

Look Before You Move: Agent-Interface-Aligned Supervision for Open-World Mobile Manipulation

Ronghe QIU Teli MA Jiaming ZHOU Zeying GONG
Yiting HE Qiang HUANG Junwei LIANG

The Hong Kong University of Science and Technology (Guangzhou)
{rqiu683,tma184,jzhou760,zgong313,yhe398}@connect.hkust-gz.edu.cn
{qiangh,junweiliang}@hkust-gz.edu.cn

Abstract: Open-world mobile manipulation requires a robot to find an object, approach it, and move it to a target receptacle from a language instruction. Training vision-language model (VLM) agents for these tasks requires demonstrations of both action selection and active vision. Existing pipelines often convert planned trajectories into high-level action labels, but shortest-path progress provides limited supervision for deciding where to look. We introduce Agent-Interface Data Alignment (AIDA), a data collection pipeline in which a teacher directly selects and executes actions through the deployed agent’s skill interface. For mobile manipulation, we design teachers that maintain target visibility, recover lost targets, and approach for picking or placing. We add template-based spatial reasoning supervision and use a shared harness to organize observations, history, and outputs during training and inference. The resulting agent, AIDA-OWMM, achieves 71.76% full-task success on 170 held-out OWMM episodes. Teacher sensitivity analysis shows that active-vision demonstrations improve student success. The simulation-trained agent achieves 38% end-to-end success on a real mobile robot without additional training. Applying AIDA to EmbodiedBench further demonstrates the pipeline’s utility for navigation tasks. [Project website](#).

Keywords: Open-world mobile manipulation, vision-language model agents, imitation learning

1 Introduction

Open-world mobile manipulation (OWMM) studies how robots can follow language instructions to find objects and move them to target receptacles in everyday environments [1, 2]. Solving these tasks requires coordinating perception, navigation, and manipulation: the robot must decide where to look, how to approach a target, and when it is ready to pick or place. In an agentic mobile manipulation system, a vision-language model (VLM) selects high-level actions from visual observations, task instructions, and interaction history, while dedicated controllers execute the corresponding motions. A central challenge is training the VLM to make effective decisions as the robot moves, its view changes, and new actions become possible.

Prior work constructs training data by converting planned or executed robot trajectories into high-level action labels [2, 3]. When these trajectories are driven primarily by shortest-path progress, they can provide limited supervision for active vision: keeping a target visible during approach, recovering it after it leaves view, and obtaining a suitable view for manipulation. This motivates collecting demonstrations from a teacher that actively decides where to look and how to move.

To this end, we introduce **Agent-Interface Data Alignment (AIDA)**, a pipeline for constructing demonstrations through the deployed agent’s decision interface, and apply it to mobile manipulation. At each step, the teacher selects a skill and its arguments, executes the call, and uses the

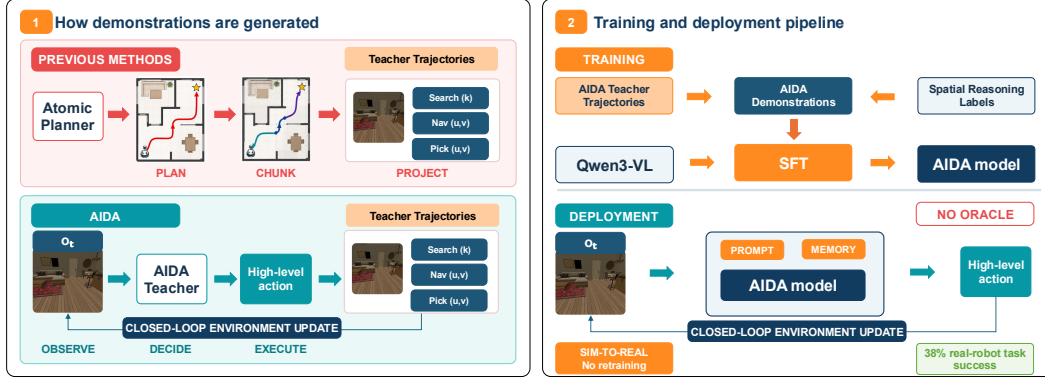


Figure 1: **AIDA-OWMM overview.** Left: the teacher generates demonstrations by selecting and executing high-level actions. Right: spatial annotations and a shared agent harness support training and deployment. The real-robot rate shown includes low-level grasp failures among recorded attempts.

resulting observation to choose the next action (Figure 1). High-level actions thus guide demonstration collection instead of being inferred afterward from motion trajectories. The teacher can use privileged simulator information, while the student’s inputs contain only observations and context available at deployment.

