EMBRACE: EMBODIED REASONING AND ACTION IN COMPLEX ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models are increasingly run as embodied agents in a closed loop in which some decisions depend on what the agent saw earlier. Placed in this loop, current models act on the view in front of them, revisit views already searched, lose a target that has left the frame, and switch referents as the viewpoint changes. The demonstrations that embodied models learn from record the view and the action at each step, not the observations the action was based on. We present EmbRACE, a dataset of 3,421 human demonstrations recorded from the egocentric view in photorealistic Unreal Engine environments, indoor and outdoor, on tasks with targets to search for, relational and ordered targets, and door and object interactions. Each of its 48,264 steps carries a rationale that states what the action was based on and is verified against the frames. A benchmark of 686 tasks in 7 held-out environments runs models in the closed loop and scores them on the final position and on the door and object states. Frontier models score highest when the target is in view from the start, lower on all three task types that depend on an earlier observation, and below half when the target has to be searched for or returned to. Fine-tuning on EmbRACE takes four open models of 2B to 9B parameters from 3% to 8% overall to 57% to 68%. Relative to training on the same trajectories without them, the rationales raise success by 7% to 18% across task types and halve the fraction of episodes that return to views already seen. The dataset, the benchmark, and the code will be made publicly available.



Figure 1: **EmbRACE.** (a) Photorealistic Unreal Engine environments, indoor and outdoor. (b) The closed loop. At each step the agent receives an egocentric view, produces a rationale, and issues one action, which changes the next view. (c) A human demonstration of the instruction shown, with the path taken and, at each step along it, the egocentric view, the rationale, and the action.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 2: Recurring closed-loop behaviors: (a) short-sighted exploration, (b) fragmented goal pursuit, (c) spatial-semantic instability.

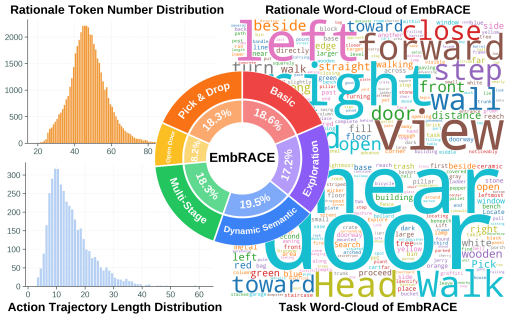


Figure 3: Statistical overview of the **EmbRACE** dataset, showing task-type distribution, trajectory lengths, and word clouds.

1 Introduction

Vision-language models now recognize and localize objects, read text and diagrams, reason over long videos, and follow open-ended instructions about what they see, and each generation extends them further (OpenAI, 2026; Anthropic, 2026; Google DeepMind, 2026; Qwen Team, 2026; Zhu et al., 2025). This progress has made them the natural candidates for embodied agents, and they are increasingly run in the loop, choosing each action from the current camera view (Zhang et al., 2024b; Cheng et al., 2024; Kim et al., 2024; Liu et al., 2026). In that loop the input is produced by the agent's own actions, and some decisions depend on what the agent saw earlier. Consider an agent asked to bring the red mug to the table by the window. It has to keep track of where it has already looked for the mug, return to a table that has since left its view, maintain the identity of that table as the viewpoint changes, and verify that the mug is in its hand. What these decisions depend on is not in the frame at the time and has to be retained across steps. Placed in this loop, a vision-language model acts on the view in front of it and retains little of what it saw before. Its search revisits views already covered (Figure 2a), a target that leaves the frame is lost (Figure 2b), and the referent fixed in an earlier view gives way to another as the viewpoint changes (Figure 2c).

A model learns such behavior from demonstrations, and the demonstrations available today are of three kinds. In R2R, VLN-CE and ALFRED the demonstration is a path computed by a planner from the map, so each step is chosen from the map rather than from what the agent has observed so far (Anderson et al., 2018; Krantz et al., 2020; Shridhar et al., 2020). Datasets that add a written rationale to every step build it on such paths (Wang et al., 2025; Liu et al., 2025) or on tabletop teleoperation (Zawalski et al., 2024; Sun et al., 2025), so the rationales describe steps chosen from the map or taken over a scene that is fully in view. In human demonstrations each step is chosen from what the demonstrator saw, but the record holds only the view and the action taken, not the observations the decision was based on (Ramrakhya et al., 2022; Padmakumar et al., 2022).

We present **EmbRACE** (Embodied Reasoning and Action in Complex Environments), a dataset of human demonstrations with a rationale at every step stating what the demonstrated action was based on, and a benchmark on environments held out from training. The tasks are built around the situations of the example above, set in photorealistic Unreal Engine environments, with targets to search for, targets that leave the view, referents that have to be held fixed as the viewpoint changes, and doors and objects that have to be acted on, so that navigation and interaction share one platform. The dataset contains 3,421 demonstrations with 48,264 steps, and the rationale for each step is written from the frames seen up to that step and verified by annotators against them. The 686 tasks on the 7 held-out environments form the benchmark, scored on the final position and on the door and object states, with human and random reference rows.

On this benchmark, frontier VLMs perform best when the target is visible initially, but succeed on fewer than half of the Exploration and Multi-stage tasks. Fine-tuning on EmbRACE raises the overall success of four open models from 3% to 8% to 57% to 68%. On the same trajectories, adding rationales improves success by 7% to 18% across task types. For the 9B model, it reduces episodes

Table 1: Demonstration datasets for embodied agents: scenes, coverage of navigation and object interaction, who produced the trajectories, and whether each step has a rationale.

Dataset	Scenes	Navigation	Interaction	Trajectory	Rationale
R2R, VLN-CE (Anderson et al., 2018; Krantz et al., 2020)	indoor	✓	×	planner	×
ALFRED (Shridhar et al., 2020)	indoor	✓	✓	planner	×
SPOC (Ehsani et al., 2023)	indoor	✓	✓	planner	×
Habitat-Web (Ramrakhya et al., 2022)	indoor	✓	✓	human	×
TEACh (Padmakumar et al., 2022)	indoor	✓	✓	human	×
BridgeData V2 (Walke et al., 2023)	tabletop	×	✓	human	×
R2R-CoT-320k (Wang et al., 2025)	indoor	✓	×	planner	✓
Nav-CoT-110K (Liu et al., 2025)	indoor	✓	×	planner	✓
ECoT, EMMA-X (Zawalski et al., 2024; Sun et al., 2025)	tabletop	×	✓	human	✓
EmbRACE	indoor, outdoor	✓	✓	human	✓

that revisit previously seen views from 29.2% to 13.6% and raises recovery after failed interactions from 40.5% to 70.7%.

The value of the dataset also extends beyond simulation: without real-robot demonstrations, the fine-tuned 9B model achieves 86% success across 375 episodes on 75 tasks in three physical scenes, compared with 4% before fine-tuning and 34% to 40% for zero-shot frontier VLMs. These results suggest that task-level decision behavior learned from EmbRACE can transfer to physical execution. The dataset, benchmark, and code will be made publicly available.

2 Related Work

Simulators and benchmarks. Embodied datasets are built in a few simulators. Habitat renders 3D scans of real buildings and hosts ObjectNav and VLN-CE, in which an agent reaches an object category or follows a narrated route and success is judged by the distance to the goal (Savva et al., 2019; Ramrakhya et al., 2022; Krantz et al., 2020). AI2-THOR and ProcTHOR are synthetic indoor houses and host ALFRED, TEACh, and CHORES, in which household and object tasks are judged on the resulting state (Kolve et al., 2017; Deitke et al., 2022; Shridhar et al., 2020; Padmakumar et al., 2022; Ehsani et al., 2023), and VirtualHome executes household programs from a third-person view (Puig et al., 2018). OpenEQA asks for answers to questions about a scene (Majumdar et al., 2024), and EmbodiedBench evaluates vision-language models as agents on 1,128 tasks in four environments at high and low action levels (Yang et al., 2025). EmbRACE is built in UnrealZoo, a collection of Unreal Engine scenes, indoor and outdoor (Zhong et al., 2025). Its benchmark holds out seven of these scenes, and success is judged on the agent's position and on the door and object states.

Demonstration datasets. What a demonstration records decides what can be learned from it. In R2R, VLN-CE, ALFRED, and CHORES the demonstration is a path computed by a planner from the map (Anderson et al., 2018; Krantz et al., 2020; Shridhar et al., 2020; Ehsani et al., 2023), and RT-1 and BridgeData V2 record teleoperated tabletop episodes, RT-1 keeping only the successful ones (Brohan et al., 2022; Walke et al., 2023). Habitat-Web and TEACh record human demonstrators acting in the loop, and Habitat-Web reports that imitating human demonstrations rather than shortest paths is what produces search behavior (Ramrakhya et al., 2022; Padmakumar et al., 2022). R2R-CoT-320k, Nav-CoT-110K, ECoT, and EMMA-X add a rationale to each step of such trajectories, written by Qwen2.5-VL-72B, by Gemini 2.5 Pro, or by a pipeline of detectors and language models (Wang et al., 2025; Liu et al., 2025; Zawalski et al., 2024; Sun et al., 2025). Navigation agents such as NaVid, Uni-NaVid, and NaVILA are trained on hundreds of thousands to millions of such steps (Zhang et al., 2024b;a; Cheng et al., 2024). Apart from these rationales, a step is recorded as the view and the action taken, and the rationales themselves sit on paths chosen from the map or on tabletops in full view. By contrast, EmbRACE records demonstrations performed by human demonstrators from the egocentric view, on tasks with targets to search for, relational and ordered targets, and door and object interactions, with a rationale for every step (Table 1).

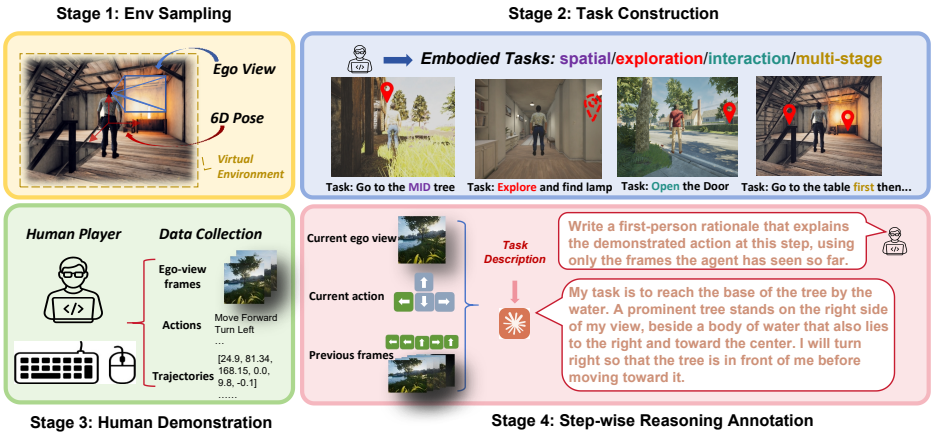


Figure 4: **EmbRACE** data construction pipeline.

3 Dataset and Benchmark Construction

EmbRACE consists of two parts. The dataset is a set of 3,421 human demonstrations for closed-loop tasks in 55 environments, 48,264 steps in all, in which every step is paired with a rationale written from the agent's viewpoint and verified by annotators. The benchmark is a set of 686 tasks in 7 further environments, run in the closed loop and scored on the agent's final position and on the door and object states. Both are produced by the same four-stage pipeline (Figure 4), and the tasks are of six types, built around the situations of the introduction (Appendix A).

3.1 Simulation Platform and Task Types

All data are collected in UnrealZoo (Zhong et al., 2025), a collection of photorealistic Unreal Engine environments with controllable characters, interactable doors and objects, and API-level access. We control one character and observe the scene through its egocentric camera. The environments come from UnrealZoo, and the tasks, demonstrations, rationales, and evaluation protocol are contributed here. We use 62 environments spanning indoor and outdoor scenes, daytime and night lighting, and a wide range of object density (Appendix B.1). 55 are used for training and 7 are held out, so the benchmark environments do not appear in the training split.

Task types. *Basic*: the target is visible from the start pose and directly reachable. *Exploration*: the target is out of view at the start, so the agent searches the environment and commits to a direction before the target comes into view. *Dynamic Spatial-Semantic*: the target is visible but specified relationally or ordinally (*e.g.*, "the second chair from the left"), and the relation is defined in the initial view, which changes as the agent moves toward the target. *Multi-stage*: the task names two targets to be reached in order, with the second usually out of view while the first is pursued. *Open Door* and *Pick & Drop* are interaction tasks, in which the agent opens a door, or picks up an object and releases it beside a landmark, and success is judged on the door state and the object position rather than on the actions issued. Appendix B.6 shows one demonstration of each type.

3.2 Collection Protocol

The pipeline turns a sampled start pose into a task, a demonstration of that task, and a rationale for every demonstrated step, in that order, and each stage ends with a check that its artifact has to pass before the next stage builds on it. The annotator who checks an artifact is never the one who produced it. Four annotators carried out the human stages (interfaces in Appendix C).

Stage 1: Environment Sampling. Four annotators sampled 10,000 candidate start poses across the environments. Each candidate records the full 6-DoF state and an egocentric RGB frame, which is the view a task is written from.

Stage 2: Task Construction. Three annotators wrote the instructions, working through the candidates and skipping those they judged unsuitable for a task of the intended type. For a kept candidate,

the annotator moves through the environment from the egocentric view, fixes the start pose at which the task is well-posed, and writes the instruction. The fourth annotator checks every task from its start pose on three criteria. *Well-posedness*: the named target exists and the type constraint holds (visible for Basic, out of view for Exploration, a valid relation for Dynamic Spatial-Semantic, two targets for Multi-stage, an interactable door or object for the interaction types). *Reachability*: the target can be reached from the start pose. *Ambiguity*: no second object of the same class fits the description, and spatial relations are stated with respect to the initial view. A task that fails any criterion is rewritten or discarded. Claude Opus 4.8 then corrects grammar and spelling, and the annotator confirms the corrected instruction before it enters the dataset (standards in Appendix B.2).

Stage 3: Human Demonstration. Three annotators recorded the demonstrations, each completing a task by controlling the character from the egocentric view with the agent's discrete action set. The recorder logs the egocentric frame before every action, the action, the 6-DoF pose, and the door and object state streams. The check has two parts, in a fixed order. *Reproducibility*: the recorded action sequence is re-executed on a freshly loaded environment under the fixed simulator configuration used throughout this work, and the replayed agent pose (and, for Pick & Drop, the object pose) has to match the recorded pose at every step exactly. *Completion*: the fourth annotator reviews the recording and verifies that the task was accomplished, that is, the agent stopped at the named target, the door is open at `Finish`, or the object was picked up and released within 2.2 m of the landmark. A demonstration that fails either part is discarded and re-recorded from scratch.

