Memory-Augmented Scene Understanding and Exploration for Open-World Aerial Object-Goal Navigation
Supplementary Material

1. Overall Experimental Details

Software and Hardware Configurations. We conduct our experiments on servers equipped with NVIDIA L20 GPUs (48GB VRAM) running NVIDIA CUDA Toolkit 12.4. For navigation action inference, we use the same hardware setup and perform inference across four GPUs in parallel.

Inference Settings. We use the AirSim plugin [2] in Unreal Engine to collect RGB images and depth maps at a resolution of 512×512. Our trained model takes RGB-D inputs along with the object description and outputs navigation actions, which are then executed by AirSim to control the UAV's movement. In all experiments, we limit each navigation episode to at most 100 steps.

Trajectory Generation Configuration. Since UAV-ON [3] provides only annotated data without trajectory information, we generate training trajectories following the procedure described in UAV-ON. For each environment, we first collect the scene point cloud and construct a global voxel map. Then, using the start and goal locations specified in the UAV-ON dataset, we employ the A* [1] algorithm to generate a collision-free trajectory on the global voxel map. Finally, through the AirSim interface, we obtain RGB images and depth corresponding to the points along the generated trajectory.

2. Additional Experiments

Empirical Evaluation of Robustness to Depth and Pose estimation noises. We have added a noise study by injecting Gaussian noise with standard deviation 0.1 meters to (i) depth and (ii) pose. OctMem-Agent shows only a slight drop under noise, suggesting that our memory and instruction-guided extraction are reasonably robust to sensing and pose noises. we can further enhance robustness by injecting such noise during training.

Method	SR	OSR	SPL
OctMem-Agent + depth noise	17.80%	27.00%	5.85%
OctMem-Agent + pose noise	18.51%	28.09%	6.35%
OctMem-Agent	19.50%	29.30%	6.37%

Hyperparameters Sensitivity Analysis. We performed a sensitivity analysis on voxel sizes s_k and the distance threshold d_b. Our configuration ($s_k \in \{5.0, 25.0\}$, $d_b = 50$) provides the strongest overall performance, while both smaller and larger s_k degrade SR/OSR, indicating that overly fine or coarse voxelization is suboptimal. By contrast, performance varies only mildly when changing d_b.

Method	SR	OSR	SPL
$s_k \in \{2.5, 12.5\}$	18.10%	28.00%	5.99%
$s_k \in \{5.0, 25.0\}$	**19.50%**	**29.30%**	**6.37%**
$s_k \in \{7.5, 37.5\}$	16.15%	24.63%	5.80%
Method	**SR**	**OSR**	**SPL**
$d_b = 25$	18.30%	28.20%	6.33%
$d_b = 50$	**19.50%**	**29.30%**	**6.37%**
$d_b = 100$	19.20%	28.30%	5.87%

3. Additional Case Studies

We present additional qualitative comparisons between our OctMem-Agent and the baseline methods on UAV-ON. As shown in Figure 1, Figure 2, and Figure 3, our OctMem-Agent consistently performs effective exploration, reaches the target area, successfully identifies the target object, and navigates to it. However, the baseline methods often lose track of the target during navigation due to the lack of target-specific guidance, resulting in ineffective exploration that prevents them from reaching the target region. Even when they approach the correct area, they frequently misidentify the target object and subsequently deviate from the correct trajectory, ultimately failing to complete the task.

4. Trajectory Visualization for Baseline Comparison

We further visualize the navigation trajectories of our method and the baseline methods for a more detailed comparison, as shown in Figure 4. From these visualizations, we observe that the baseline methods, which rely solely on the current local observation, exhibit myopic behavior: their trajectories gradually drift away from the target and, due to the lack of an effective correction mechanism, they struggle