Within AIDA, we design the teacher to demonstrate *active target acquisition*. Finding a target is only the beginning: it may leave view during approach or remain out of reach when the robot stops. Building on active perception [4, 5], the teacher moves to collect useful views, turns to recover lost targets, and approaches to prepare for picking or placing. These behaviors teach the student to coordinate looking and moving in preparation for manipulation.

We further provide **spatial reasoning supervision** to make the judgments behind these actions explicit. For each teacher action, an offline annotator describes whether the target is visible, where it lies relative to the robot, whether the next skill can be executed, and how far the task has progressed. A shared *agent harness* assembles prompts, maintains episode memory, and defines the model’s output format consistently during training and deployment. We use supervised fine-tuning to teach the VLM to predict spatial state, reasoning, and actions from its observations and history. At deployment, the model generates these outputs itself.

Our mobile manipulation agent, **AIDA-OWMM**, achieves 71.76% full-task success on 170 held-out OWMM episodes. Teacher sensitivity analysis shows that demonstrations from teachers that maintain visibility and recover lost targets yield better student performance than those from a simple approach policy; harness ablations further support action history and spatial reasoning supervision. Trained entirely in simulation, the agent also performs pick-and-place tasks on a real mobile robot in an office environment without additional model training. Finally, we evaluate the approach on short-horizon navigation tasks in EmbodiedBench to examine its applicability beyond the OWMM benchmark.

Our contributions are:

- **A pipeline for agent-aligned data collection:** AIDA generates teacher demonstrations directly through the deployed agent’s high-level action interface.
- **A teacher for active-vision demonstrations:** we design behaviors for maintaining visibility, recovering targets, and approaching for manipulation, and evaluate their effect on student performance.
- **A harness for spatial reasoning supervision:** structured target state and task progress guide action learning through a shared prompting and memory interface.

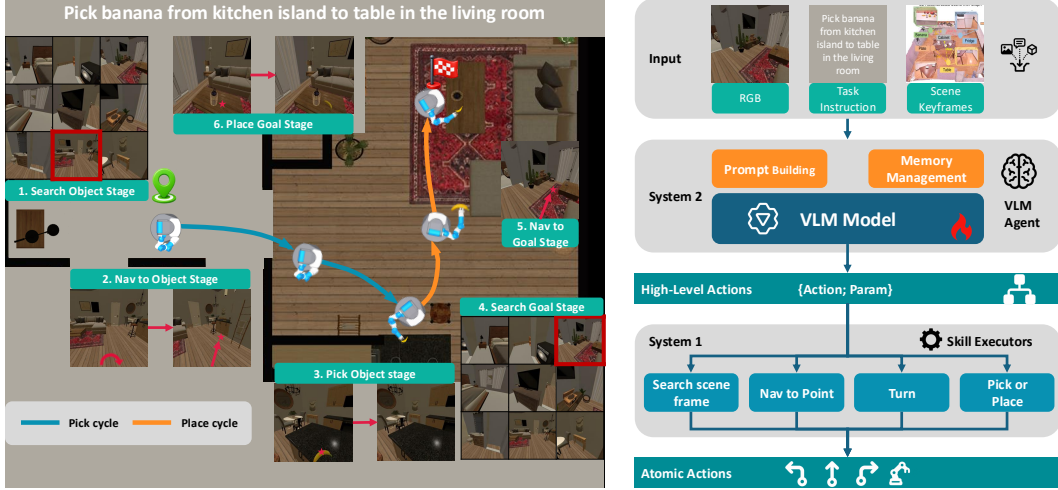


Figure 2: **OWMM agent interface.** The VLM selects search, navigation, turn, pick, and place skills, which fixed controllers execute as robot motions. Scene key frames are provided during search.

2 Related Work

VLM agents for mobile manipulation. Mobile manipulation systems combine semantic task reasoning with physically grounded navigation and manipulation. Prior work grounds language plans in skill affordances and learns language-conditioned robot policies from large demonstration corpora [6, 7, 8, 9], with recent extensions to cross-domain generalization and unseen manipulation tasks [10, 11, 12]. Open-vocabulary mobile-manipulation systems combine language, perception, maps, and robot APIs to complete household tasks [1, 13, 14, 15, 2], while whole-body work highlights stage-dependent perception demands under changing viewpoints [16]. These systems establish the usefulness of modular high-level skills and open-vocabulary perception. Our focus is different: we study what form of synthetic supervision teaches the VLM controller to use those skills as a closed-loop active target-acquisition policy.

