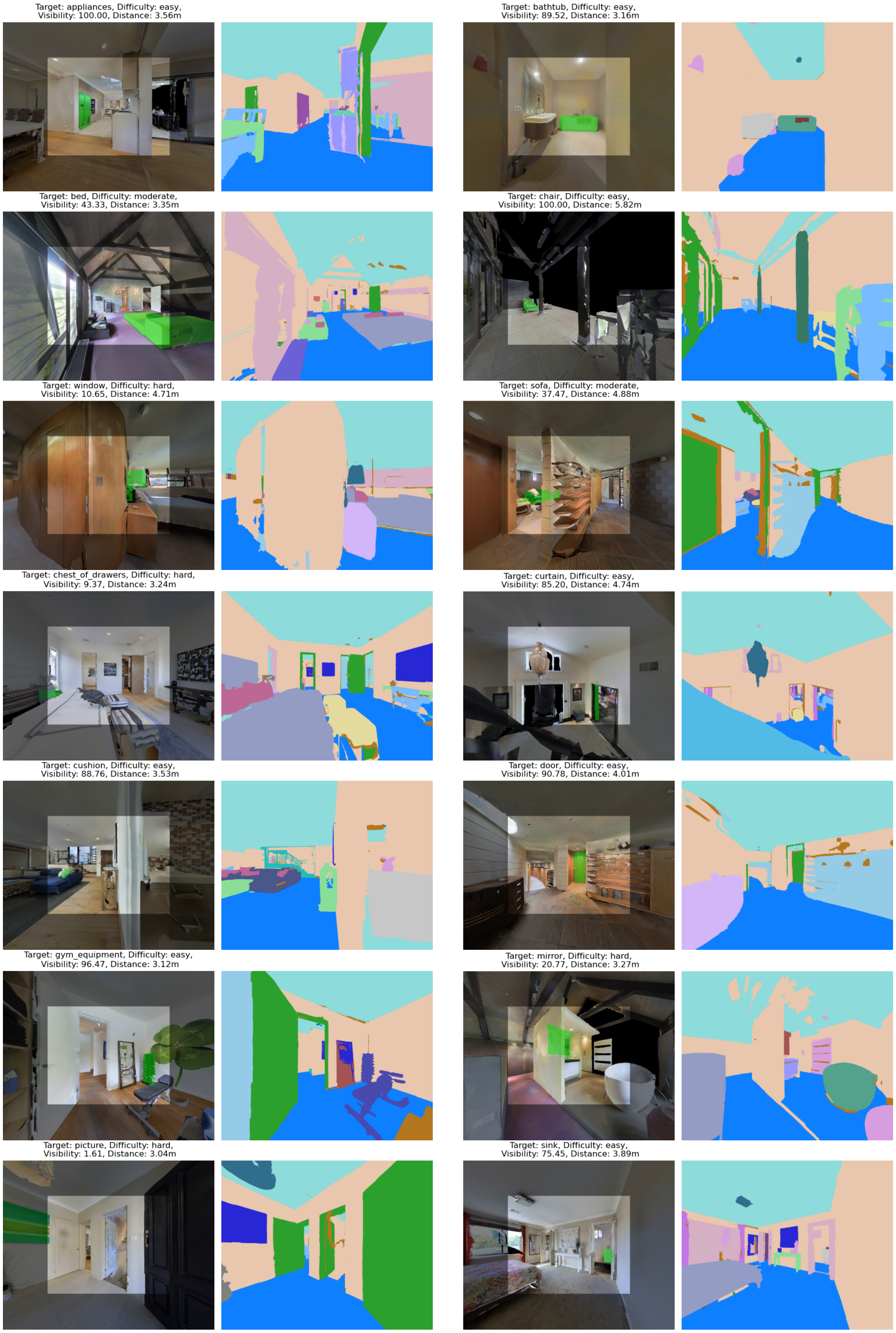

Target: appliances, Difficulty: easy,
Visibility: 100.00, Distance: 3.56m
Target: bathtub, Difficulty: easy,
Visibility: 89.52, Distance: 3.16m
Target: bed, Difficulty: moderate,
Visibility: 43.33, Distance: 3.35m
Target: chair, Difficulty: easy,
Visibility: 100.00, Distance: 5.82m
Target: window, Difficulty: hard,
Visibility: 10.65, Distance: 4.71m
Target: sofa, Difficulty: moderate,
Visibility: 37.47, Distance: 4.88m
Target: chest_of_drawers, Difficulty: hard,
Visibility: 9.37, Distance: 3.24m
Target: curtain, Difficulty: easy,
Visibility: 85.20, Distance: 4.74m
Target: cushion, Difficulty: easy,
Visibility: 88.76, Distance: 3.53m
Target: door, Difficulty: easy,
Visibility: 90.78, Distance: 4.01m
Target: gym_equipment, Difficulty: easy,
Visibility: 96.47, Distance: 3.12m
Target: mirror, Difficulty: hard,
Visibility: 20.77, Distance: 3.27m
Target: picture, Difficulty: hard,
Visibility: 1.61, Distance: 3.04m
Target: sink, Difficulty: easy,
Visibility: 75.45, Distance: 3.89m