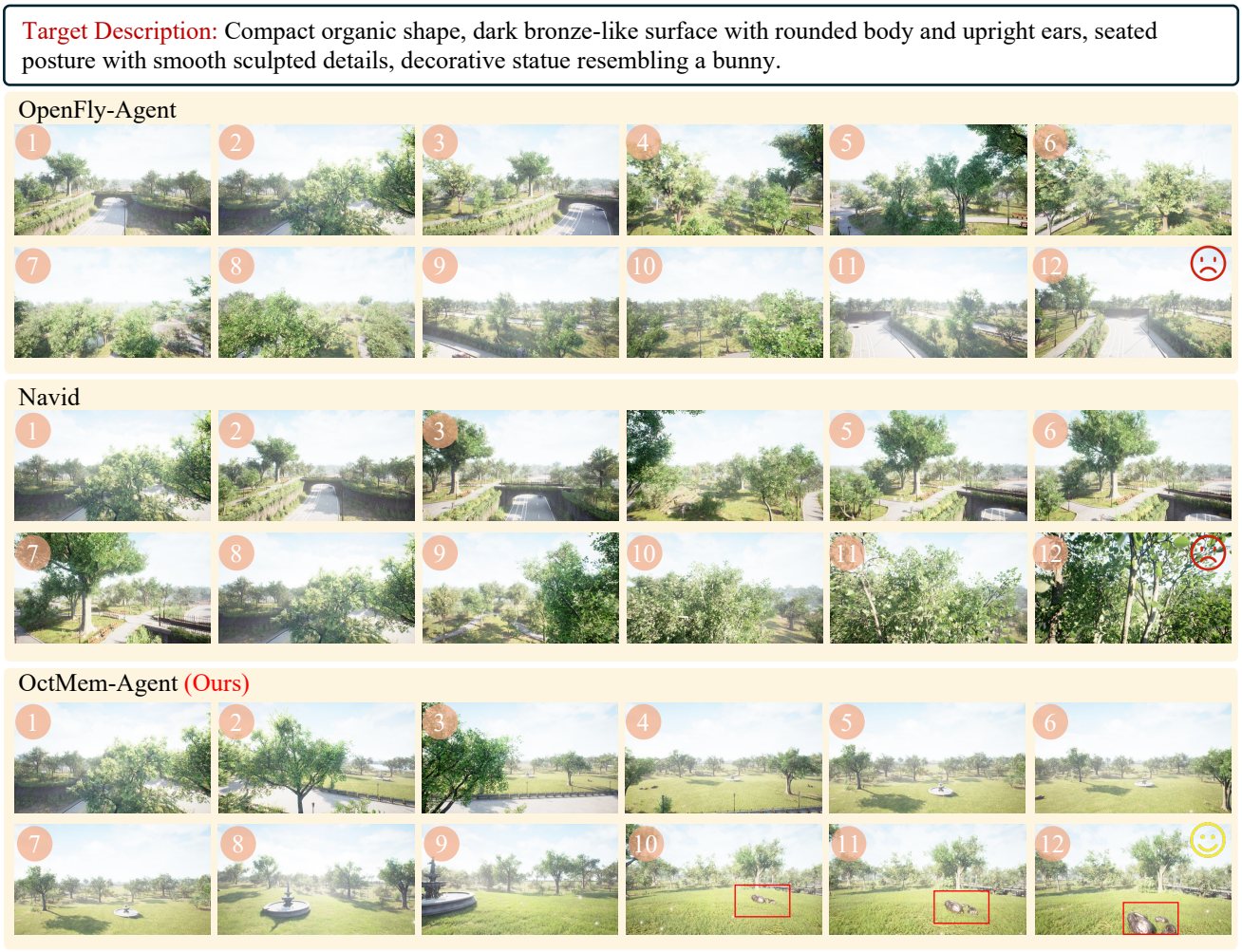

Figure 1. Qualitative comparison between our OctMem-Agent and the baseline methods on UAV-ON. In this example, both Open-Fly and Navid lose track of the target and engage in ineffective exploration—either flying straight ahead or wandering aimlessly—whereas our method performs purposeful exploration and ultimately localizes the target statue.

to return and re-initiate meaningful exploration once they deviate from the desired path. In contrast, our method leverages long-term spatial memory through the adaptive octree memory, enabling both robust target recognition and effective exploration that mitigates myopic behavior and ultimately guides the agent to the target.