Embodied VLM navigation and view control. VLM navigation methods train agents to follow instructions, use visual history, and select navigation actions from partial egocentric observations. Recent systems study visual history, map use, and action prediction for embodied VLM navigation [17, 18, 19, 20, 21, 22, 23]. Object-goal navigation and active VLN further use spatial abstraction, exploration, and viewpoint correction to search for open-vocabulary targets or reduce route-following uncertainty [24, 25, 26]. These works show that VLMs can learn route following, memory use, and action prediction, but OWMM target acquisition has a different endpoint: the policy must stop in a robot-object relation from which manipulation can succeed. AIDA-OWMM therefore trains active view control around task entities and manipulation readiness rather than route progress alone.

Active perception and next-best-view methods choose viewpoints or paths to reveal occluded structure, reduce grasp uncertainty, or improve reachability [27, 4, 28, 29, 5]. Here, view control is distilled into a VLM high-level policy that decides when to search, turn, approach, pick, and place from the deployed agent interface.

Spatial reasoning as policy state. Spatial-VLM work studies depth, metric relations, object grounding, cross-view correspondence, and persistent spatial memory [30, 31, 32, 33, 34, 35, 36]. Other work turns visual-spatial understanding into points, visual prompts, or multi-turn state-and-transition reasoning for robot actions [37, 38, 39]. AIDA-OWMM uses spatial reasoning in a narrower way: object-centric state is an intermediate policy variable for deciding whether to search, turn, approach, pick, or place. The novelty is using spatial state inside active target-acquisition demonstrations generated in the deployed OWMM agent interface.

3 Method

AIDA-OWMM trains a VLM to select high-level skills for mobile manipulation. The pipeline has three components: a teacher collects demonstrations through the agent’s action interface, an annotator adds spatial reasoning targets, and a shared harness organizes prompts and episode memory for training and deployment (Figure 3).

3.1 Problem Formulation

An OWMM episode is specified by an instruction q , such as moving an object from a source receptacle to a goal receptacle. Following the OWMM benchmark [2], we assume a pre-mapped environment and known robot localization. The map provides scene geometry for navigation; the agent identifies the object and goal receptacle through RGB key frames with known camera poses and its own visual exploration.

At decision step t , the VLM receives $o_t = (I_t^c, q, \mathcal{A}, H_{t-1}, R_t)$, where I_t^c is the current head-camera RGB image, $\mathcal{A}$ is the skill vocabulary, and H_{t-1} contains prior skill calls, their arguments, and step summaries. R_t contains scene key frames during source-object or goal-receptacle search and is omitted during local navigation and manipulation. The VLM outputs $y_t = (r_t, a_t, u_t, m_t)$, where r_t is explicit reasoning about spatial relations and task progress before selecting skill a_t and argument u_t , and m_t summarizes the step for subsequent decisions.

Teacher and student share five skills. A fixed executor $\mathcal{E}$ converts each call into low-level robot actions, $\mathbf{a}_t^{\text{low}} = \mathcal{E}(a_t, u_t, I_t^c, D_t^c)$, using the geometric map, robot localization, and stored key-frame poses as context. Here, D_t^c is the depth image used to ground pixel coordinates in 3D. The skills are:

- `search_scene_frame` selects a key frame by its displayed ID; the executor uses its associated pose as the navigation destination.
- `turn` specifies an angle for the controller to rotate the robot in place.
- `nav_to_point` specifies a normalized pixel coordinate; the executor back-projects it using depth and camera pose to obtain a navigation destination.
- `pick` specifies a normalized pixel coordinate on the object; the manipulation controller moves the end effector toward the corresponding 3D location and grasps the object.
- `place` specifies a normalized pixel coordinate on the destination; the controller moves the held object toward the corresponding 3D location and releases it.

In simulation, grasping is simplified: the object is attached to the gripper when the end effector is within the configured grasp distance.

3.2 Agent-Interface-Aligned Supervision

AIDA collects demonstrations by executing teacher decisions through the student’s skill interface. During collection, the teacher π_T uses privileged simulator state z_t to select a skill and its argument, $(a_t^*, u_t^*) = \pi_T(I_t^c, q, H_{t-1}^T, R_t, z_t, \phi_t)$, where H_{t-1}^T is the observation and action history available during collection and ϕ_t denotes the current teacher phase. The selected call is executed, and the resulting observation is used for the next decision. Each recorded action therefore comes from a high-level decision made during the rollout.

3.3 VLM-Native Target-Acquisition Teacher

The teacher uses target visibility, direction, and proximity to choose actions through the VLM’s skill interface. It follows the task from source-object search and approach to picking, then goal-receptacle search and approach to placing. Simulator masks, poses, projected paths, and bearings support these decisions during data collection.