Stage 4: Step-wise Reasoning Annotation. For every recorded step, Claude Opus 4.8 (Anthropic, 2026) writes a first-person rationale explaining why the demonstrated action is appropriate. It is given only what the agent could know at that moment, the instruction, the initial frame, the preceding frames, the current frame, and the action that was executed, and it never sees later frames or the remaining action list. Rationales are two to three sentences long, and format constraints on person, length, and reference to future steps are enforced automatically at generation time (prompt in Appendix B.3). The four annotators then verify every trajectory frame by frame against three criteria. *Grounding*: the objects and sides the rationale mentions are visible where it says they are. *Knowledge boundary*: the rationale asserts nothing that is unobservable at that step, such as the effect of the current action. *Action consistency*: the stated intent matches the executed action. In the first pass, 73% of the training trajectories passed all three criteria, 13% failed grounding, 20% failed the knowledge boundary, and 9% failed action consistency, where one trajectory can fail several. The 27% that failed any criterion were regenerated from the reviewer's note and passed a second check. As an independent check, two additional auditors who had not taken part in the verification made an accept-or-flag decision on 400 sampled trajectories under the same three criteria, 200 from the final set and 200 flagged during verification. The two sets were mixed, so the auditors did not know which set a trajectory came from, and each decided independently. Both auditors accepted all 200 trajectories from the final set and flagged all 200 that had been flagged during verification, naming the same failed criterion for all but 6 of them, so the verification decisions are reproducible by annotators who did not make them.

Each trajectory is released as one record: the instruction, and for every step the frame, the action, the pose, the door and object states, and, on the training split, the rationale (fields in Appendix B.4). Figure 3 shows the distributions of task types, trajectory lengths, and rationale lengths.

3.3 The Benchmark

Interface. An episode starts from the recorded start pose with the task instruction. At each step the agent receives the current egocentric frame together with the initial frame, the preceding frames, and the actions taken so far, and it returns one of the agent's discrete actions (prompt in Appendix B.5). An episode ends when the agent issues `Finish` or after 60 steps. Under the fixed simulator settings used throughout this work, a 20 Hz engine tick and 640×480 rendering, an action sequence replays to the same poses across devices.

Success regions. The region a task is judged against is reconstructed from its target rather than placed by hand. Color and depth are captured from a ring of viewpoints around the target and back-projected into a point cloud, and the points belonging to the target are kept by a connected-component test seeded at its pose. The footprint is the minimum-area rectangle enclosing their

projection onto the ground, and a thin rectangle on the fitted plane for doors, windows and wall patches. The success region is every ground position within 2.2 m of that footprint, and an annotator approved a rendered view of every region before its task entered the benchmark. Appendix D.2 gives the capture settings, the quality checks, and the radius used for Pick & Drop.

Success. An episode counts as a success only if the agent issues `Finish` while standing inside the success region of the target, at a height within 1 m of the height the demonstrator reached there, and orientation is not required. Multi-stage tasks additionally require that the last `MidwayTarget`, the action that declares arrival at the first target, was issued inside the region of the first target. Open Door tasks require the door to be open at `Finish`. Pick & Drop tasks require that the object was picked up, released, and lies inside the region of the landmark at `Finish`, with the agent inside its own region around that landmark.

Metrics. We report the standard metrics of R2R and VLN-CE (Anderson et al., 2018; Krantz et al., 2020): success rate (SR), oracle success (OS), success weighted by path length (SPL), navigation error (NE), and trajectory length (TL). Their adaptation to multi-stage and interaction tasks is given in Appendix D.3.

Reference rows. Result tables carry two reference rows: *Human*, the average of five participants who each performed every benchmark task through the same interface, and *Random*, a policy that draws each action uniformly at random from those available to the task.

4 Experiments

4.1 Setup

Models. Three closed-source models are evaluated zero-shot and serve as reference systems: GPT-5.5, Claude Opus 4.8, and Gemini 3.1 Pro Preview. Four open models of 2B to 9B parameters are evaluated zero-shot and after supervised fine-tuning on EmbRACE: Qwen3.5-2B, Qwen3.5-9B, InternVL3-8B, and GLM-4.1V-9B. All models use the interface of Section 3.3 with the same prompt, and the open models are served with vLLM on one GPU each. Causal statements in this section compare a model before and after training on the same base, and the closed-source rows are reference points only.

Training. Each training trajectory is unrolled into one sample per step. A sample contains the instruction, the initial frame, the preceding frames, and the actions taken so far, and its target is the rationale of that step followed by the action. The model is trained for three epochs, updating all language-model parameters and keeping the vision encoder frozen, with further details in Appendix E. The Actions only variant is trained on the same samples with the rationale removed from the target and is otherwise identical.

Metrics. Success rate, oracle success, SPL, navigation error, and trajectory length are reported per task type on the 686 benchmark tasks, as defined in Section 3.3, together with the Human and Random reference rows.

4.2 Zero-Shot Results

Closed-source models. Table 2 reports the zero-shot results of the three closed-source models by task type. Among the six types, the three models score highest on Basic, where the target is visible from the start pose. They score lower on Exploration, Dynamic Spatial-Semantic, and Multi-stage, the three types built around the behaviors of Figure 2. On Exploration and Multi-stage all three succeed in fewer than half of the tasks. On the two interaction types the three models differ more from one another than on the navigation types. The five participants solve nearly all tasks on the six types through the same interface, and the Random policy solves none.

Open models. Zero-shot, the four open models of 2B to 9B parameters stay below 15% success on the six task types. Their oracle success is also below 25%, so most of their episodes do not pass through the success region at any step.

Table 2: Performance comparison on Basic, Exploration, Dynamic Spatial-Semantic, Multi-stage, Open Door and Pick&Drop tasks.

Method	Basic (116)					Exploration (117)					Dynamic Spatial-Semantic (114)				
	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑
Reference															
Random	10.83	11.97	0.0	0.0	0.0	10.84	9.79	0.0	0.0	0.0	10.27	12.43	0.0	0.0	0.0
Human	12.82	0.58	100.0	100.0	86.7	15.98	0.57	99.1	99.1	83.8	13.56	0.59	100.0	100.0	86.8
Closed-source VLMs (zero-shot)															
GPT-5.5 (OpenAI, 2026)	17.47	2.03	75.9	72.4	61.3	30.64	14.19	41.9	39.3	26.8	19.42	2.64	67.5	61.4	53.3
Claude Opus 4.8 (Anthropic, 2026)	13.69	2.22	59.5	59.5	55.4	25.96	8.57	43.6	43.6	35.9	14.63	3.12	54.4	51.8	45.0
Gemini 3.1 Pro Preview (Google DeepMind, 2026)	12.22	2.27	67.2	67.2	59.5	28.77	12.19	46.2	43.6	32.3	15.13	4.02	58.8	57.0	49.2
Open-source VLMs (zero-shot)															
Qwen3.5-2B (Qwen Team, 2026)	39.78	10.35	18.1	8.6	6.8	52.92	22.48	1.7	0.9	0.6	38.72	12.95	6.1	3.5	3.4
Qwen3.5-9B (Qwen Team, 2026)	20.26	8.89	23.3	12.1	10.8	49.01	22.01	4.3	3.4	2.7	18.94	9.30	20.2	7.9	6.8
InternVL3-8B (Zhu et al., 2025)	25.34	10.10	19.0	10.3	8.0	47.26	22.12	2.6	1.7	1.2	22.98	11.12	19.3	6.1	4.9
GLM-4.1V-9B (GLM-V Team, 2025)	36.29	7.77	21.6	13.8	12.2	58.08	22.54	3.4	3.4	1.9	30.88	9.97	17.5	7.0	6.3
Trained on EmbRACE (SFT)															
Qwen3.5-2B (Qwen Team, 2026)	14.13	2.05	70.7	69.8	61.4	20.13	10.24	35.9	35.9	28.4	12.93	3.82	57.9	55.3	50.7
Qwen3.5-9B (Qwen Team, 2026)	11.96	1.61	83.6	80.2	72.2	19.88	6.56	55.6	54.7	50.5	13.14	2.60	73.7	67.5	59.6
InternVL3-8B (Zhu et al., 2025)	13.62	2.16	76.7	76.7	69.9	24.14	7.06	52.1	49.6	44.5	12.55	2.67	70.2	64.0	55.9
GLM-4.1V-9B (GLM-V Team, 2025)	12.42	1.95	80.2	75.9	68.4	19.88	7.35	51.3	46.2	38.5	12.39	2.70	68.4	66.7	58.7

Method	Multi-stage (106)					Interaction - Open Door (109)					Interaction - Pick and Drop (124)				
	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑
Reference															
Random	11.74	11.27	0.0	0.0	0.0	11.12	5.73	0.0	0.0	0.0	10.98	8.25	0.0	0.0	0.0
Human	26.65	0.48	100.0	99.1	84.3	8.96	0.54	100.0	100.0	78.8	15.15	1.09	100.0	100.0	80.0
Closed-source VLMs (zero-shot)															
GPT-5.5 (OpenAI, 2026)	26.80	4.15	57.5	35.8	28.2	18.71	0.98	67.9	67.9	54.9	11.60	3.92	58.9	54.8	47.1
Claude Opus 4.8 (Anthropic, 2026)	24.50	5.30	42.5	33.0	31.0	14.27	1.57	44.0	44.0	36.0	19.96	3.47	42.7	40.3	34.3
Gemini 3.1 Pro Preview (Google DeepMind, 2026)	20.01	4.51	51.9	38.7	32.5	14.14	1.28	65.1	65.1	50.1	16.27	2.90	61.3	61.3	55.9
Open-source VLMs (zero-shot)															
Qwen3.5-2B (Qwen Team, 2026)	49.32	14.89	1.9	0.9	0.7	32.20	7.47	6.4	4.6	3.7	43.71	9.03	1.6	0.8	0.7
Qwen3.5-9B (Qwen Team, 2026)	28.36	11.99	8.5	7.5	5.5	30.07	4.47	10.1	9.2	8.7	26.13	6.40	6.5	6.5	5.9
InternVL3-8B (Zhu et al., 2025)	39.71	13.38	5.7	3.8	2.8	16.06	6.72	13.8	10.1	8.0	24.68	5.91	3.2	3.2	2.7
GLM-4.1V-9B (GLM-V Team, 2025)	50.00	16.22	10.4	7.5	5.8	33.95	3.93	12.8	11.9	11.6	18.97	6.30	7.3	5.6	4.5
Trained on EmbRACE (SFT)															
Qwen3.5-2B (Qwen Team, 2026)	28.07	5.63	47.2	42.5	37.3	14.28	1.37	73.4	69.7	61.1	15.14	2.88	66.1	66.1	55.1
Qwen3.5-9B (Qwen Team, 2026)	25.54	3.89	56.6	52.8	46.8	13.51	1.09	89.0	85.3	66.1	14.48	2.67	76.6	69.4	59.0
InternVL3-8B (Zhu et al., 2025)	29.97	5.84	51.9	45.3	41.3	13.35	1.42	76.1	75.2	59.2	16.69	3.51	62.1	56.5	49.4
GLM-4.1V-9B (GLM-V Team, 2025)	27.25	4.14	58.5	53.8	46.9	13.73	1.55	86.2	84.4	66.4	15.14	2.94	67.7	67.7	57.2

Table 3: Effect of the rationale in EmbRACE.

Metric	Training data	Basic	Explore	Dynamic	Multi	Open	P&D
SR (%) ↑	Actions only	73.3	45.3	50.0	42.5	67.0	62.1
	Actions + rationale	80.2	54.7	67.5	52.8	85.3	69.4
	Δ	+6.9	+9.4	+17.5	+10.3	+18.3	+7.3
TL (steps)	Actions only	14.3	25.8	17.5	30.7	17.4	18.1
	Actions + rationale	12.0	19.9	13.1	25.5	13.5	14.5
	Δ	-2.3	-5.9	-4.4	-5.2	-3.9	-3.6

Table 4: Action quality metrics.

Method	Block ↓	Revisit ↓	Recovery ↑
GPT-5.5	16.9	24.9	33.8
Claude Opus 4.8	10.6	31.8	37.9
Gemini 3.1 Pro Preview	12.5	23.2	42.0
Qwen3.5-9B (Origin)	22.2	43.9	1.0
Qwen3.5-9B (Actions only)	14.7	29.2	40.5
Qwen3.5-9B (SFT)	5.6	13.6	70.7

4.3 Effect of Training on EmbRACE

Four base models. After supervised fine-tuning on EmbRACE, the four open models go from 3% to 8% overall to 57% to 68%, against 99.7% for the human participants. The gain is on every task type and on every backbone from 2B to 9B parameters. On the three types built around the behaviors of Figure 2 the four models gain 35% to 60%, and the largest gains, 65% to 76%, are on Open Door. Exploration and Multi-stage gain the least and remain the two lowest types for the four fine-tuned models, more than 40% below the human participants. We also measure how long a model takes to act. The fine-tuned models are served on one GPU each, where the first fourteen decisions of an episode, the median length of an episode on the benchmark, take 37 s with the 2B model and 51 s with the 9B model. For reference, the three closed-source models take 79 to 123 s for the same fourteen decisions, including the API round trip.

Effect of the rationale. Table 3 trains the same base on the same trajectories with and without the rationale in the target. The rationale raises the success rate on the six task types by 7% to 18%, with the largest gains on Dynamic Spatial-Semantic and Open Door. It also shortens trajectories on the six types by 16% to 25%, so the models succeed more while moving less.

Table 5: Real-robot results. Success rate over five tasks per cell, each run five times. The last row replaces the training corpus of the row above it, with the same base model, the same full-parameter recipe, and the same number of reasoning steps.

Method	Workspace					Home					Outdoor				
	Basic	Explore	Dynamic	Multi	P&D	Basic	Explore	Dynamic	Multi	P&D	Basic	Explore	Dynamic	Multi	P&D
Closed-source VLMs (zero-shot)															
GPT-5.5	1.00	0.20	0.40	0.20	0.04	0.80	0.20	0.20	0.40	0.20	0.80	0.00	0.40	0.00	0.20
Claude Opus 4.8	0.92	0.44	0.20	0.20	0.00	0.80	0.40	0.20	0.40	0.00	1.00	0.20	0.40	0.20	0.00
Gemini 3.1 Pro Preview	1.00	0.20	0.40	0.40	0.20	0.80	0.20	0.20	0.40	0.20	1.00	0.20	0.40	0.40	0.00
Open-source VLM															
Qwen3.5-9B (Origin)	0.20	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00
Qwen3.5-9B (SFT)	1.00	0.84	1.00	0.80	0.80	1.00	0.80	0.80	1.00	0.60	1.00	0.80	1.00	0.80	0.60
Qwen3.5-9B (SFT on R2R-CoT)	1.00	0.00	0.00	0.24	N/A	0.80	0.20	0.00	0.20	N/A	0.80	0.00	0.00	0.00	N/A



Figure 5: Real-world qualitative comparison of different VLMs. The **red** parts illustrate the incorrect reasoning.

