

Supplementary for Submission No. 568

Contents

A Additional Experiments	1
A.1 Dynamic Obstacles in Goal-Conditioned Navigation	1
A.2 Performance in Seen Environments	2
B Experimental Details	3
B.1 Test Instances for Goal-Conditioned Navigation	3
B.2 Robot Platforms and Specifications	3
B.3 Common Parameters for CARE	4
B.4 Depth Estimation Model: UniDepthV2	4
B.5 Trajectory Selection in NoMaD and ViNT	5

A Additional Experiments

A.1 Dynamic Obstacles in Goal-Conditioned Navigation

To assess the robustness of CARE under dynamic conditions, we evaluate the system in an image goal-conditioned navigation setting with abrupt human intervention. Unlike the evaluations in the paper, this setup contains no static out-of-distribution (OOD) obstacles. All tests are conducted using the LoCoBot platform.

Dynamic Obstacle Scenarios. We introduce up to three humans acting as dynamic, unexpected obstacles, designed to test reactive collision avoidance. Each test trial includes one of the following scenarios¹.

- (i) A person appears from a side outside the field of view.
- (ii) A person appears from behind and overtakes the robot, stopping in front.
- (iii) A person walks toward the robot from the front and stops directly in its path.

Metrics and setup. For each of the three human intervention types, we run 10 trials with four methods: NoMaD, NoMaD+CARE, ViNT, and ViNT+CARE. We record the number trials where collisions occur.

Results and Analysis. Table 1 shows the number of collisions in each dynamic obstacle scenario. Without CARE, both NoMaD and ViNT fail to react effectively to abrupt human interventions, resulting in 7 to 10 collisions. In contrast, CARE-integrated policies complete all trials without a single collision, where this improvement is attributed to the repulsive force estimation based on predicted depth, enabling reliable detection of human legs even under sudden appearances.

¹Please refer to the supplementary video for better understanding.

Table 1: Number trials where collisions occur (out of 10 trials) for each dynamic obstacle type

Model	(i) Corner-appear	(ii) Behind-to-front	(iii) Front-approach
NoMaD	8/10	10/10	10/10
NoMaD + CARE	0/10	0/10	0/10
ViNT	7/10	10/10	10/10
ViNT + CARE	0/10	0/10	0/10

In scenario (i), where a person enters from the side, both NoMaD and ViNT occasionally succeed in generating avoidance waypoints in the opposite direction, thereby avoiding some collisions. However, in scenarios (ii) and (iii), where the human appears directly in front of the robot, both models fail in all trials. The key reason is the lack of geometric awareness in NoMaD and ViNT: although the models sometimes attempt avoidance by turning slightly, the generated waypoints do not ensure sufficient clearance. As a result, the robots consistently collide after small heading changes, or fail to respond to sudden frontal appearances.

In contrast, CARE-enabled policies successfully avoid all collisions in these scenarios. The estimated depth information allows CARE to compute repulsive directions with adequate lateral displacement, while the Safe-FOV mechanism suppresses forward motion until the robot achieves a sufficiently safe heading. This combination enables reliable and robust collision avoidance even under abrupt dynamic disturbances.

A.2 Performance in Seen Environments

We evaluate NoMaD and ViNT in a fully seen environment without added OOD obstacles as a sanity check to verify that our implementations reproduce the original models. All experiments are conducted using the LoCoBot platform.

Setup. The test environment and overall procedure follow the same setup as in the goal-conditioned navigation experiments in our paper. A topological graph is predefined using a sequence of image goals. For fair comparison, the subgoal image sequences fed to both NoMaD and ViNT are fixed and reused across all trials. This sequence is also identical to those used in the OOD experiments presented in the main paper.

Metric. We measure the success rate, defined as the percentage of trials in which the robot reaches the image goal without any collisions. Each configuration is tested for 10 trials.

Table 2: Success rate in seen environments without OOD obstacles

Model	Success Rate (10 trials)
NoMaD	100%
ViNT	100%

Results and Analysis. Both NoMaD and ViNT achieve a 100% success rate in the seen environment, completing all 10 trials without collision or failure. These results confirm that both the pre-trained vision-based models and the constructed topological navigation pipeline function reliably under seen conditions.

The experiments are conducted on the LoCoBot platform (see Sec. B.2), which closely matches the setup used in the original NoMaD and ViNT papers and is also employed to train and tune the released pretrained weights². The consistent performance thus validates the correct integration and implementation of the models. This outcome supports the analysis that the performance degradation observed in other experiments (e.g., with OOD obstacles or dynamic human interventions) is not

²<https://github.com/robodhruv/visualnav-transformer>

due to flaws in the experimental setup or execution, but rather reveals the inherent limitations of vision-only navigation models when deployed in unfamiliar environments without fine-tuning.

B Experimental Details

B.1 Test Instances for Goal-Conditioned Navigation

Figure 1 presents the 10 test instances used in the goal-conditioned navigation experiment described in the main paper. Each instance includes 15 OOD obstacles, shown as brown boxes, grouped into four to six clusters of varying shapes and placed randomly throughout the environment. These settings are designed to evaluate the robustness of navigation policies under unseen and cluttered conditions.



Figure 1: Test instances for goal-conditioned navigation

B.2 Robot Platforms and Specifications

We run experiments on three mobile platforms: LoCoBot, TurtleBot4, and RoboMaster S1. Figure 2 shows the actual hardware used in our experiments.