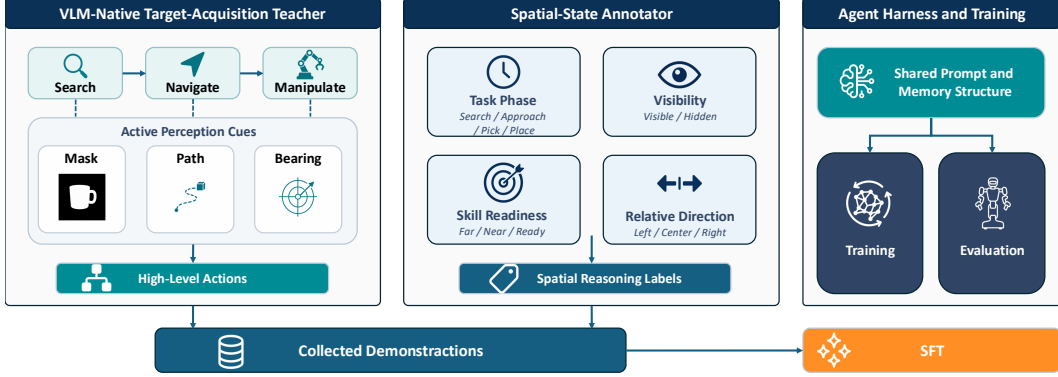


Figure 3: **AIDA-OWMM pipeline.** The teacher collects high-level skill demonstrations, the annotator provides spatial reasoning targets, and the harness constructs training examples and maintains the deployed agent’s context.

During search, the teacher selects the scene key frame whose viewpoint is closest to the target and issues `search_scene_frame`; the search phase completes after this call. During navigation, the teacher actively searches for the target and relevant visual cues, choosing between `turn` to adjust the view and `nav_to_point` to approach the target. When the target is visible or recently visible, navigation points favor keeping it in view while moving within reach. When the target has not been seen recently, projected paths guide the selection of navigable points. For object pickup during collection, the target must be within 1.0 m with a visible mask exceeding 100 pixels; the teacher uses the 15° bearing threshold to align its view. At deployment, the VLM decides when to pick from its observation.

For `pick` and `place`, the teacher obtains the object or goal-receptacle mask and uses its centroid, expressed as a normalized pixel coordinate, as the action argument.

A teacher may fail to complete an episode, for example by losing track of the target or exceeding the step budget. Such unsuccessful trajectories are excluded from the training dataset. Teacher thresholds, filtering rules, and recorded fields are detailed in Appendix A.

3.4 Spatial-State Annotator

After collection, each teacher trajectory provides a sequence of observations and high-level actions. The spatial-state annotator supplements each step with the reasoning that the student is trained to produce before selecting that action. It generates this reasoning offline using deterministic templates and the simulator information recorded during collection.

The templates describe target visibility, relative direction, readiness for the next skill, and task progress, then connect these judgments to the teacher’s action. Visibility and direction are derived from the target mask; when the target is outside the image, its direction is obtained from the recorded robot pose and target position. Readiness describes whether the robot should adjust its view, approach, pick, or place, while progress identifies the current part of the task. For example, a template may state that the target is visible on the left but too far away to pick, motivating further approach.

These descriptions form the pre-action reasoning target r_t , complementing the recorded skill a_t and argument u_t . The resulting supervision teaches the student to reason explicitly about spatial relations and task progress before acting. Label sources and template details are provided in Appendix A.

3.5 Shared Agent Harness and Training

The shared harness uses the same prompt format, history representation, and output schema for training and inference. Each prompt x_t contains the instruction, current RGB image, skill definitions,

history H_{t-1} , and scene key frames during search. Privileged simulator information is excluded from the prompt in both training and inference.

OWMM-style history recursively carries forward the latest summary [2]. Our history retains previous skill calls, their arguments, and step summaries (a_i, u_i, m_i) , keeping an explicit record of the agent’s actions.

The prompt specifies the output format for $y_t = (r_t, a_t, u_t, m_t)$, including the available skills and their argument formats. We fine-tune the VLM on annotated prompt–output pairs using $\mathcal{L}_{\text{SFT}}(\theta) = -\sum_t \log p_\theta(y_t | x_t)$. At inference, the harness executes the predicted skill call and updates history and observations for the next decision.

4 Experiments

Our experiments examine whether AIDA improves mobile manipulation, how teacher behavior and agent supervision contribute to performance, and whether the learned policy transfers to a real robot. We then use EmbodiedBench to test whether the AIDA pipeline also applies to navigation tasks.

4.1 Experimental Setup

All policies trained in our experiments use Qwen3-VL-Instruct 8B [40] as the base model and are fine-tuned with LoRA using LLaMA-Factory [41]. The OWMM comparisons vary the demonstration policy, skill set, or harness while keeping the base model, fine-tuning method, and low-level executors fixed.