4.4 Beyond Success Rate

Success rate records the end of an episode and not its course. Table 4 reports three measures of the course for six of the models. *Block* is the fraction of move actions that do not move the agent, counted from the third such move in a row. *Revisit* is the fraction of episodes in which the agent takes two consecutive steps back to views it has already seen. *Recovery* is computed over the episodes in which an `OpenDoor` leaves the door closed or a `Pick` leaves the object unheld, and is the fraction of them in which the door is open or the object is held at a later step. Definitions are given in Appendix D.4. On the same base, training on EmbRACE lowers Block from 22.2% to 5.6% and Revisit from 43.9% to 13.6%, and raises Recovery from 1.0% to 70.7%. The Actions only variant reaches a success rate close to the closed-source models, 56.9 against 45.5% to 55.7%, and its Block and Revisit fall inside their range as well, 14.7 against 10.6% to 16.9% and 29.2 against 23.2% to 31.8%. Adding the rationale to the same trajectories takes both below that range, to 5.6 and 13.6%. It also raises Recovery from 40.5% to 70.7%, so a door or a pick that the first attempt left undone is more often completed later.

4.5 Beyond the Simulator

Physical robot. The same models are also run outside the simulator, on a physical robot. The robot is a wheeled base carrying a front-facing RGB camera and a gripper, controlled by a Raspberry Pi 5. It is run in three scenes, a workspace, a home, and an outdoor area, on five tasks of each of five types in each scene, each run five times, 375 episodes per model. The 9B model fine-tuned on EmbRACE succeeds in 86% of its episodes, against 4% for the same model before fine-tuning and 34% to 40% for the three closed-source models. It succeeds in every Basic task and in 0.6 to 0.8 of the Pick & Drop tasks. The closed-source models succeed in most Basic tasks, in fewer than half of the Exploration, Dynamic Spatial-Semantic and Multi-stage tasks, and in at most one Pick & Drop task in five in any scene. Table 5 reports success per scene and type, and Appendix F gives the platform, the action mapping, the success criterion, and the failure modes. Figure 5 shows one Dynamic Spatial-Semantic task in the workspace, reaching the second closest red sofa, for GPT-5.5 and for the 9B model before and after fine-tuning.

Table 6: Cross-benchmark results on VLN-CE, 100 episodes sampled from each Val-Unseen split.

Method	R2R Val-Unseen					RxR Val-Unseen				
	TL	NE↓	OS↑	SR↑	SPL↑	TL	NE↓	OS↑	SR↑	SPL↑
GPT-5.5	7.23	6.86	31.0	28.0	25.7	8.53	7.91	24.0	19.0	15.7
Claude Opus 4.8	7.85	6.81	30.0	30.0	26.3	8.94	8.04	22.0	19.0	15.5
Gemini 3.1 Pro Preview	7.62	6.64	30.0	26.0	23.8	6.30	8.13	22.0	17.0	15.0
Qwen3.5-9B (Origin)	4.21	8.36	10.0	6.0	4.9	4.33	9.31	5.0	5.0	4.2
Qwen3.5-9B (SFT)	7.05	6.55	37.0	32.0	29.5	5.81	7.76	28.0	22.0	19.4

Table 7: Success rate (%) after SFT and after GRPO.

Task type	SFT	GRPO
Basic	80.2	79.3
Exploration	54.7	50.3
Dynamic	67.5	64.9
Multi-stage	52.8	52.8
Open Door	85.3	79.8
Pick & Drop	69.4	67.8

Comparison with an existing reasoning corpus. The last row of Table 5 changes only the training corpus. The same Qwen3.5-9B is fine-tuned with the same recipe for three epochs on 48K reasoning steps sampled from R2R-CoT (Wang et al., 2025), the size of the EmbRACE training split. Both models are run on the same tasks. Neither corpus was collected on this robot. On Basic both models succeed in most tasks. On Exploration and Dynamic Spatial-Semantic the model trained on R2R-CoT succeeds in at most one task in five in any scene while the model trained on EmbRACE succeeds in three or more. On Multi-stage the model trained on R2R-CoT succeeds in at most one task in four. Pick & Drop is not compared, since R2R-CoT has no object actions.

Benchmarks in real-scan environments. We also run a cross-benchmark evaluation on VLN-CE R2R and RxR (Krantz et al., 2020), which are built on scans of real buildings, so the views a model receives are photographs rather than renderings. The two benchmarks ask for route following from a language instruction. The models are run through the same interface on 100 episodes sampled at random from each Val-Unseen split, following Zhang et al. (2024b). The fine-tuned models are not trained on either split, nor does EmbRACE contain any data from these environments. Training on EmbRACE raises success from 6.0% to 32.0% on R2R and from 5.0% to 22.0% on RxR. It raises SPL from 4.9 to 29.5 on R2R and from 4.2 to 19.4 on RxR. The gain therefore holds on photographs and on a task format that EmbRACE does not contain.

4.6 GRPO after Supervised Fine-Tuning

The recipe released with EmbRACE is supervised fine-tuning alone. Table 7 continues training the fine-tuned checkpoint with GRPO on trajectories not used for supervised fine-tuning, with two rewards, one for the required output format and one for matching the demonstrated action. On these trajectories the fine-tuned model already produces the demonstrated action in 93% of the sampled completions. Most groups therefore contain no spread in reward, which is the condition under which the group advantage vanishes (He et al., 2026). Resampling groups without spread, up to three times per batch, does not change this. Success on the six task types accordingly stays within 5.5% of the fine-tuned checkpoint. Going further may require a reward defined not on the demonstrated action but on the agent's position and the door and object states.

5 Conclusion

We present EmbRACE, a dataset of human demonstrations recorded from the egocentric view, with a rationale at every step stating what the demonstrated action was based on, and a benchmark of 686 tasks in environments held out from training. On the benchmark, frontier models score highest when the target is in view from the start and below half when it has to be searched for or returned to after leaving the view. Fine-tuning on EmbRACE takes four open models of 2B to 9B parameters from 3% to 8% overall to 57% to 68%. The gain holds on a physical robot and on benchmarks built from scans of real buildings. Relative to training on the same trajectories without them, the rationales raise success on the six task types and shorten the trajectories, so the record of what each action was based on changes behavior beyond the effect of the actions alone. To facilitate further work on this problem, the dataset, the benchmark, and the code will be made publicly available.

AI USE STATEMENT

Generative AI was used in this work in the roles reported below. Claude Opus 4.8 wrote the rationale of every step of the training split from the instruction, the frames up to that step, and the action taken at it, under the prompt of Appendix B.3. The annotators verified every rationale against the frames (Section 3.2). Claude Opus 4.8 also corrected the grammar and the spelling of the instructions written by the annotators, who confirmed every corrected instruction before it entered the dataset. Because Claude Opus 4.8 is also one of the evaluated models, Appendix D.5 evaluates the three closed-source models on rewrites of the benchmark instructions produced by GPT-5.5, Gemini 3.1 Pro Preview, and DeepSeek-V4-Flash. The authors used Claude Code, a coding assistant, to write and edit the code of this work, to produce the figures and the tables from the recorded results, and to polish the text of this paper. The authors reviewed all AI-assisted work, tested the code, checked the reported numbers against the recorded episodes, and revised the text. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Four annotators carried out the human stages of the collection protocol, two further auditors performed the independent check of Section 3.2, and five participants produced the Human reference row by performing the benchmark tasks through the same interface as the models. Their work consisted of operating a simulator and of reviewing rendered frames and text. No personal data were collected from them. Nothing in the released data identifies them. The environments are the synthetic scenes of UnrealZoo, obtained under its terms. The rendered frames show no real people and contain no personal data. The dataset and the benchmark will be released for research use and the code under an open-source license, under the terms given in Appendix B.7. Models trained on EmbRACE control an embodied agent, in the simulator and on the physical robot of Section 4.5. Their deployment outside a controlled setting requires safety measures and human oversight.

REPRODUCIBILITY STATEMENT

The dataset and the benchmark are constructed by the four-stage protocol of Section 3.2, with the instruction standards, the rationale prompt, and the record format given in Appendix B. Every demonstration was verified to replay exactly on a freshly loaded environment with the fixed simulator configuration of Section 3.3. With that configuration, an action sequence replays to the same poses across devices. The success regions, their reconstruction, and their checks are given in Appendix D.2, the five metrics in Appendix D.3, and the three action quality metrics in Appendix D.4. The interface prompt is given in full in Appendix B.5, the training and reinforcement fine-tuning recipes with their settings in Appendix E, and the robot platform, its tasks, and its success criterion in Appendix F. The supplementary material contains the benchmark code, that is, the simulator configuration, the evaluator, the scorer, and a playground, with six benchmark tasks and their success regions on which an evaluation runs end to end, and a viewer of example trajectories. The dataset, the benchmark, and the code will be made publicly available.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3674–3683, 2018.

Anthropic. Claude Opus 4.8 system card. `https://www.anthropic.com/claude-opus-4-8-system-card`, 2026. Accessed: 2026-09-02.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez,

Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. *arXiv preprint arXiv:2206.06994*, 2022.

Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, Ranjay Krishna, Dustin Schwenk, Eli VanderBilt, and Aniruddha Kembhavi. SPOC: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. *arXiv preprint arXiv:2312.02976*, 2023.

GLM-V Team. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Google DeepMind. Gemini 3.1 Pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026. Accessed: 2026-09-02.

Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. In *International Conference on Machine Learning (ICML)*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-R1: Reasoning and navigation in embodied scenes. *arXiv preprint arXiv:2509.10884*, 2025.

Yuqi Liu, Shengju Qian, Tianyuan Qu, Mingxian Lin, Zixuan Wang, Xin Wang, Bei Yu, and Jiaya Jia. MemVLN: Episodic and procedural memory for vision-and-language navigation. *arXiv preprint arXiv:2607.23504*, 2026.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. OpenEQA: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

OpenAI. GPT-5.5 system card. `https://openai.com/index/gpt-5-5-system-card/`, 2026. Accessed: 2026-09-02.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. TEACh: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2017–2025, 2022.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8494–8502, 2018.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, 2026. Accessed: 2026-09-02.

Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-Web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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*, pp. 9339–9347, 2019.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10740–10749, 2020.

Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. EMMA-X: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14199–14214, 2025.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. *arXiv preprint arXiv:2308.12952*, 2023.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-Think: Exploring reasoning strategies for data-efficient vision-language navigation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.11886.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *International Conference on Machine Learning (ICML)*, 2025.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *Robotics: Science and Systems (RSS)*, 2024b. arXiv:2402.15852.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. *arXiv preprint arXiv:2408.05517*, 2024.

Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. Unreal-Zoo: Enriching photo-realistic virtual worlds for embodied AI. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2412.20977.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Closed-Loop Behaviors of Vision-Language Models

Setup. Four models are run on instruction-following tasks in UnrealZoo environments (Zhong et al., 2025), of the kinds described in Section 3: GPT-5.5, Claude Opus 4.8, and Gemini 3.1 Pro Preview zero-shot, and Qwen3.5-9B fine-tuned on the VLN-CE R2R training split (10,819 trajectories) with the supervised fine-tuning recipe of Appendix E, using action labels only and no rationales. The fourth model tests whether action-level training removes the behaviors seen in the zero-shot models. All four follow the closed-loop protocol of the benchmark: given a task instruction, a model sees the initial frame, the preceding frames, and the actions so far, and returns one action at a time. Figure 2 in the main text and Figure 6 here show representative cases.

Short-sighted exploration. The search does not accumulate across steps, and models return to views already seen instead of moving on to new ones. In Figure 2(a) the agent alternates left and right turns through the same views and the target does not appear. The zero-shot models mostly turn in place, alternating between the same two or three views. The action-trained model moves more, along open paths, but the target does not appear either.

Fragmented goal pursuit. Under partial observability, models lose track of previously observed targets and fail to integrate observations across steps. In Figure 2(b) the agent approaches the red car, turns right as the car leaves the frame, and does not turn back to it. When a target that was visible leaves the frame during self-motion, the zero-shot models head elsewhere and do not turn back to it. The action-trained model keeps moving but does not turn back toward where the target was. In *Go to the flowerbed* (Figure 6, bottom), the agent approaches the flowerbed, overshoots it, and ends against a corner.

Spatial-semantic instability. Relations in the instruction are re-interpreted as the viewpoint changes. In Figure 2(c) the agent, asked to go to the second pillar, switches to a different pillar after a few steps. The zero-shot models head for the correct object in the first frames and, after a few steps, for a different one. The action-trained model shows the same switch. In *Go to the right light* (Figure 6, top), the agent first heads toward the correct light and, after the view changes, reaches the other light.



Figure 6: Additional cases. Top: spatial-semantic instability, the agent is asked to go to the right light and reaches the other light after the view changes. Bottom: fragmented goal pursuit, the agent overshoots the flowerbed and ends against a corner.

B More Details on Data Construction

This appendix lists the environments and gives the standards the instructions of Stage 2 were written to, the constraints the rationales of Stage 4 were generated under, the fields of a released record, the prompt of the benchmark interface, one example trajectory per task type, and the terms of the release.

Table 8: The 62 environments of EmbRACE. The setting states where the tasks of an environment take place and under which lighting. A mixed environment holds both indoor and outdoor areas. The released data name each environment by its UnrealZoo identifier.

Environment	Setting	Environment	Setting
Training split (55 environments)			
Abandoned District	Outdoor, day	Modular Neighborhood	Mixed, day
Asian Medieval City	Outdoor, day	Museum	Indoor
Asian Temple	Indoor	Old Factory	Mixed, day
Bazaar	Outdoor, day	Old Town	Mixed, day
Brass Palace	Outdoor, day	Opera House	Indoor
Brass Streets	Outdoor, day	Post-Soviet Village	Outdoor, day
Bunker	Indoor	Pyramid	Outdoor, day
Castle	Outdoor, day	Racing Track	Outdoor, day
Chemical Plant	Outdoor, day	Roof	Outdoor, day
Chinese Water Town	Outdoor, day	Rural Australia 1	Outdoor, day
Command Center	Indoor	Rural Australia 2	Outdoor, day
Container Yard (Day)	Outdoor, day	Rural Australia 3	Outdoor, day
Container Yard (Night)	Outdoor, night	Russian Winter Town	Outdoor, day
Desert Colosseum	Outdoor, day	Science Fiction Valley Town	Outdoor, night
Desert Ruins	Outdoor, day	Stadium	Outdoor, night
Eastern Garden	Outdoor, day	Storage House	Indoor
EF Grounds	Outdoor, day	Suburb Neighborhood (Day)	Mixed, day
EF Gus	Outdoor, day	Suburb Neighborhood (Night)	Mixed, night
EF Lewis	Outdoor, day	Supermarket	Indoor
English College	Outdoor, day	Swimming Pool	Indoor
Farm	Outdoor, day	Tokyo	Outdoor, day
Forest Gas Station	Mixed, day	Underground Parking	Indoor
Holding Cells	Indoor	Venice	Outdoor, day
Hospital	Indoor	Victorian Train Station	Indoor
Hotel Corridor	Indoor	Watermills	Outdoor, day
Japan Train Station	Outdoor, day	Western Garden	Outdoor, day
Medieval Castle	Outdoor, day		
Middle East	Outdoor, day		
Modular Building	Mixed, day		
Held out for the benchmark (7 environments)			
Courtyard	Outdoor, day	Medieval Village (Day)	Outdoor, day
Downtown West	Outdoor, day	Medieval Village (Night)	Outdoor, night
Greek Island	Outdoor, day	Temple Plaza	Outdoor, day
Industrial Area	Mixed, day		

B.1 Environments

Table 8 lists the 62 environments, the 55 of the training split and the 7 held out for the benchmark. Each is one UnrealZoo scene, named here by its subject. The setting records where the tasks of an environment take place, indoors, outdoors, or in both, under daytime or night lighting.