A note for LoCoBot. Although the original LoCoBot is built on a TurtleBot2 base, we implement our LoCoBot using a TurtleBot4 as TurtleBot2 has been discontinued. Nevertheless, two models are both differential drive type with very similar wheelbase width, wheel diameter, and etc. A custom 3D-printed camera mount is designed to match the fisheye camera position used in prior NoMaD and ViNT implementations.

Notes for parameters in Table 3.

- **Max Depth Range τ_z (m):** The maximum range (in meters) for depth sensing used during obstacle detection. This value is empirically tuned to prevent false positives caused by distant walls or floor misclassification.
- **Depth Offset (m):** An offset subtracted from the estimated depth to make obstacles appear closer (or further) than predicted. This compensates for the physical body radius of robots and camera mounting position, and is further tuned empirically to improve depth estimation.

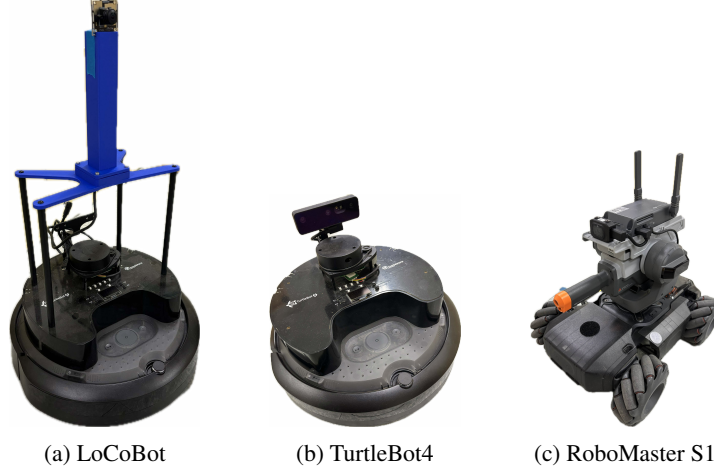


Figure 2: Mobile robot platforms used for evaluation. Each robot was equipped with a monocular RGB camera.

Table 3: Robot specifications and camera configuration. All resolutions are in pixels, and dimensions are in millimeters (mm).

Specification	LoCoBot	TurtleBot4	RoboMaster S1
Max Depth Range τ_z (m)	1.0	1.2	1.0
Depth Offset (m)	0.05	0.2	-0.1
Published Image Resolution (pixels)	320×240	320×200	640×360
Robot Size (L \times W \times H, mm)	$341 \times 339 \times 350$	$341 \times 339 \times 351$	$320 \times 240 \times 270$
Camera Height (mm)	340	245	240
Camera X-offset (mm)	10	-60	70

- **Published Image Resolution:** The resolution of the RGB images published by each camera over ROS 2.
- **Robot size (L \times W \times H):** Physical dimensions of each robot, measured in millimeters (mm).
- **Camera Height:** The vertical distance from the ground to the optical center of the camera.
- **Camera X-offset:** The horizontal distance (in mm) between the geometric center of robots and the camera. A positive value indicates a forward-facing offset (camera mounted ahead of center), while a negative value indicates a rear-facing offset.

B.3 Common Parameters for CARE

CARE uses a small set of task-agnostic parameters that are experimentally tuned for effective and safe navigation across all platforms. Table 4 summarizes the key scalar parameters used in the CARE module.

The value of θ_{clip} is set to allow sufficient rotation in response to repulsive forces while preventing excessive deviation or backward-pointing waypoints. The value of θ_{thres} is chosen to be smaller than θ_{clip} to trigger in-place rotation only in high-risk situations with large heading changes. Finally, the vertical offset ϵ is used to exclude points located above the robot (e.g., ceilings or high shelves) during depth-based top-down projection, improving the relevance of obstacle detection.

B.4 Depth Estimation Model: UniDepthV2

For monocular metric depth estimation, we employ UniDepthV2 with the ViT-S backbone (the smallest variant)³, enabling real-time inference. The model is used without any fine-tuning or adap-

³<https://github.com/lpiccinelli-eth/UniDepth>

Table 4: CARE module parameters used across all experiments.

Parameter	Value	Description
θ_{clip} (rad)	$\pi/4$	Maximum heading adjustment angle induced by repulsive force
θ_{thres} (rad)	$\pi/6$	Threshold for triggering in-place rotation when the desired heading change exceeds this value
ϵ (m)	-0.05	Vertical offset for filtering out ceiling points during top-down projection. A negative value indicates upward exclusion

tation. Top-down projections are computed using platform-specific parameters for maximum sensing range τ_z and vertical offset ϵ , as summarized in Table 3 and 4.

B.5 Trajectory Selection in NoMaD and ViNT

For NoMaD, we use the default diffusion-based sampling setup to generate 8 trajectories, each consisting of 8 waypoints in the robot’s local frame. In our experiments, we consistently select the second waypoint \mathbf{p}_2 from the first generated trajectory for control (as implemented in the open-source codes). ViNT, on the other hand, produces a single predicted trajectory using a Transformer decoder. Similarly, we select the second waypoint from this predicted trajectory as the control target (also the same with the original implementation of the open-source codes).

This fixed selection strategy is to isolate and evaluate the contribution of our proposed collision avoidance mechanism at the level of local planning. Since our focus is not on high-level trajectory selection or adaptive replanning, we maintain the same trajectory and waypoint index to control for variation across trials. All experiments are conducted under this consistent setting, using fixed random seeds and sampling strategies to ensure fair comparison.