The teacher sensitivity study uses four skills, excluding turn to match the OWMM action interface. The main AIDA-OWMM model and harness ablations use all five skills: adding turn lets the agent orient toward targets without translating, expanding the viewpoints it can reach. The three AIDA teachers in the sensitivity study each provide 28,480 training examples. The main model and all harness variants use the same larger AIDA dataset of 67,183 examples. Unless varied in the harness ablation, agents use the shared harness described in Section 3.5.

Following the OWMM evaluation protocol [2], we evaluate on 170 held-out episodes in pre-mapped simulated homes. Table 1 reports the fraction of episodes that retrieve a relevant object image (Image Retrieval, Object), pick up the object (Object Picked), retrieve a relevant receptacle image (Image Retrieval, Goal), and approach the receptacle (Robot Close to Goal). Full Task measures successful completion of the object transfer, including placement. Training and evaluation settings are listed in Appendix D.

4.2 OWMM Benchmark

We first compare AIDA-OWMM with existing OWMM systems (Table 1). Results for OWMM-VLM-38B, OWMM-VLM-8B, GPT-4o+PIVOT, and GPT-4o+RoboPoint are taken from OWMM-Agent [2, 42, 38, 37]. The following studies examine teacher behavior and harness design under the shared training setup.

Table 1: OWMM system comparison. Baseline results are reported by OWMM-Agent; AIDA-OWMM is evaluated on 170 held-out episodes. Intermediate columns report stage completion, and Full Task reports complete object transfer.

Method	Base model	Image Retrieval (Object)	Object Picked	Image Retrieval (Goal)	Robot Close to Goal	Full Task
OWMM-VLM-38B	InternVL-38B	88.56%	38.56%	30.39%	23.53%	21.90%
OWMM-VLM-8B	InternVL-8B	81.43%	17.92%	15.96%	10.42%	9.45%
GPT-4o + PIVOT	GPT-4o	59.15%	0.65%	0.33%	0.00%	0.33%
GPT-4o + RoboPoint	GPT-4o	56.86%	1.31%	0.00%	0.00%	0.33%
AIDA-OWMM (ours)	Qwen3-VL-Instruct-8B	90.59%	82.35%	74.12%	72.35%	71.76%

Table 2: **Teacher sensitivity.** All students use four skills without turn. Sample counts denote training examples.

Demonstrations	Samples	Success
OWMM, relabeled	152,478	38.8%
AIDA: distance-greedy	28,480	46.3%
AIDA: visibility-aware	28,480	65.5%
AIDA: target recovery	28,480	69.6%

AIDA-OWMM achieves 71.76% full-task success, compared with the reported 21.90% for OWMM-VLM-38B and 9.45% for OWMM-VLM-8B. Object-image retrieval is similar for AIDA-OWMM and OWMM-VLM-38B (90.59% versus 88.56%), while object pickup reaches 82.35% versus 38.56%. The larger difference in pickup shows that the overall improvement extends beyond finding a relevant scene image to approaching and manipulating the object.

4.3 Agent Sensitivity to Teacher Behavior

AIDA is a data collection pipeline that accommodates teacher policies designed for different embodied tasks. Even within one benchmark, it can collect demonstrations from teachers with different decision rules. Here, we test whether different AIDA teachers improve on trajectory-derived supervision and whether adding active vision to the teacher improves student success.

Table 2 compares four demonstration sources. The OWMM baseline represents trajectory-based data construction: simulator trajectories are divided into chunks, and trajectory points are projected into image coordinates to form high-level action labels. We apply the AIDA annotation and harness to these examples. The other three rows use teachers that directly generate and execute high-level actions through the AIDA pipeline. They share search and pick/place routines but differ in their navigation policy. The *distance-greedy* teacher selects navigable points that reduce distance to the object or receptacle. The *visibility-aware* teacher also selects approach points that keep the target visible after motion. The *target-recovery* teacher includes behavior for finding a previously visible target after it leaves view.

All four students use the same base model, LoRA training method, four-skill interface, spatial annotation, and harness. The OWMM baseline uses 152,478 training examples; each AIDA teacher provides 28,480 examples. We train the three AIDA students with the same seed and report mean success over three evaluations of the 170 episodes.

All three AIDA teachers outperform the trajectory-derived baseline: student success rises from 38.8% for OWMM to 46.3% for the distance-greedy teacher, 65.5% for the visibility-aware teacher, and 69.6% for the target-recovery teacher. These results show that AIDA supports effective supervision from multiple teacher policies. Among the equally sized AIDA datasets, demonstrating active visual behavior further improves task completion.

4.4 Harness Ablations

We next examine how the harness presents the demonstrations to the agent. Table 3 holds the AIDA trajectories fixed and varies scene key-frame context, action history, and reasoning supervision. Each row uses its specified format for both training and inference.