B.2 Instruction Standards

The instructions of Stage 2 (Section 3.2) were written to the standards below, which fix what may serve as a target, how the target is described, and what each task type adds.

Target. A target is a precise spot with a small ground footprint. It is either a single object that the agent can walk up to and stand beside or else one specific part of a larger structure, *e.g.*, the doorway under an archway, the base of a tower, or one ground-floor window. Excluded are surfaces and open areas (a lawn, a road, a plaza), extended structures (a hedge, a fence, a wall, a railing), whole buildings, and anything the agent cannot stand beside at ground level (a rooftop feature, an upper-floor window). An object within one step of the start pose is not a target.

Description. An instruction names the target with the fewest words that make it unique in the start-pose frame. A target that cannot be described uniquely is replaced by another. The instruction states only what is visible in that frame, so it names no floor, room, or route that the frame does not show. A spatial relation such as beside or behind appears only where the visible geometry supports it. Words that fix a side or an order, such as left, right, first, or nearest, carry the relation of a Dynamic Spatial-Semantic task and are otherwise used only to separate two like objects that are both in view. Across the tasks written from one start pose, the targets are objects of different kinds spread over the scene. Verbs and sentence openings vary from task to task.

Basic. The task names one target that is clearly visible in the start-pose frame.

Exploration. The target is absent from the start-pose frame and present in one of the views to the right, behind, or to the left of it, which the annotation tool renders from the same pose. The instruction has two explicit actions, to find the target and to go to it, *e.g.*, "Locate the small pushcart topped with a yellow-and-red umbrella on the plaza, then navigate over to it".

Dynamic Spatial-Semantic. The task names one of several like objects in view by an ordinal or a side relation that holds in the start-pose frame.

Multi-stage. The task names two targets in a fixed order, both visible in the start-pose frame, in the form "[verb] A, then [verb] B".

Open Door. The task names a navigable door or doorway at ground level and tells the agent to go to it and open it. A double door counts as one door. Cabinet doors, appliance doors, windows, and gates do not qualify. When the door is not in the start-pose frame, the instruction adds a short search before it, *e.g.*, "Turn away from the tall built-in wardrobe, locate the bedroom door, and open it".

Pick & Drop. The task assigns an object drawn at random, independently of the environment, from a fixed list of ten household objects (tomato, green bell pepper, wicker basket, ceramic pot, jerry can, ceramic vase, plate, candelabra, metal bucket, trash bag). The object is placed at the source when the task is set up, so it is absent from the frame the instruction is written from. The annotator chooses the source and a different destination among visible landmarks at ground level on which an object can rest. The instruction states the pickup with pick up, lift, or grab and gives the source as a modifier of the object rather than as a separate navigation step. A placement follows then, stated with place, set down, drop, leave, or put down. No transport clause such as carry or bring stands between the two actions.

B.3 Rationale Generation

The rationales of Stage 4 are generated by Claude Opus 4.8 under the prompt of Figure 7. For a step, the model is given the instruction, the same fourteen frames that the benchmark interface gives a policy at that step, the initial frame and the thirteen frames before the action, and the action that was executed, which it explains rather than chooses. The prompt fixes the perspective, the length, and the knowledge boundary of the rationale.

A rationale is written in the first person at the moment immediately before the action, in two or three sentences of at most 55 words. The rationale of `Finish` is two sentences of at most 30 words in all, one naming the visible or persistent state that satisfies the task and one stating that the task is complete. A rationale may assert only what the instruction and the frames it was given establish before the action. The effect of the current action and of every later one is expressed only as an expectation, with should, may, likely, try, or attempt. After a `Pick`, a `Drop`, or an `OpenDoor`, the first sentence of the next rationale reports the state of the door or the object as the current frame shows it. A `MidwayTarget` rationale names what shows that the first destination has been reached and where the agent heads next. An ordinal or a superlative in the instruction is resolved once in the initial frame, after which the target is referred to by its own appearance. Every left or right is checked against the frame before the action and the direction of the demonstrated turn. The rationale refers to no frame, image, index, or annotation process.

You are an offline expert annotator creating grounded supervised rationales for demonstrated embodied-agent actions. You are not selecting the action: the demonstrated current action is a fixed condition that you must explain.

Write each rationale as the embodied agent interacting with the environment through its own first-person ego view immediately before the current action. The narrator has the initial ego view, up to the twelve ego views immediately preceding the current step, and the current pre-action view, for at most 14 unique ego views. These are concise action explanations, not hidden model thoughts. Return strict JSON and no markdown.

TASK: {instruction}
TASK TYPE: T{type}
CURRENT ACTION: {action}

TYPE-SPECIFIC FOCUS:
{one line for the task type, e.g. Relational navigation: preserve the ordinal or spatial relation that disambiguates the target.}

EGO OBSERVATIONS:
The attached visual observations are the agent's own first-person experience, in order: the initial task view, the views immediately preceding the current step, and the current pre-action view. Never mention them, their numbers, or the visual medium in a rationale.

Write a rationale of exactly 2 or 3 complete English sentences from the agent's first-person perspective at the moment immediately before the action is executed. A rationale for a non-Finish action must not exceed 55 English words. Say what is decisive and stop; do not pad with restated scenery.

- Ground the explanation primarily in the current ego state. Do not write travelogue-style summaries of cumulative milestones, such as claiming that I crossed the lawn and stepped onto the porch, when the visible door and my current position already justify the interaction.
- The initial observation preserves the original task and target context. It does not give me continuous episodic memory of every intermediate event.
- Finish is a completion decision, not a summary. For a Finish action, write exactly two short sentences totaling at most 30 English words. The first sentence must identify only the decisive current visual or persistent-state evidence that satisfies the task's success condition. The second must state that the task is complete and justify stopping.

CAUSAL KNOWLEDGE BOUNDARY FOR RATIONALE:

- The rationale may assert only facts available from the task, the ego observations above, and outcomes visible inside them before the current action.
- Do not assert the result of the current action or any later action as a fact. Express an unobserved immediate effect only as an expectation or hypothesis with wording such as "should", "may", "likely", "try", or "attempt".
- You may use "I will" to restate execution of the fixed current action. Do not use certainty claims such as "will succeed", "will be successful", or "will successfully" for an outcome that has not yet occurred.
- MidwayTarget means I have arrived at the earlier of my two destinations. Write it in three beats: what I see that tells me I am there, that I have reached it, and that I mark reaching it before setting off for my other destination. Name both destinations by what they are, the way I would point at them.
- An ordinal or superlative in my instruction, such as the second door from the left or the rightmost bench, is how I pick my target out once. Settle the count in my opening view, and in the same breath pin the target to whatever actually tells it apart, so that from then on I can follow that thing rather than a position in a row. After that, refer to my target by what it is.
- Check every left/right claim against the ego view immediately before the action and the demonstrated turn. Do not say a target is on the right and justify TurnLeft as turning toward it, or say it is on the left and justify TurnRight as turning toward it.
- Never refer to frames, images, screenshots, observation numbers, action indices, a trajectory, a demonstration, an annotator, or an annotation process.
- Never invent an obstacle, transition, successful door opening, successful grasp, or successful drop without evidence.

Return this exact JSON shape:

```
{"annotations": [{"index": 0, "action": "MoveForward", "rationale": "..."}]}
```

Figure 7: The prompt of Stage 4, abridged to its inputs and constraints. The instruction, the task type, and the executed action are filled in for the step. The frames are attached in the order given under the ego observations.

B.4 Record Format

Each trajectory is released as one directory that holds its frames and one record with the fields of Table 9. The frames are the egocentric frame before every action and the frame after the last one, so a demonstration of n actions has $n + 1$ frames. The pose and state streams have $n + 1$ entries aligned with the frames. Poses are given in the coordinate frame of the environment, in centimeters and degrees. On the training split, the rationale of every step is released with the record.

B.5 Interface Prompt and Training Target

Figure 8 gives the system prompt of the benchmark interface as it is sent, with the instruction of the task in place of `{instruction}`, followed by the texts of the three user turns. The initial frame is a user turn, followed by the action taken from it as an assistant turn. The retained frames and their actions alternate in the same way. The current frame closes the conversation with the request for exactly one action. The reply carries the rationale in `<reasoning>` tags and the action in `<action>` tags. A supervised training sample lays out the same conversation for one step of a demonstration, with the rationale and the action of that step as the answer. The Actions only variant

Table 9: Fields of a released trajectory record. A stream marked $n + 1$ has one entry per frame, before each of the n actions and after the last one.

Field	Content
`Instruction`	the instruction
`Type`	the task type, 0 (Basic) to 5 (Pick & Drop)
`Start_Pose`	position and orientation of the agent at the start
`PD_ID`, `PD_Size`	Pick & Drop only, the object and its scale
`PD_Start_Pose`	Pick & Drop only, the pose at which the object is placed
`000.jpg`, `001.jpg`, ...	the egocentric frames ($n + 1$)
`Action`	the demonstrated actions (n)
`Pose_Trajectory`	the agent pose ($n + 1$)
`Door_State_Trajectory`	Open Door only, the door state, 0 closed and 1 open ($n + 1$)
`PD_Pose_Trajectory`	Pick & Drop only, the object pose ($n + 1$)
`PD_Dist_2D`	Pick & Drop only, the horizontal distance from the agent to the object ($n + 1$)
`Rationale`	training split only, the rationale of each step (n)

drops the rationale from the answer and from every retained assistant turn. Its prompt asks for `<action>` alone.

B.6 Example Trajectories

Figures 9 to 14 show one trajectory per task type from the training split, each with its instruction and, for the steps shown, the frame before the action, the action, and its rationale. A trajectory longer than six steps is shown by six of them, including the first and the last.

B.7 Data Release

The dataset, the benchmark, and the code will be made publicly available. The dataset and the benchmark, that is, the task files, the trajectory records of Table 9 with their frames and rationales, and the success regions, will be released under the Creative Commons Attribution-NonCommercial 4.0 license, for research use with attribution. The code, that is, the annotation interfaces, the benchmark harness, and the training recipes, will be released under the Apache 2.0 license. The environments are those of UnrealZoo (Zhong et al., 2025) and are obtained from its release under its terms. The release will be versioned, with corrections to the data recorded in its changelog.

C Annotation Interfaces

Figures 15 to 18 show the interfaces of the four stages of Section 3.2. In the interfaces of Stages 1 to 3, the annotator controls the character in a running environment from its egocentric view. The interface of Stage 4 shows the frames recorded in Stage 3.

In Stage 1 (Figure 15), the annotator moves the character with the agent's actions or directly to a typed pose and adds the current pose as a candidate start pose. In Stage 2 (Figure 16), the annotator opens a candidate and either skips it or saves a task with the start pose fixed from the egocentric view, the task type, and the instruction written from the start-pose frame. In Stage 3 (Figure 17), the annotator performs the task with the agent's discrete action set and saves the demonstration after `Finish` or discards it. In Stage 4 (Figure 18), the annotator steps through the trajectory with the frames before and after each action, checks each rationale against the three criteria, and accepts the trajectory or flags it with a note from which its rationales are regenerated.

You are an autonomous embodied agent operating in a 3D environment.

TASK: "{instruction}"

MESSAGE STRUCTURE:
- The first user message contains the initial task observation.
- An assistant response following it is the action originally chosen from that initial observation.
- The remaining completed user/assistant turns form the retained recent sliding-window segment in chronological order.
- Intermediate interactions between the initial interaction and the retained recent segment may have been omitted.
- The final user message has no assistant response and contains the observation from which you must act now.

Use the task, the initial interaction, and the retained recent conversation to choose exactly one next action.

FULL ACTION HISTORY STATUS:
- `MidwayTarget` has not appeared in the complete action history, including interactions outside the retained sliding window.

Available actions: `MoveForward`, `MoveBackward`, `TurnLeft`, `TurnRight`, `LookUp`, `LookDown`, `OpenDoor`, `Pick`, `Drop`, `MidwayTarget`, `Finish`

Action rules:
- `MoveForward` and `MoveBackward` move one discrete step; `TurnLeft` and `TurnRight` rotate in place; `LookUp` and `LookDown` tilt the view.
- `OpenDoor` opens a door near you. It does not reach far, so approach the door first.
- `Pick` picks up an object near you. It does not reach far, so approach the object first.
- Right after `Pick`, the view swings down to your own hands for that one observation so you can see what you are holding, and it returns to normal on your next action. Read that frame: empty hands mean the pick failed and the object is still where it was.
- `Drop` releases the object you are carrying, leaving it where you are.
- Use `MidwayTarget` only when the TASK names at least two ordered destinations, and only after reaching the earlier one while the TASK still requires continuing to a different final destination.
- If the TASK names only one destination, never use `MidwayTarget`.
- Do not use `MidwayTarget` to report progress, and do not use it in place of `OpenDoor`, `Pick` or `Drop`.
- Reaching a target means standing against it, not seeing it.
- Stopping short fails the task, while one extra `MoveForward` costs almost nothing. Whenever you are not certain you are already touching the target, take another `MoveForward` instead of `Finish`.
- Only `Finish` at a destination after a `MoveForward` has left the view essentially unchanged, showing that the target is physically stopping you. If what blocks you is not the target, turn and go around it.
- `Finish` is terminal: use it only when the task's required final state is satisfied.
- The action must be exactly one name from the list above (case-sensitive). Do not invent actions.

Output exactly:

```
<reasoning>Brief visual reasoning in at most 4 sentences.</reasoning><action>ACTION</action>
```

User turn carrying the initial frame

Initial task observation.

User turn carrying each retained frame

Observation obtained after the previous assistant action.

User turn carrying the current frame

This is Current. Choose exactly one next action.

Figure 8: The system prompt of the benchmark interface and the texts of the three user turns. The line on the action history reads "has already appeared" once `MidwayTarget` has been issued.

D More Experiments

D.1 More Details on Experiment Setup

Observation window. A step is given the initial frame and the thirteen frames before it, at most fourteen images. The window is sized so that an agent turning in place keeps the whole circle around it within the frames it still has.

D.2 Target Regions

Distances in the simulator are reported in engine units, and one unit is one centimeter. Every published region is a set of polygons in the world ground plane, together with the boundaries of any holes and a reference height. Figure 19 shows three regions, each with the reconstructed target, its footprint, and the region in the ground plane and over the scene.

Reconstruction. A free camera is placed on a ring around the target, at a radius of 3.8 m and 1.9 m above it, and captures color and depth from twelve viewpoints with a 70 degree horizontal field of view. Each frame is back-projected with a pinhole model, keeping depths between 0.2 m and 25 m, and the frames are fused on a voxel grid. The points belonging to the target are selected geometrically. The bounds of the target actor give a search window, the points inside it are binned on

Basic Walk to the bench located next to the phone booth. 6 steps



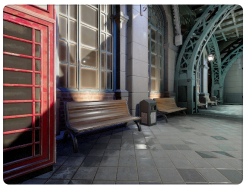
1 MoveForward
I can see the red phone booth ahead against the station wall, with a wooden bench immediately to its right, so that bench is my target. My heading already points down the walkway toward it, so I will step forward to begin closing the gap.