Figure 2. Visualization of testing instances from our dataset. Each visualization includes an image with the target covered by a green amodal mask and the corresponding semantic segmentation. Note that both the visualized image and the semantic segmentation are enlarged for calculating visibility. The actual viewing window is displayed with increased brightness on the left of each example.

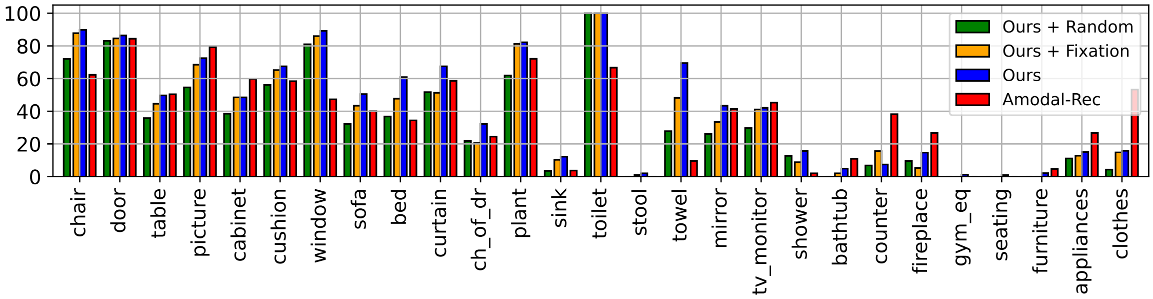


Figure 3. The recognition success rate on each object category.

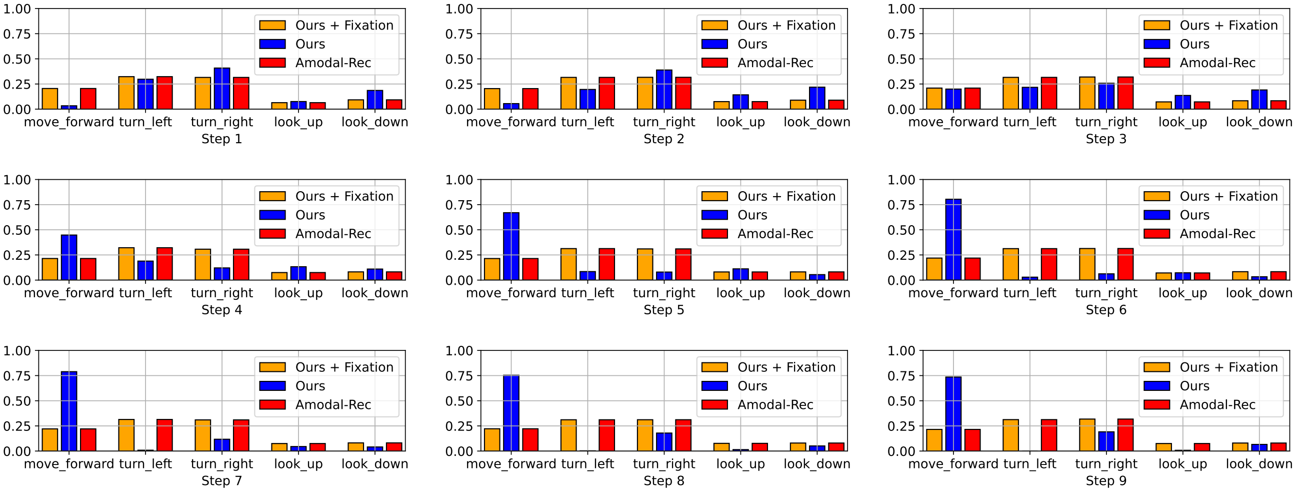


Figure 4. Action distributions at steps $t = 1, \ldots, 9$ on the proposed testing dataset.

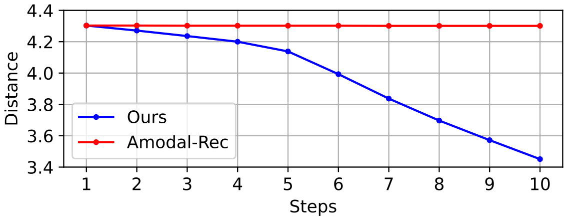


Figure 5. Distance to the target (meters) at steps $t = 1, \ldots, 10$ on the proposed testing dataset.