Providing key frames only during search improves success from 55.88% to 60.00% in the configuration without history or reasoning. Adding action history raises success to 65.88%, and adding spatial reasoning and step summaries reaches 71.76%. The results favor providing scene key frames during search and retaining past actions and summaries to support subsequent decisions.

Table 3: **Harness ablations.** All rows use the same AIDA demonstrations and five skills. Reasoning denotes spatial reasoning supervision.

Key frames	History	Reasoning	Success
Every stage	None	No	55.88%
Search only	None	No	60.00%
Search only	Actions	No	65.88%
Search only	Actions	Yes	71.76%

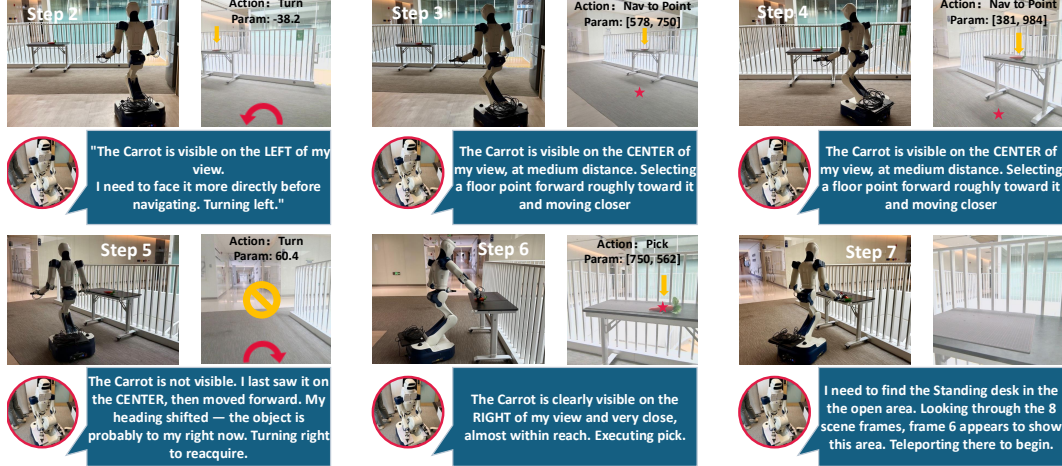


Figure 4: **Real-robot rollout.** The simulation-trained agent turns to locate the object, approaches using pixel-coordinate goals, picks it, and searches for the destination receptacle.

4.5 Real-Robot Evaluation

To test transfer from simulation to physical mobile manipulation, we deploy AIDA-OWMM on an Atribot S1 without further model training. The office environment contains two office rooms and an open tea room, with five movable objects and five receptacles. The agent uses RGB observations and scene key frames through the same high-level skill interface; the robot’s controllers execute navigation and manipulation. Hardware and execution details are given in Appendix B.

The robot achieves 38% end-to-end success with a 15-step horizon, including low-level grasp failures in the evaluation. This result demonstrates transfer from simulation to physical mobile manipulation without additional model training.

Figure 4 shows the resulting sequence of viewpoint changes, navigation, and manipulation. Failure analysis is provided in Appendix C.

4.6 EmbodiedBench Navigation

To test whether the AIDA pipeline extends beyond mobile manipulation, we apply it to EmbodiedBench [43], a language-conditioned, single-room navigation benchmark in AI2-THOR. We train a policy on 1,200 navigation episodes constructed with the AIDA pipeline. We also include the OWMM-trained policy as a zero-shot reference. Evaluation uses 60 episodes from each of the base and common_sense splits, with results averaged over two runs. Hosted-model baselines use the same prompt format, action parser, skill executor, and 10-step horizon. Model-version and protocol details are provided in Appendix B.

The EmbodiedBench-trained policy reaches 0.84 average success, while the OWMM-trained policy reaches 0.55 without additional training (Table 4). The zero-shot Qwen3-VL-8B base model reaches 0.02 under the same evaluation interface. The performance of the EmbodiedBench-trained policy supports applying the AIDA pipeline to navigation beyond the mobile manipulation setting.

Table 4: **EmbodiedBench navigation.** The EB-trained policy tests the AIDA pipeline on navigation; the OWMM-trained policy is a zero-shot reference.