2 MoveForward
The phone booth and the bench beside it now appear larger and lower in my view, confirming I am walking straight at them. The tiled floor ahead stays clear, so I will continue forward without adjusting my heading.



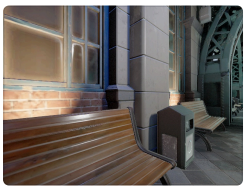
3 MoveForward
The booth now fills the left side of my view and the bench next to it sits just ahead with the litter bin at its far end. I am still a couple of steps short of standing at that bench, so I will keep moving forward.



4 MoveForward
I have drawn level with the edge of the phone booth, and the bench beside it lies just ahead and slightly left along the wall. Another forward step should bring me right up alongside its seat.



5 MoveForward
The bench now occupies most of the near ground in front of me, with the litter bin at its right end within arm's reach. One more short step should place me directly at the bench beside the phone booth.



6 Finish
I am standing right at the wooden bench beside the phone booth, its seat within arm's reach. The task is complete, so I stop here.

Figure 9: The six steps of a Basic trajectory. Each panel shows the frame before an action, the action, and its rationale.

Exploration Explore the area to find the red door beneath the D Sector sign, then walk toward it. 8 steps



1 TurnLeft
From here I only see storage racks, a yellow covered pallet and an A1 bay marker, so neither the D Sector sign nor a red door is in view. I will turn left to begin a steady scan of the surrounding bays for that signage.



2 TurnLeft
Rotating has opened up an aisle to the left where large wall lettering ending in TOR is partially visible above a workbench area, which looks like a sector sign. I will keep turning left in the same direction to bring that lettering and the wall beneath it fully into view.



3 TurnLeft
The D SECTOR sign is now readable on the far wall and a red door sits directly beneath it, but both are off to my left with racking crowding the right side of my view. I will turn left once more to place that red door near the center of my heading before walking.



4 MoveForward
The red door under the D SECTOR sign is now centered ahead of me with clear blue floor between us. I will walk forward along this open lane to close the distance to the door.



7 MoveForward
The red door and its frame now dominate my view just a short step away, with the first aid board and table clear of my line of travel. I will take one more step forward to stand directly at the door.



8 Finish
The red door fills my view at arm's length, its handle directly in front of me. The task is complete, so I stop here.

Figure 10: Six of the eight steps of an Exploration trajectory. Each panel shows the frame before an action, the action, and its rationale.

Dynamic Spatial-Semantic Navigate to the second cone from the left. 6 steps



1 TurnRight
Two orange cones stand on the deck ahead among slim striped posts; the second from the left is the one right of centre, roughly in line with the blue barrels. It lies to the right of my heading, so I turn right to face it.



2 MoveForward
The cone I picked, the one standing in line with the stack of blue barrels, now sits near the middle of my view. The concrete between us is open and clear, so I walk forward to close the distance.



3 MoveForward
The cone by the barrels has grown larger and stays centred in front of me, confirming my heading is good. A long stretch of deck still separates us, so I keep walking forward.



4 MoveForward
The cone stands squarely ahead with its stripes clearly readable, yet it is still a few paces away from me. I advance again along the same line.



5 MoveForward
The cone now fills much of the lower view but I am not yet beside it. The other cone has dropped out of my view to the left, so I simply keep following this one and step forward to cover the last stretch.



6 Finish
The cone stands directly at my feet, filling the near view. I have reached it, so there is no reason to move further.

Figure 11: The six steps of a Dynamic Spatial-Semantic trajectory. Each panel shows the frame before an action, the action, and its rationale.

Multi-stage Head toward the gold-framed painting, then proceed to the red suitcase. 11 steps



1 TurnLeft
The gold-framed painting hangs on the left wall of this corridor, well off to the side of my current heading. I will turn left so that wall and its painting come into the center of my view before I start walking. That should give me a clean line toward it.



4 MidwayTarget
The gold-framed painting now fills the left edge of my view at arm's reach, its frame and glass close enough to see the texture. I have arrived at the painting. I mark reaching it, and now I turn my attention to the red suitcase further along the corridor.



5 TurnRight
Standing this close to the painting, the wall blocks most of my view and I cannot yet see the red suitcase. I will turn right to swing my heading back into the open corridor where the luggage stands. That should start bringing the suitcase into view.



6 TurnRight
The corridor has opened up and I can now make out the red suitcase beside the brass luggage cart, but it still sits right of my center. Another turn right should line my heading up with it before I walk.



10 MoveForward
The red suitcase is now large in the lower right of my view, close to the guest room door, but I am still slightly short of standing beside it. One more step forward should bring me right up alongside it.



11 Finish
The red suitcase is directly beside me at arm's reach on the right. The task is complete, so I stop here.

Figure 12: Six of the eleven steps of a Multi-stage trajectory. Each panel shows the frame before an action, the action, and its rationale.

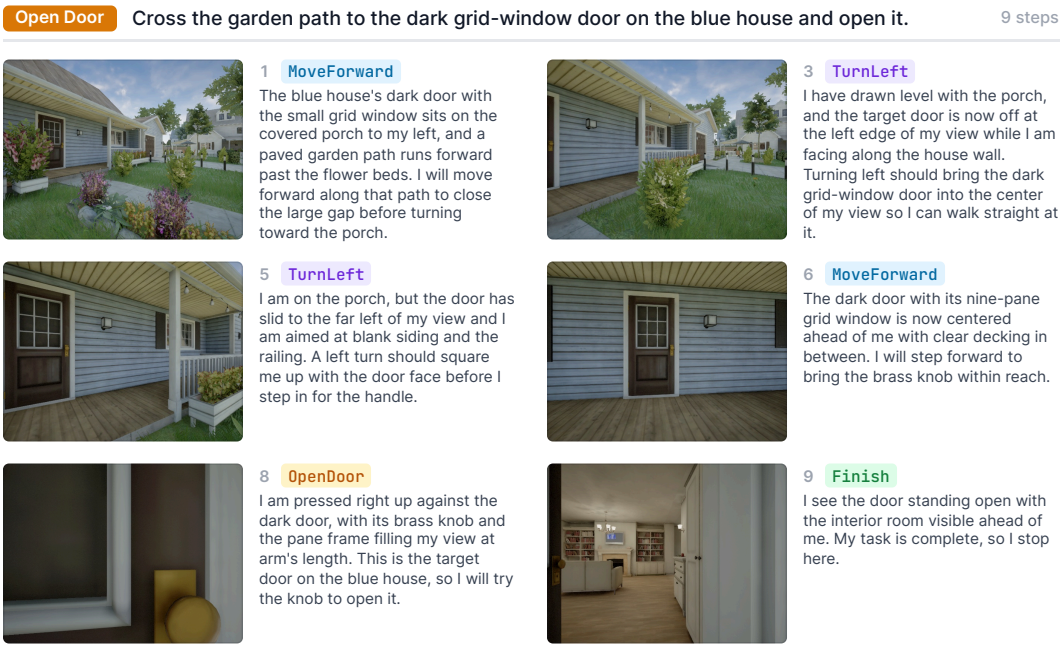


Figure 13: Six of the nine steps of an Open Door trajectory. Each panel shows the frame before an action, the action, and its rationale.



Figure 14: Six of the nine steps of a Pick & Drop trajectory. Each panel shows the frame before an action, the action, and its rationale.

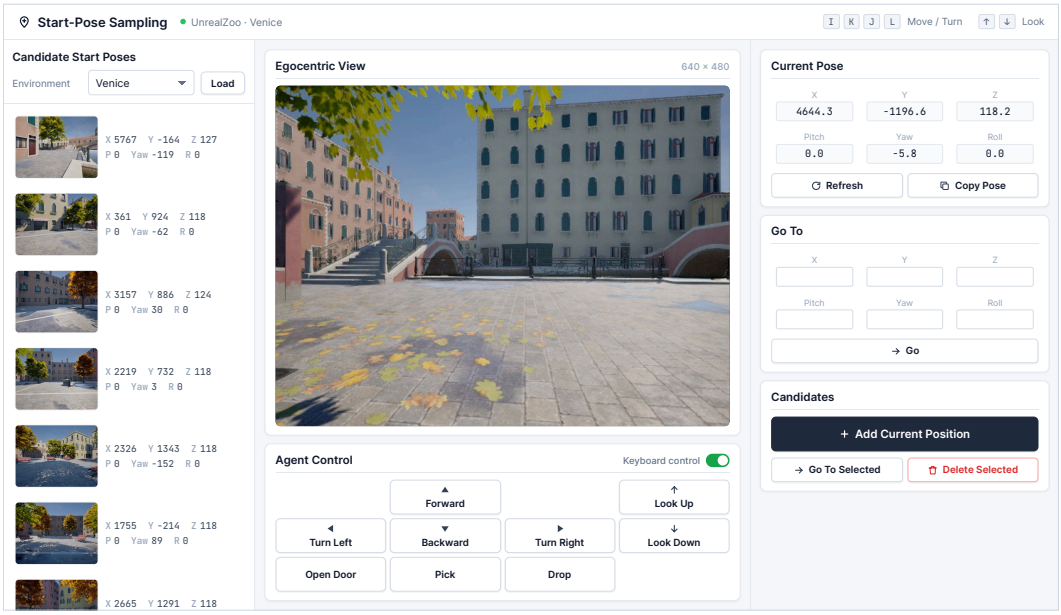


Figure 15: The start-pose sampling interface of Stage 1. The left panel lists the candidate start poses sampled in one environment. The center panel shows the egocentric view and the agent's actions. The right panel holds the current pose, the fields to move the character to a given pose, and the candidate controls.

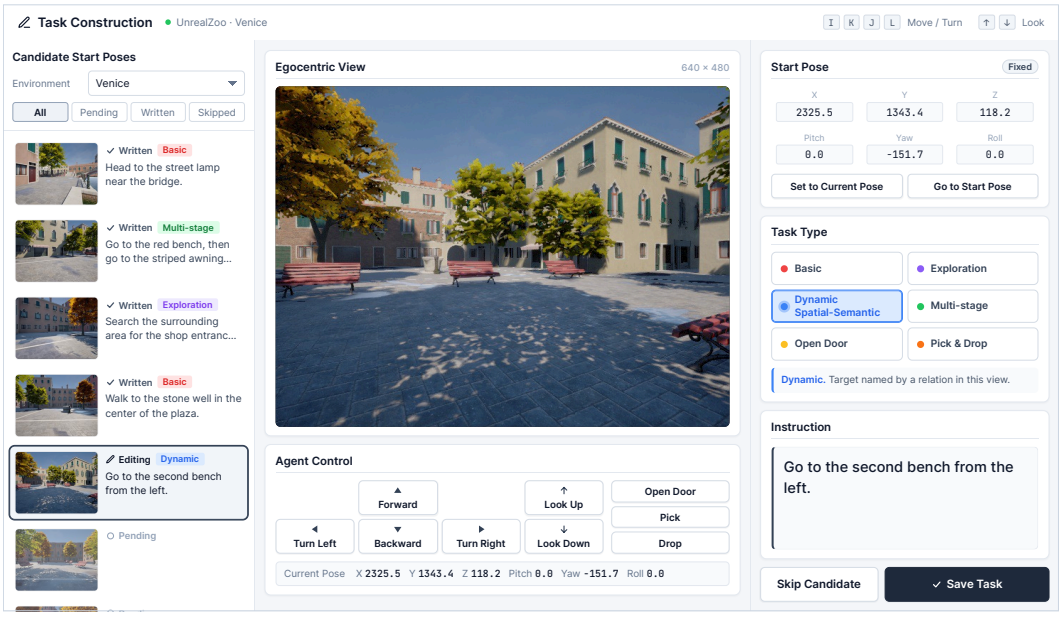


Figure 16: The task construction interface of Stage 2. The left panel lists the candidate start poses of one environment. The center panel shows the egocentric view and the agent's actions. The right panel holds the start pose, the task type, and the instruction.

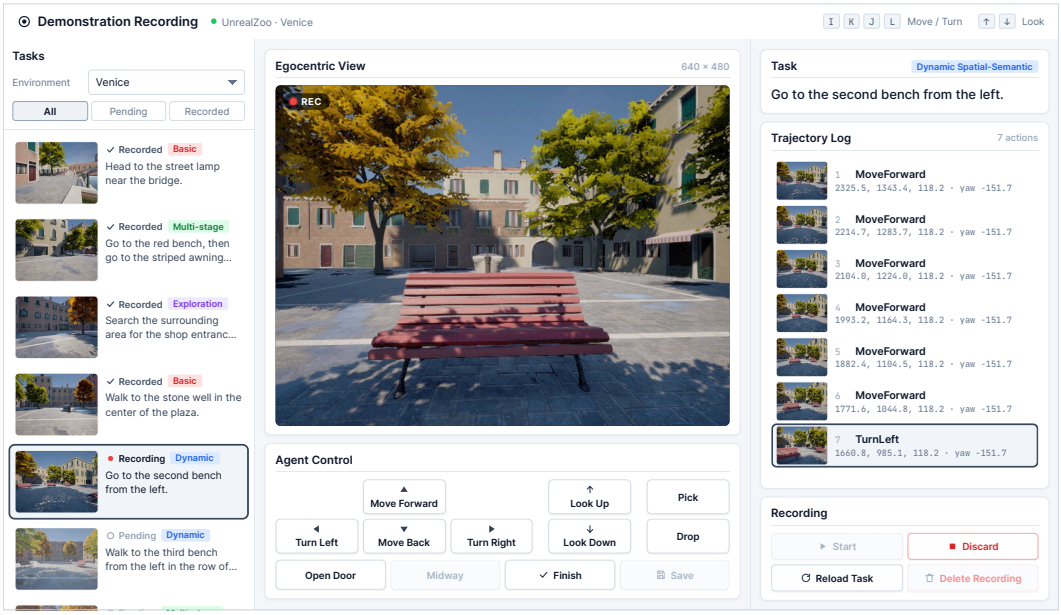


Figure 17: The demonstration recording interface of Stage 3. The left panel lists the tasks of one environment. The center panel shows the egocentric view and the agent's actions. The right panel holds the instruction, the logged steps, and the recording controls. Each logged step shows the frame before the action, the action, and the pose.