References

[1] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 1968. 1

[2] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *arXiv preprint arXiv:1705.05065*, 2017. 1

[3] Jianqiang Xiao, Yuexuan Sun, Yixin Shao, Boxi Gan, Rongqiang Liu, Yanjing Wu, Weili Guan, and Xiang Deng. Uav-on: A benchmark for open-world object goal navigation with aerial agents. *arXiv preprint arXiv:2508.00288*, 2025. 1

Figure 2. Qualitative comparison of our OctMem-Agent and baseline methods on UAV-ON. In this example, Open-Fly explores the region near the target but fails to accurately localize it, whereas NAVID continues to fly straight ahead aimlessly.

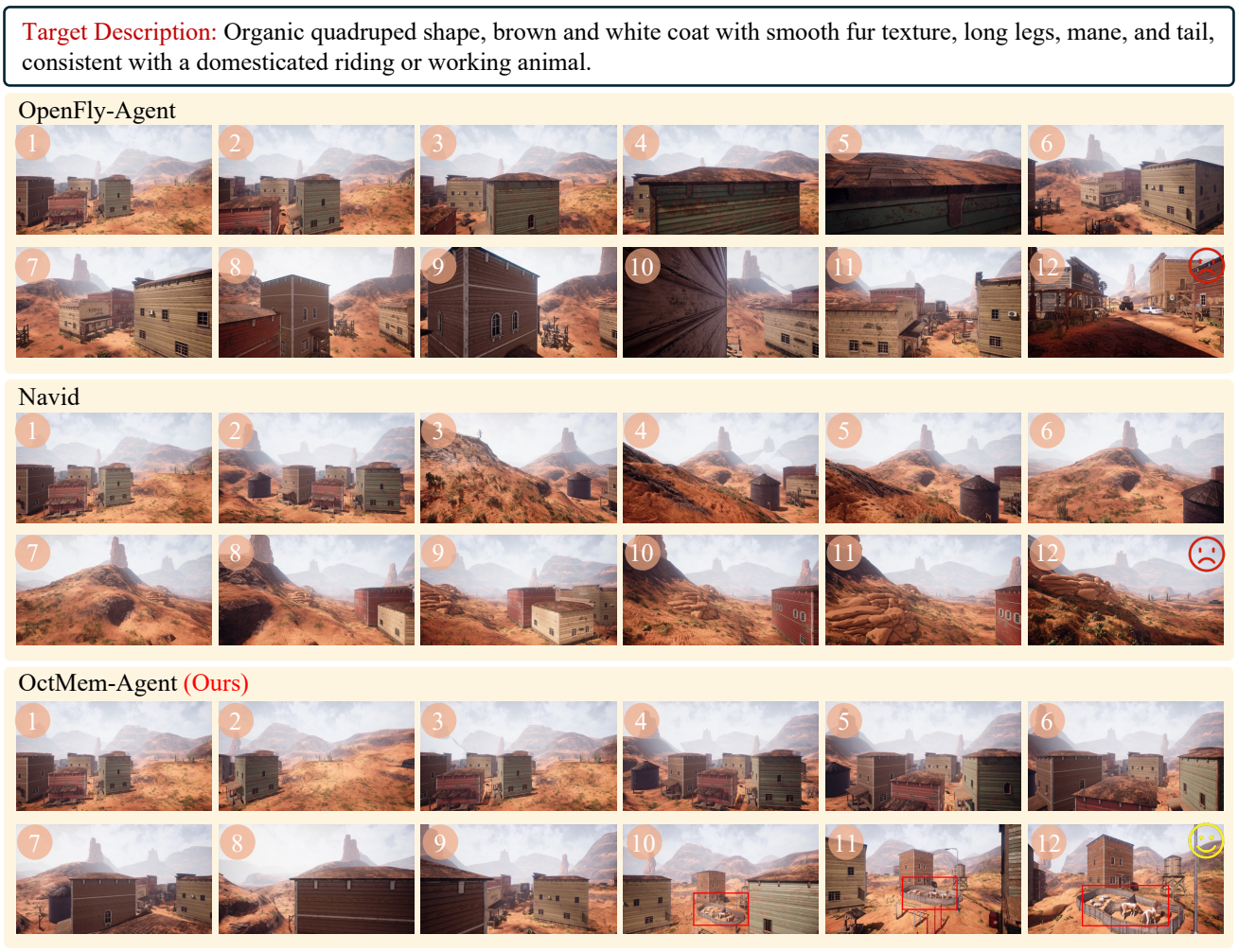

Figure 3. Qualitative comparison of our OctMem-Agent and baseline methods on UAV-ON.