Model	Base	Common	Avg.
<i>Zero-shot VLMs</i>			
GPT-5.5	0.78	0.77	0.78
GPT-4o	0.43	0.37	0.40
OpenAI o3	0.53	0.50	0.52
Gemini 3 Flash	0.08	0.15	0.12
Claude Sonnet 4.6	0.27	0.33	0.30
Qwen3-Max	0.07	0.10	0.08
Qwen3-VL-Instruct-8B	0.02	0.02	0.02
<i>AIDA policies</i>			
AIDA (EB-trained)	0.85	0.83	0.84
AIDA (OWMM-trained)	0.54	0.55	0.55

5 Conclusion

We presented AIDA, a pipeline that collects demonstrations through the high-level skill interface used by the deployed agent. For mobile manipulation, we combine active-vision teachers with spatial reasoning supervision and a shared training and inference harness. The pipeline supports different teacher policies, with visibility-aware approach and target recovery improving student success. Real-robot deployment without additional training and the application to EmbodiedBench navigation extend the evidence beyond simulated mobile manipulation. Together, these results highlight the value of designing demonstrations around the decisions an embodied agent must make, including where to look and how to move before manipulation.

6 Limitations

AIDA-OWMM uses an offline teacher with privileged simulator state, so generalization is bounded by the teacher data diversity across scenes, objects, embodiments, sensors, and skills. The OWMM simulator abstracts contact-rich grasping, slip, force control, and recovery; real-robot results show that low-level pick reliability remains a bottleneck. For navigation, our evaluation is limited to relatively short EmbodiedBench episodes. Whether agent-interface-aligned supervision provides similar benefits for longer instruction-following and search-based navigation settings, such as VLN-CE [44] and ObjectNav [45], remains an important direction for future work.

Acknowledgments

We thank the reviewers and collaborators for helpful feedback.

References

- [1] S. Yenamandra, A. Ramachandran, K. Yadav, A. S. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. M. Turner, Z. Kira, M. Savva, A. X. Chang, D. S. Chaplot, D. Batra, R. Mottaghi, Y. Bisk, and C. Paxton. HomeRobot: Open-vocabulary mobile manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1975–2011. PMLR, 2023. URL <https://proceedings.mlr.press/v229/yenamandra23a.html>.
- [2] J. Chen, H. Liang, L. Du, W. Wang, M. Hu, Y. Mu, W. Wang, J. Dai, P. Luo, W. Shao, and L. Shao. OWMMAgent: Open world mobile manipulation with multi-modal agentic data synthesis. In *Advances in Neural Information Processing Systems*, 2025. URL <https://mlanthology.org/neurips/2025/chen2025neurips-owmmagent/>.
- [3] Intern Robotics. InternVLA-N1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, Shanghai AI Laboratory, Sept. 2025. URL https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf.
- [4] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018. doi:10.1007/s10514-017-9615-3.
- [5] S. Jauhri, S. Lueth, and G. Chalvatzaki. Active-perceptive motion generation for mobile manipulation. *arXiv preprint arXiv:2310.00433*, 2023.
- [6] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, B. Ibarz, A. Ichter, E. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, C. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 287–318. PMLR, 2023. URL <https://proceedings.mlr.press/v229/ahn23a.html>.
- [7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*. PMLR, 2023. URL <https://proceedings.mlr.press/v229/zitkovich23a.html>.
- [8] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.
- [9] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, P. Sanketi, Q. Vuong, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [10] J. Zhou, T. Ma, K.-Y. Lin, Z. Wang, R. Qiu, and J. Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22551–22561, 2025.

- [11] T. Ma, J. Zheng, Z. Wang, Z. Gao, J. Zhou, and J. Liang. GLOVER++: Unleashing the potential of affordance learning from human behaviors for robotic manipulation. In *Proceedings of the 9th Conference on Robot Learning*, 2025. arXiv:2505.11865.
- [12] J. Zhou, K. Ye, J. Liu, T. Ma, Z. Wang, R. Qiu, K.-Y. Lin, Z. Zhao, and J. Liang. Exploring the limits of vision-language-action manipulations in cross-task generalization. *arXiv preprint arXiv:2505.15660*, 2025.
- [13] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto. OK-Robot: What really matters in integrating open-knowledge models for robotics. *Robotics: Science and Systems*, 2024. Presented at RSS 2024, Delft, Netherlands.
- [14] P. Zhi, Z. Zhang, Y. Zhao, M. Han, Z. Zhang, Z. Li, Z. Jiao, B. Jia, and S. Huang. Closed-loop open-vocabulary mobile manipulation with GPT-4V. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. arXiv:2404.10220.
- [15] S. Tan, D. Zhou, X. Shao, J. Wang, and G. Sun. Language-conditioned open-vocabulary mobile manipulation with pretrained models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 8778–8786, 2025. URL <https://www.ijcai.org/proceedings/2025/0976.pdf>. arXiv:2507.17379.
- [16] J. Liu, W. Cui, H. Li, and D. Zhao. InCoM: Intent-driven perception and structured coordination for whole-body mobile manipulation. *arXiv preprint arXiv:2602.23024*, 2026.
- [17] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.
- [18] D. Zheng, S. Huang, L. Zhao, Y. Zhong, and L. Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13624–13634, 2024.
- [19] L. Zhang, X. Hao, Q. Xu, Q. Zhang, X. Zhang, P. Wang, J. Zhang, Z. Wang, S. Zhang, and R. Xu. MapNav: A novel memory representation via annotated semantic maps for vision-and-language navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. arXiv:2502.13451.
- [20] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Proceedings of Robotics: Science and Systems*, 2025. arXiv:2412.04453.
- [21] Z. Wang, J. Li, Y. Hong, Y. Wang, Q. Wu, M. Bansal, S. Gould, H. Tan, and Y. Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. arXiv:2307.15644.
- [22] M. Wei, C. Wan, X. Yu, T. Wang, Y. Yang, X. Mao, C. Zhu, W. Cai, H. Wang, Y. Chen, X. Liu, and J. Pang. StreamVLN: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.
- [23] M. Han, L. Ma, K. Zhumakhanova, E. Radionova, J. Zhang, X. Chang, X. Liang, and I. Laptev. RoomTour3D: Geometry-aware video-instruction tuning for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27586–27596, 2025.
- [24] Z. Gong, R. Li, T. Hu, R. Qiu, L. Kong, L. Zhang, Y. Ding, L. Zhang, and J. Liang. Stairway to success: Zero-shot floor-aware object-goal navigation via LLM-driven coarse-to-fine exploration. *arXiv preprint arXiv:2505.23019*, 2025.