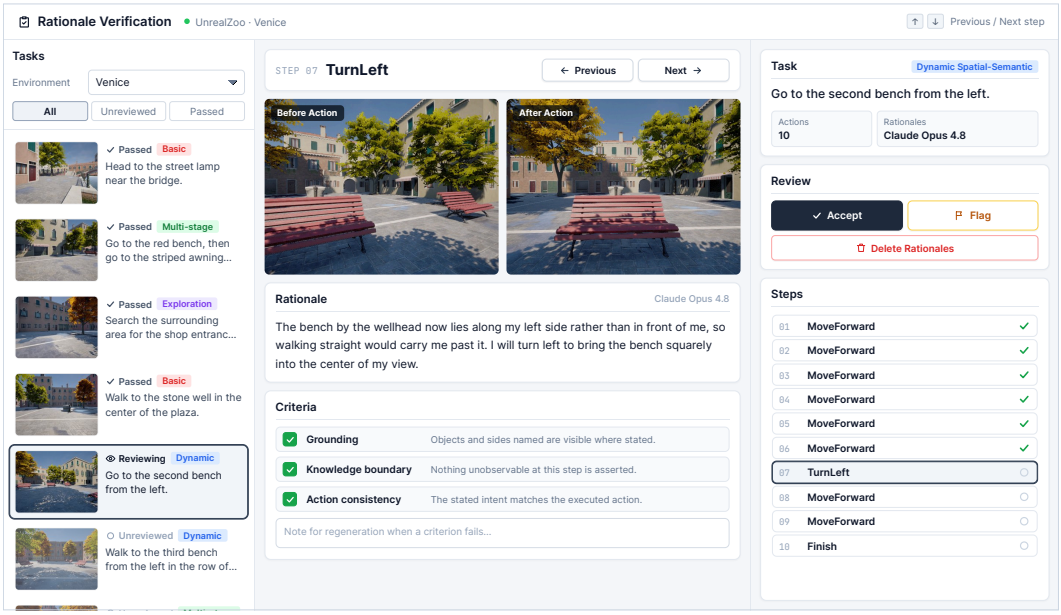


Figure 18: The rationale verification interface of Stage 4. The left panel lists the tasks of one environment. The center panel shows the step under review with the frames before and after the action, the rationale, and the three criteria. The right panel holds the instruction, the review controls, and the steps of the trajectory.

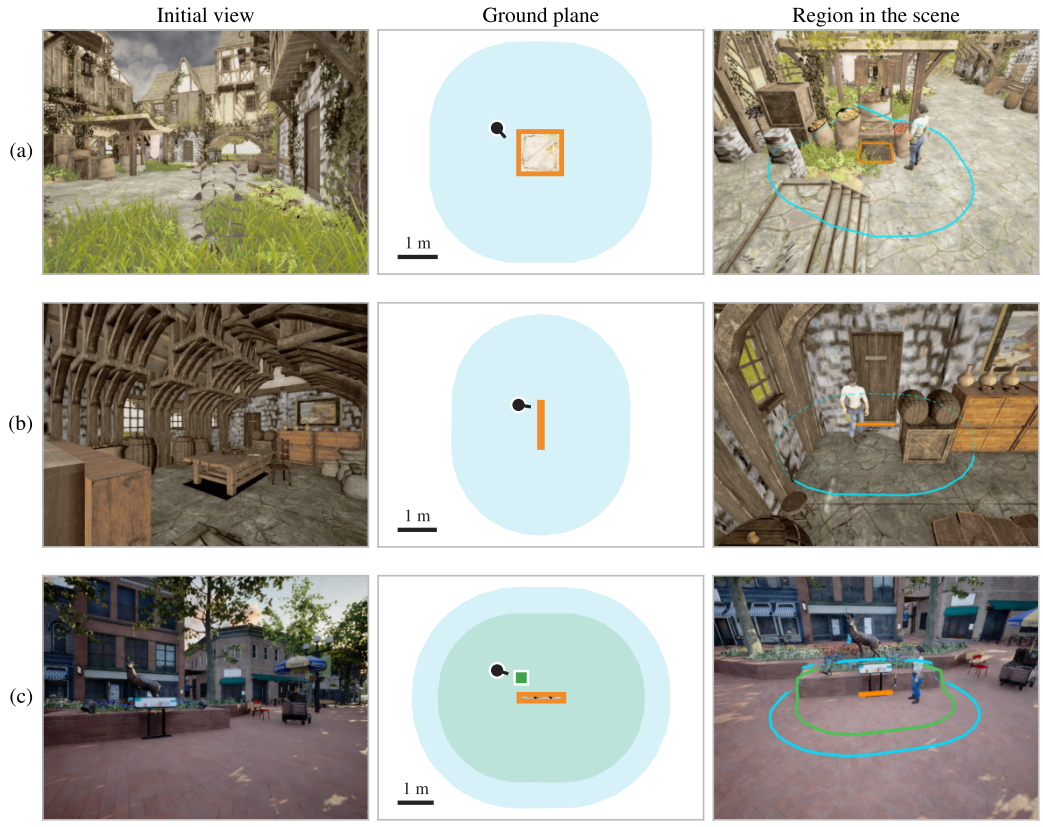


Figure 19: Success regions of three benchmark tasks: (a) a Basic task whose target is a stack of crates, (b) an Open Door task, and (c) a Pick & Drop task whose landmark is a signboard. The initial view is the frame the task is written from. In the ground plane, the reconstructed target points are drawn in color, the footprint in orange, the region of the agent in cyan, and the region of the object in green, with the demonstrator at `Finish` as a black marker and the released object as a green square. The right column is the render an annotator approved, with the same boundaries drawn over the scene.

a 10 cm grid, cells holding fewer than three points are discarded as depth noise, and the connected component containing the target's own pose is kept.

Footprint. The selected points are projected onto the ground plane, trimmed to the interval between the 0.25 and 99.75 percentiles in each axis, and enclosed in a convex hull. The published footprint is the minimum-area rectangle around that hull, which is fitted to the target rather than aligned with the world axes. Doors, windows and wall patches are handled separately. Their points are fitted with a plane, their extent along the plane is taken between the 1 and 99 percentiles, and the footprint is a rectangle 4 cm thick on that plane.

From footprint to region. The footprint is rasterized at 2 cm, a distance transform gives every cell its distance to the footprint, and the cells within 2.2 m are kept. Regions of several footprints are unioned before the raster is traced back into polygons. The radius is a body half-width of about 0.45 m plus one forward step of about 1.2 m, rounded up. The footprint itself is part of the region, and obstacles are not subtracted from it, since the simulator blocks the character at contact and contact is permitted during an episode.

Height. Each region stores the height the demonstrator was at when the corresponding action was issued, the pose at `Finish` for a final region and the pose at `MidwayTarget` for the region of a

first target. An episode satisfies the height condition when the agent is within 1 m of that value. The tolerance belongs to the metric and is applied at scoring time, so changing it does not rewrite the published geometry.

Pick & Drop. These tasks publish two regions around the landmark. The released object is required to lie within 2.2 m of the footprint, and the agent within 2.92 m. The wider allowance for the agent is the same 2.2 m plus the 0.72 m by which the `Drop` action carries the object ahead of the agent, measured in the simulator.

Checks. A reconstruction is accepted only if it is covered on a 2 cm grid to at least 95 percent, is fused from at least three viewpoints, and has most of its volume supported by two or more of them. A reconstruction that fails is rescanned automatically with a wider and denser sweep. Every region is then checked against the demonstration that produced the task, and a region that does not contain the demonstrator's pose at the corresponding action is rejected. Finally, an annotator reviewed two renders of every region, the region drawn in the ground plane and the region drawn over a top-down view of the scene, and recorded an approval bound to a hash of the geometry, so that any later change to the geometry withdraws it. All 686 benchmark tasks carry such an approval.

D.3 Evaluation Metrics

We report the five metrics of R2R and VLN-CE (Anderson et al., 2018; Krantz et al., 2020), adapted to the multi-stage and the interaction task types. Each is computed from the recorded episode and from the published region of its task. The region also records the reference length of SPL, so no demonstration is consulted at scoring time. An episode ends when the agent issues `Finish` or after 60 steps. An episode that reaches the cap has no `Finish` and counts as a failure for SR and SPL, while OS and NE remain defined.

Success Rate (SR). An episode succeeds when `Finish` is its last action and the pose recorded at that action lies in the success region of the final target, in the ground plane and in height. Multi-stage tasks require in addition a `MidwayTarget` before `Finish`, and the pose at the last such action has to lie in the region of the first target. Open Door tasks require the door to be open in the state recorded at `Finish`. Pick & Drop tasks read the recorded distance between the agent and the object, which counts as held below 30 cm. The object has to be held at some point and released before `Finish`, and at `Finish` the agent has to lie in its region and the object in its own.

Oracle Success (OS). OS applies the same conditions on the recorded state at any step of the episode and drops the two conditions on declaring, so an agent that passes through the goal state without stopping there is counted. The order of the two stages of a multi-stage task is kept. The door has to be open at the same step at which the agent is in the region, and for Pick & Drop the object has to have been held earlier and released at a step at which both are in their regions.

Navigation Error (NE). NE is the distance in the ground plane from the last recorded pose to the nearest target footprint, and it is zero when the pose is inside one. The footprint is the target itself rather than the region around it. An NE below 2.2 m therefore means the agent stands inside the success region. Success requires this and a height within 1 m and a `Finish`. For multi-stage tasks NE is measured against the final target, and for Pick & Drop against the landmark for the agent alone, since an object that was never picked up rests where it spawned and would put a fixed floor under the measure. NE is reported in meters.

Trajectory Length (TL). TL is the number of actions in the episode, at most 60.

Success weighted by Path Length (SPL). SPL weights each success by the ratio of a reference length to the length the agent walked and averages over episodes, and a failed episode contributes zero. The length walked is the sum of the displacements in the ground plane between consecutive poses. The reference is the shortest path from the start pose of the task to the success region, measured on a graph whose nodes are the poses recorded in the demonstrations collected in that environment, with edges between consecutive poses of one demonstration and between poses of different demonstrations that lie within 30 cm of each other. The path is routed through the region of

Table 10: Robustness to instruction wording. Success rate (%) over the 686 benchmark tasks on the original instructions and on three rewrites of them.

Method	Original	GPT-5.5	Gemini 3.1 Pro Preview	DeepSeek-V4-Flash
GPT-5.5 (OpenAI, 2026)	55.4	55.4 (Δ+0.0)	55.1 (Δ-0.3)	55.8 (Δ+0.4)
Claude Opus 4.8 (Anthropic, 2026)	45.5	45.2 (Δ-0.3)	44.9 (Δ-0.6)	44.5 (Δ-1.0)
Gemini 3.1 Pro Preview (Google DeepMind, 2026)	55.7	54.4 (Δ-1.3)	56.7 (Δ+1.0)	56.4 (Δ+0.7)

the first target for multi-stage tasks and through a pose within the pick range of 2.1 m of the object for Pick & Drop. When the start pose already lies in the success region the graph gives a length of zero, and the length of the demonstration of that task is used instead.

D.4 Specific Definitions of Action Quality Metrics

Three metrics record the course of an episode rather than its outcome.

Block (%↓). A move action is *obstructed* if the resulting displacement is below 20% of the nominal step length of that action. Within a run of consecutive obstructed moves, the first two are not counted, and every obstructed move from the third on is counted as blocked. Block is the fraction of move actions counted this way, averaged over the tasks of the evaluation set. A high Block rate means the agent keeps moving into something that stops it.

Revisit (%↓). A step *revisits* if the two consecutive views it produces have both been observed earlier in the same episode. Revisit is the fraction of episodes with at least one such step. A high Revisit rate means the agent returns to views it has already seen.

Recovery (%↑). An interaction *fails* if `OpenDoor` is issued while the door is closed and the door does not open, or if `Pick` is issued while nothing is held and the object is not held afterwards, where held means within 30 cm of the agent. Whether an interaction took effect is visible in the frame that follows it. Recovery is computed over the episodes with at least one failed interaction, and is the fraction of them in which the door is open or the object is held at a later step. A high Recovery rate means the agent goes on to open the door or pick up the object after an attempt that did not take effect.

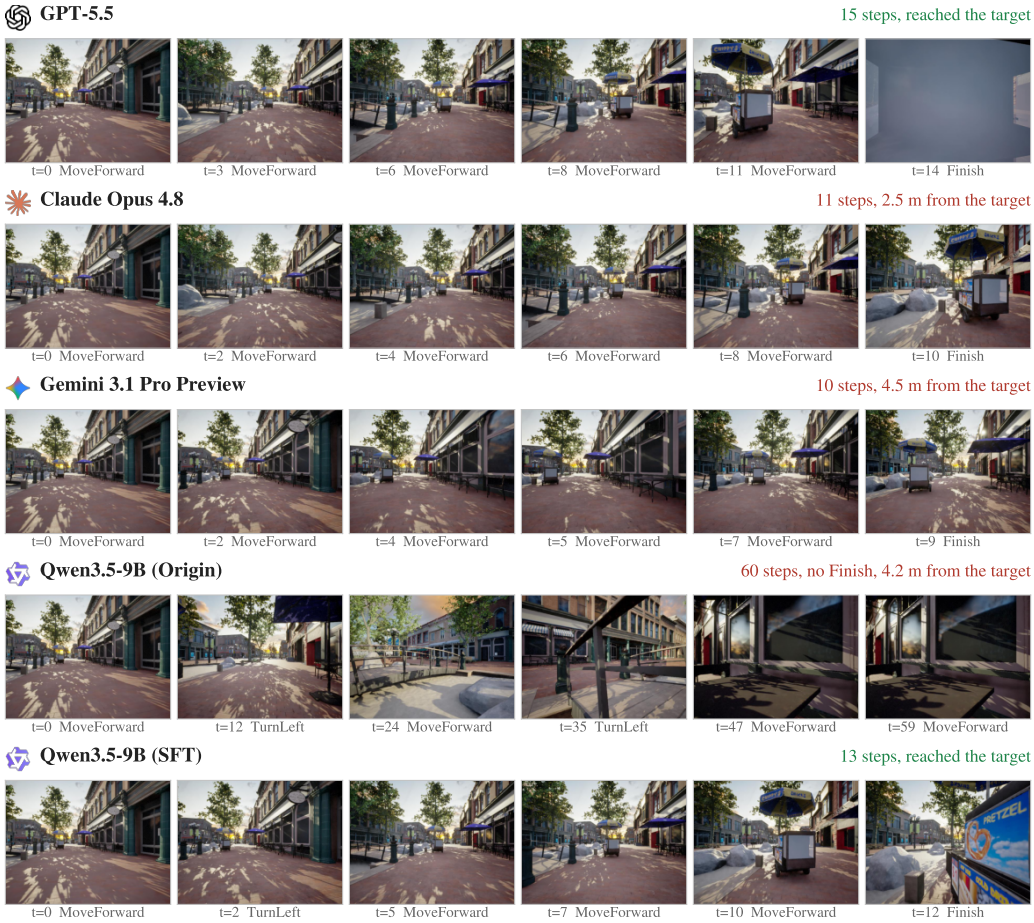
D.5 Robustness to Instruction Wording

The benchmark instructions are written by annotators and corrected for grammar and spelling by Claude Opus 4.8 (Section 3.2), which is also one of the evaluated models. A model that has passed over the instructions could be favored by their wording. To measure how much the wording matters, GPT-5.5 and Gemini 3.1 Pro Preview each rewrote the 686 benchmark instructions, and DeepSeek-V4-Flash (DeepSeek-AI, 2026), which is not one of the evaluated models, rewrote them as well. The three models were evaluated on the three rewritten benchmarks as well as on the original one. The training data is unchanged throughout. Table 10 reports the success rate under the four versions. No model changes by more than 2% under any rewrite, while the success rates of the three models span about 10%. The wording of the instructions therefore does not favor the model that proofread them.