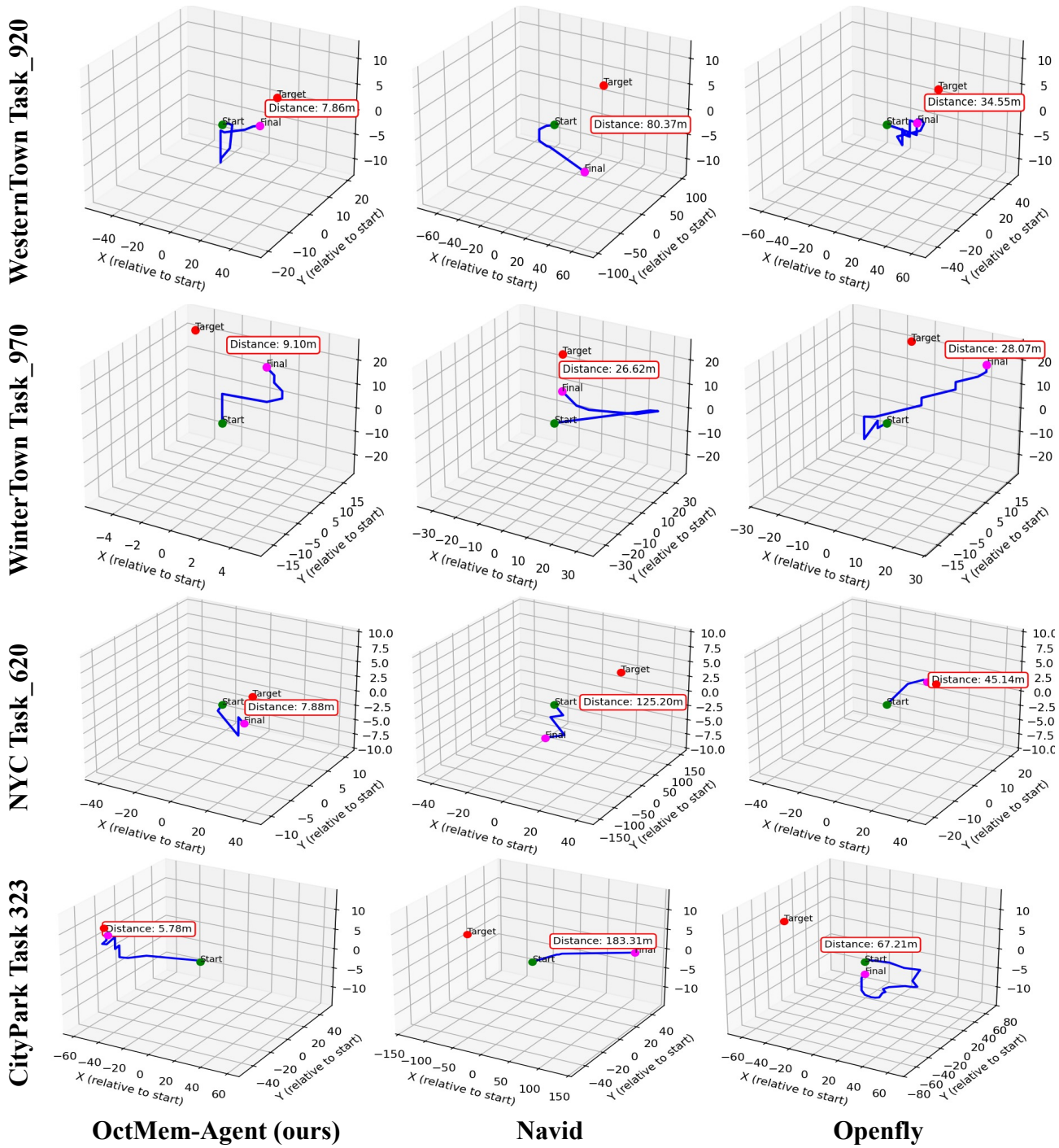

Figure 4. Trajectory comparison of our OctMem-Agent against baseline methods on UAV-ON.