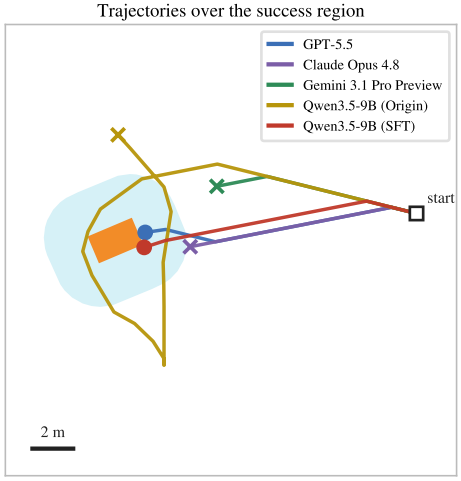
D.6 Qualitative Comparisons

Figures 20 to 25 show one benchmark task of each type, run by the three reference models and by Qwen3.5-9B before and after fine-tuning on EmbRACE. For each model, six frames of its episode are shown, the first, the last, and four evenly spaced between them, with the action chosen from each frame under it and the verdict of the scorer at the right. Below the frames, the five trajectories are drawn over the success region of the task, with the start pose as a square, a successful end as a dot, and a failed end as a cross. The rationale each model wrote at its last step is given in full. The frames, actions, poses, and rationales are those recorded during the evaluation.

Walk to the small cart with an umbrella on the walkway.



Trajectories over the success region



Rationale written at the last step

GPT-5.5, t=14 Finish

The view is filled by the cart's panel, indicating I moved into and am physically against the target. The required destination is the small cart with an umbrella, so the task is satisfied.

Claude Opus 4.8, t=10 Finish

The cart with the umbrella is now right in front of me, filling the right side of the view. I'm essentially against it. The task has one destination and I've reached it.

Gemini 3.1 Pro Preview, t=9 Finish

The view changed very little after the last step, showing that I am physically blocked by the small cart with the umbrella. I have reached the target destination. I will now finish the task.

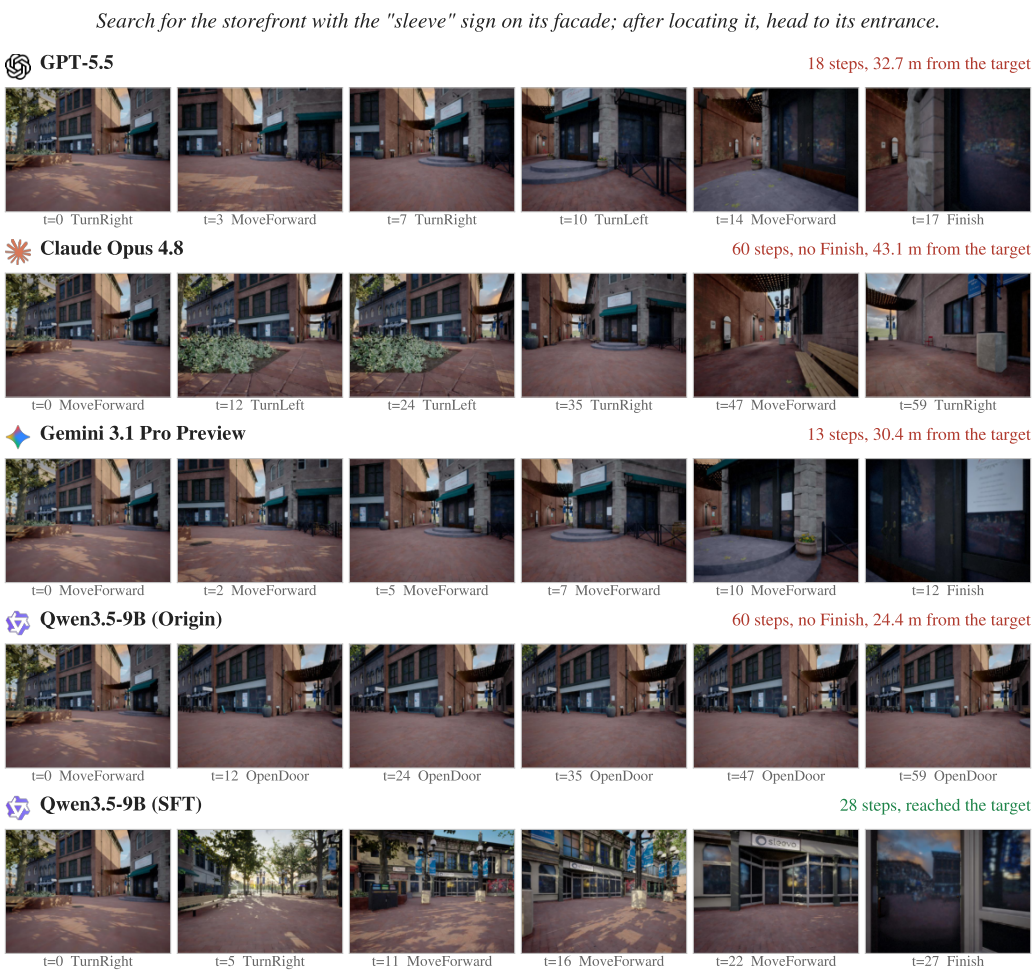
Qwen3.5-9B (Origin), t=59 MoveForward

The small cart with an umbrella is now directly in front of me, and I am very close to it. The path ahead is clear, and continuing forward will likely place me right next to the cart, fulfilling the task requirement.

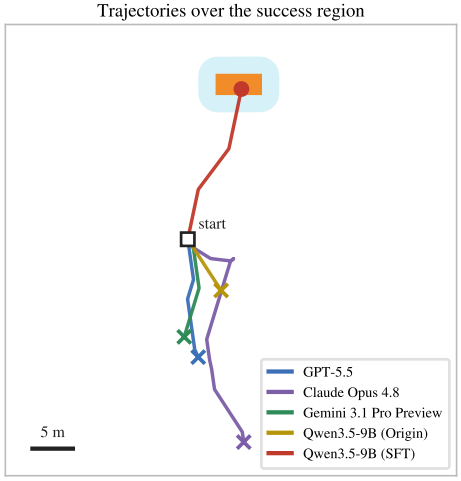
Qwen3.5-9B (SFT), t=12 Finish

I am standing immediately beside the small cart, its pretzel sign and menu panel filling my view at arm's length. The task is complete, so I stop here.

Figure 20: A Basic task in Downtown West.



Trajectories over the success region



Rationale written at the last step

GPT-5.5, t=17 Finish

I am at the storefront entrance facade, with the glass door/window and stone frame filling the view. Since I have just pushed forward into the entrance area and appear to be physically against it, the destination is reached.

Claude Opus 4.8, t=59 TurnRight

I see a building with windows and a wooden door on the right, plus a blue banner reading "DOWNTOWN WEST". This isn't the "sleeve" sign. I need to head back toward the main plaza to explore the storefronts seen initially. Let me continue turning to orient back toward the passage.

Gemini 3.1 Pro Preview, t=12 Finish

The view shows that I am right up against the doors of the storefront, with the handles clearly visible and very close. I have successfully reached the entrance as instructed. Since the required final state is satisfied, I will finish the task.

Qwen3.5-9B (Origin), t=59 OpenDoor

I am currently facing the storefront with the green awning, which appears to be the "sleeve" sign I was looking for. The entrance is directly in front of me, and I can see a door that likely leads inside. To complete the task, I should approach the door and open it.

Qwen3.5-9B (SFT), t=27 Finish

I am standing directly at the storefront's glass entrance door, with its handle within arm's reach. The task is complete, so I stop here.

Figure 21: An Exploration task in Downtown West.



Figure 22: A Dynamic Spatial-Semantic task in Courtyard.

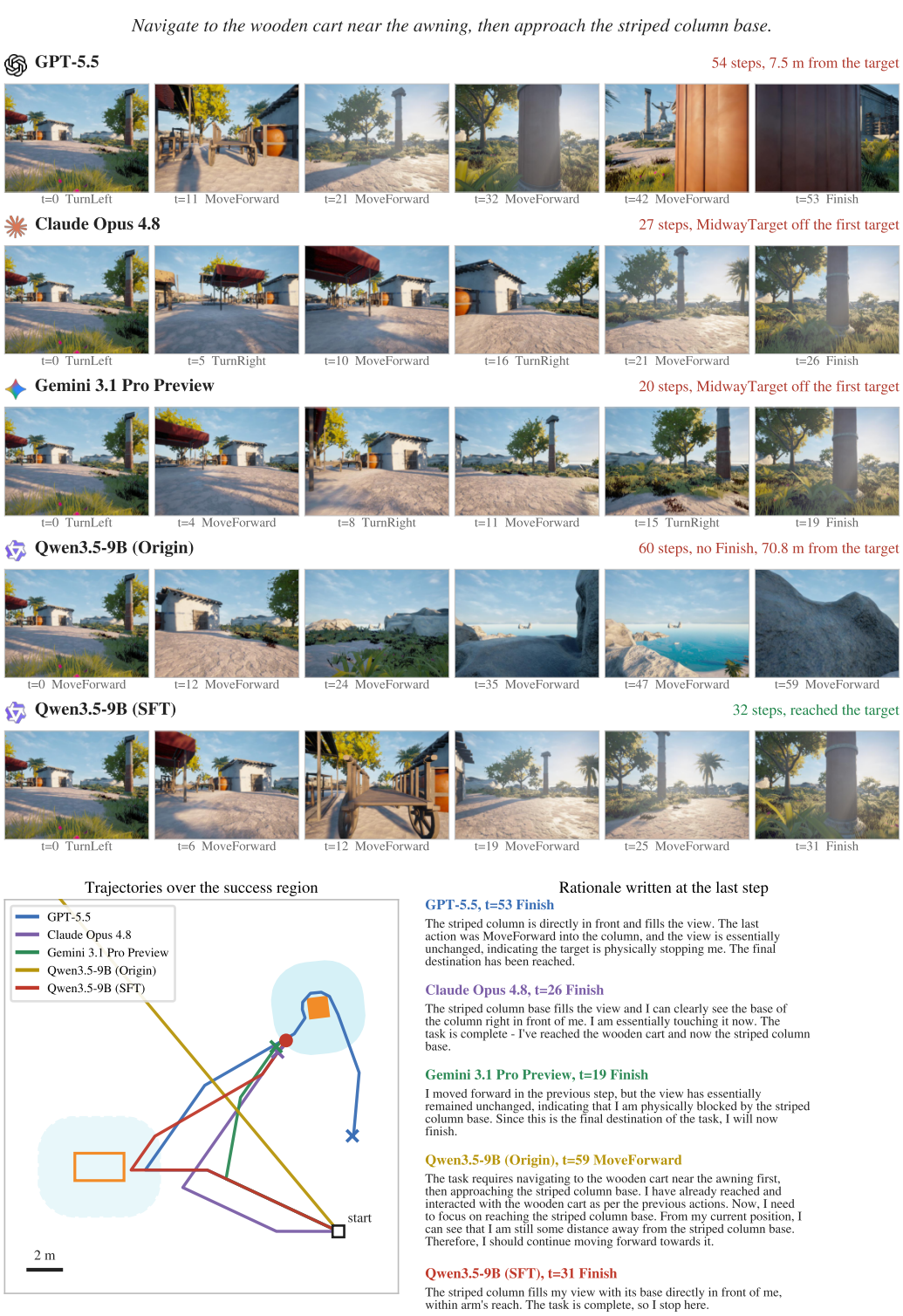
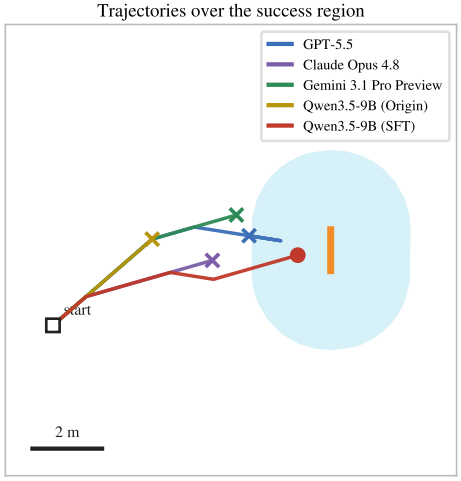


Figure 23: A Multi-stage task in Greek Island. The dashed region belongs to the first target and the solid region to the second. The trajectory of Qwen3.5-9B before fine-tuning leaves the plotted area.

Head up the wooden steps and open the large wooden door at the top of the staircase.



Trajectories over the success region



Rationale written at the last step

GPT-5.5, t=18 Finish

I am still directly in front of the large wooden door at the top of the steps, and I have already attempted to open it from close range. The task's required interaction has been performed at the correct door, so I should stop.

Claude Opus 4.8, t=6 Finish

I've climbed the steps and opened the large wooden door at the top. The task is complete.

Gemini 3.1 Pro Preview, t=9 Finish

The door has swung open to the right, revealing the space behind it. I have successfully headed up the wooden steps and opened the large wooden door as instructed. The task is complete.

Qwen3.5-9B (Origin), t=4 Finish

The task was to head up the wooden steps and open the large wooden door at the top. I have successfully ascended the stairs, reached the platform, and initiated the `OpenDoor` action. The scene remains unchanged after this attempt, suggesting the door may not be openable from this side or position. However, since I am already at the destination and have attempted the required final action, the task is considered complete.

Qwen3.5-9B (SFT), t=10 Finish

The door panel now stands swung open with the interior hall and its staircase visible beyond. The task is complete, so I stop here.

Figure 24: An Open Door task in Medieval Village (Day).

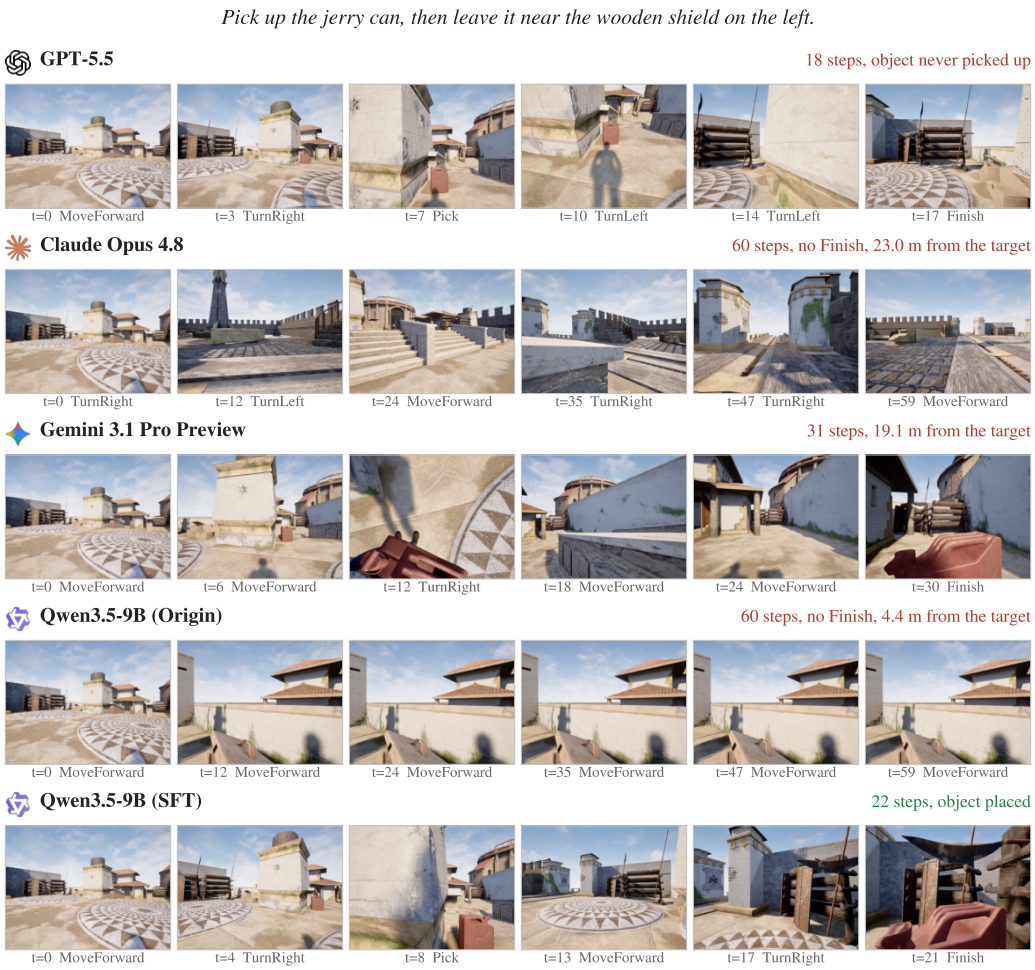


Figure 25: A Pick & Drop task in Temple Plaza. The green region is the one the object has to lie in and the cyan region the one the agent has to stand in.

E TRAINING RECIPES

E.1 SUPERVISED FINE-TUNING

Each training trajectory is unrolled into one sample per step. A sample contains the instruction, the initial frame, the frames of the preceding steps, and the actions taken so far, and its target is the rationale of that step in `<reasoning>` tags followed by the action in `<action>` tags. The observation window holds the initial frame and the thirteen frames before the current step, so a sample carries at most fourteen images.

Training uses ms-swift (Zhao et al., 2024) on eight GPUs with DeepSpeed ZeRO-3, in bfloat16 and with FlashAttention. All language-model parameters are updated and the vision encoder and the projector are frozen. The loss is taken on the last turn of each sample. The learning rate is 5×10^{-5} with a cosine schedule and no warmup, the sequence length is 8,192 tokens, and the effective batch is 128 samples, from one sample per device and sixteen gradient accumulation steps over the eight devices. The model is trained for three epochs with gradient checkpointing. The Actions only variant uses the same samples with the rationale removed from the target, and the same settings otherwise. The four base models are trained with the same recipe. The model of the last row of Table 5 is trained with the same recipe on 48K reasoning steps sampled from R2R-CoT (Wang et al., 2025), the size of the EmbRACE training split, and the steps are sampled whole trajectories at a time, so the observation history of a step is the one it was written with.

E.2 REINFORCEMENT FINE-TUNING

The GRPO stage of Section 4.6 starts from the supervised checkpoint and uses the same framework and the same sharding. A prompt is one step, drawn from trajectories not used for supervised fine-tuning, and eight completions are sampled per prompt. Two rewards are summed, one that is one when the completion has the required `<reasoning>` and `<action>` form and names an action of the action set, and one that is one when the named action is the demonstrated action. The learning rate is 1×10^{-6} and the KL coefficient is 0.04. Groups whose completions all receive the same reward are resampled up to three times. The stage runs for one epoch over its prompts.

F REAL-ROBOT EXPERIMENTS

Platform. The robot is a wheeled base with a front-facing DePtrum Aurora930 RGB-D camera and a gripper, controlled by a Raspberry Pi 5. The model sees only the color stream, at 640×400, and the depth stream is read by the grasp routine alone. At each step the current image is sent to a server that runs the model, and the returned action is executed on the robot before the next image is taken. The closed-source models are called through their APIs from that same server. The prompt, the observation history, and the action set are those of the benchmark.

Actions. MoveForward and MoveBackward move the base by 0.5 m, TurnLeft and TurnRight turn it by 30 degrees, and LookUp and LookDown tilt the camera by 30 degrees. An episode ends when the robot issues Finish or after 20 steps. Pick triggers a track-and-grab routine, which locates the object in the depth stream and closes the gripper on it, once the object is within 700 mm of the camera, and Drop moves the gripper to a fixed point about 200 mm in front of the camera and releases the object. Both interactions are threshold-based, as in the simulator, so the model decides when to issue them and the routine only executes them.

Scenes and tasks. Three scenes are used, a workspace, a home, and an outdoor area. In each scene, five tasks are set up for five of the task types of Section 3.3, Basic, Exploration, Dynamic Spatial-Semantic, Multi-stage, and Pick & Drop. Open Door is not run because the robot cannot open doors. The tasks are the same for every model, and a task is run five times per model, giving 375 episodes per model.

Success. A navigation task succeeds if the robot issues Finish within 1 m of the target, and a Multi-stage task if it does so at the second target after having reached the first. A Pick & Drop task succeeds if the object is released within 1 m of the landmark.

Failures. On the navigation types the robot loses a target that leaves the view and does not turn back to it. It also approaches another object of the same kind instead of the one named. On Pick & Drop the robot stops too far from the object for the grasp routine to reach it. It also issues Drop after a grasp that did not hold.

Examples. Figures 26 and 27 show one navigation task and one Pick & Drop task recorded in the home scene, run by the three reference models and by Qwen3.5-9B before and after fine-tuning. Each row shows the egocentric frames of one run up to $T = 16$. The label under a frame is the action chosen from it, the last frame carries no action, and the bar under a row names what the robot does over that stretch.

G LIMITATIONS

EmbRACE covers six task types across navigation and interaction. Its interaction tasks involve two forms of object manipulation, opening a door and picking an object up and releasing it. Extending EmbRACE to richer manipulation, to tool use, and to events driven by the environment is left to future work. The benchmark evaluates a single agent that decides step by step. It does not include tasks in which an agent works with a person or with another agent, which a growing share of embodied settings ask for.

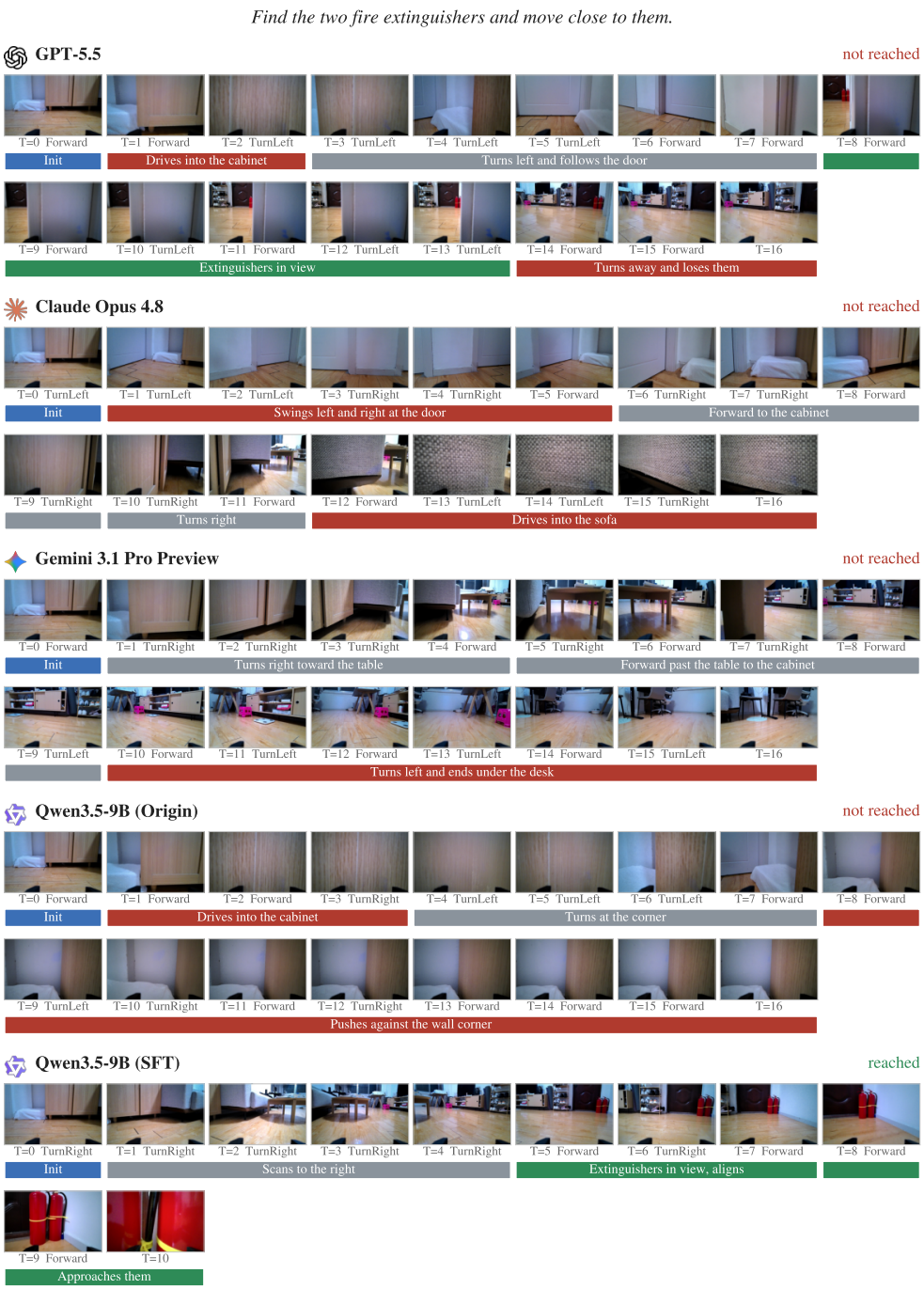


Figure 26: Real-robot navigation, *Find the two fire extinguishers and move close to them*. Each row is one run, shown up to $T = 16$, with the action chosen from each frame under it and the last frame without an action. The bars name what the robot does over each stretch, blue for the initial view, green for progress toward the target, red for a failure pattern, and gray for the rest. GPT-5.5 brings the extinguishers into view at $T = 8$ and then turns away from them. Claude Opus 4.8 swings left and right at the door and drives into the sofa. Gemini 3.1 Pro Preview passes the table and the cabinet and ends under the desk without the extinguishers entering its view. Qwen3.5-9B drives into the cabinet and then into a wall corner. After fine-tuning, the same model scans to the right until the extinguishers enter the view and drives up to them.

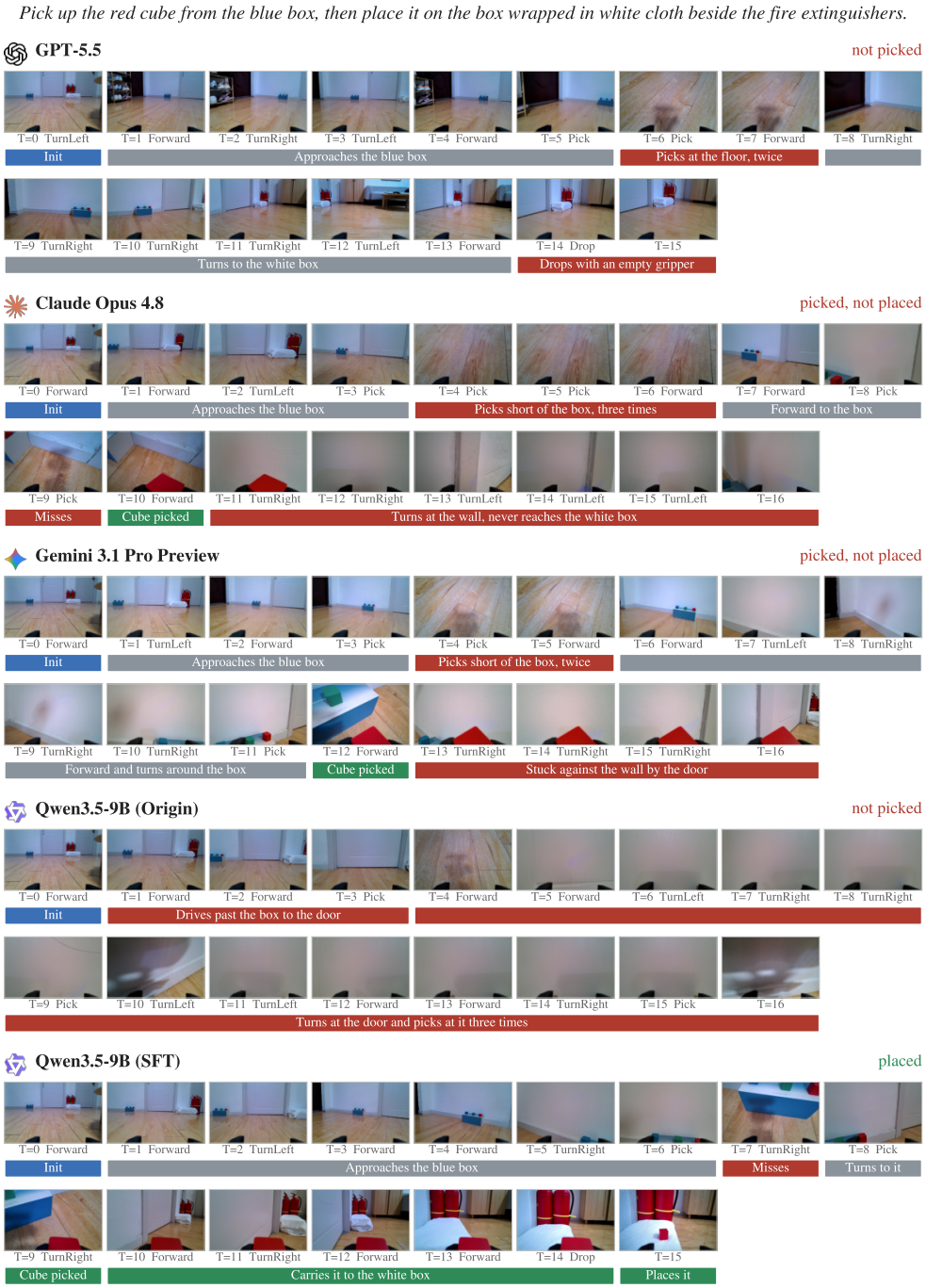


Figure 27: Real-robot Pick & Drop, *Pick up the red cube from the blue box, then place it on the box wrapped in white cloth beside the fire extinguishers*, laid out as Figure 26. GPT-5.5 issues `Pick` twice on the open floor and `Drop` at the white box with an empty gripper. Claude Opus 4.8 picks short of the box three times, picks the cube up at $T = 10$, and then turns at the wall without reaching the white box. Gemini 3.1 Pro Preview picks the cube up at $T = 12$ and ends against the wall by the door. Qwen3.5-9B drives past the box to the door and issues `Pick` against it three times. After fine-tuning, the same model misses once, picks the cube up at $T = 9$, carries it to the white box, and places it there.