- [25] H. Wang, W. Wang, T. Shu, W. Liang, and J. Shen. Active visual information gathering for vision-language navigation. In *European Conference on Computer Vision (ECCV)*, 2020. arXiv:2007.08037.
- [26] M. Wei, C. Wan, J. Peng, X. Yu, Y. Yang, D. Feng, W. Cai, C. Zhu, T. Wang, J. Pang, and X. Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2512.08186.
- [27] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988. doi:10.1007/BF00133571.
- [28] D. Morrison, P. Corke, and J. Leitner. Multi-view picking: Next-best-view reaching for improved grasping in clutter. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8762–8768, 2019. doi:10.1109/ICRA.2019.8793805.
- [29] M. Breyer, L. Ott, R. Siegwart, and J. J. Chung. Closed-loop next-best-view planning for target-driven grasping. *arXiv preprint arXiv:2207.10543*, 2022.
- [30] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. Guibas, and F. Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465, 2024.
- [31] W. Cai, Y. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao. SpatialBot: Precise spatial understanding with vision language models. In *European Conference on Computer Vision (ECCV)*, 2024. arXiv:2406.13642.
- [32] C. H. Song, V. Blukis, J. Tremblay, S. Tyree, Y. Su, and S. Birchfield. RoboSpatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2411.16537.
- [33] R. Xu, W. Wang, H. Tang, X. Chen, X. Wang, F.-J. Chu, D. Lin, M. Feiszli, and K. J. Liang. Multi-SpatialMLLM: Multi-frame spatial understanding with multi-modal large language models. *arXiv preprint arXiv:2505.17015*, 2025.
- [34] R. Yang, Z. Zhu, Y. Li, J. Huang, S. Yan, S. Zhou, Z. Liu, X. Li, S. Li, W. Wang, Y. Lin, and H. Zhao. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025.
- [35] S. Ravi, G. Sarch, V. Vineet, A. D. Wilson, and B. T. Kumaravel. Out of sight, not out of context? egocentric spatial reasoning in VLMs across disjoint frames. *arXiv preprint arXiv:2505.24257*, 2025.
- [36] Z. Liu, Y. Du, T. Fu, S. Su, C. Ho, and C. Wang. Vision-language memory for spatial reasoning. *arXiv preprint arXiv:2511.20644*, 2025.
- [37] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. RoboPoint: A vision-language model for spatial affordance prediction in robotics. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 4005–4020. PMLR, 2025. URL <https://proceedings.mlr.press/v270/yuan25c.html>.
- [38] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. E. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, A. Zeng, K. Hausman, N. Heess, C. Finn, S. Levine, and B. Ichter. PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2402.07872.

- [39] K. Wang, P. Zhang, Z. Wang, Y. Gao, L. Li, Q. Wang, H. Chen, Y. Lu, Z. Yang, L. Wang, R. Krishna, J. Wu, L. Fei-Fei, Y. Choi, and M. Li. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/fc6688d75adde86b9df910769c1d02e3-Abstract-Conference.html.
- [40] Qwen Team. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.
- [41] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, and Z. Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410. Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.acl-demos.38. URL <https://aclanthology.org/2024.acl-demos.38/>.
- [42] OpenAI. GPT-4o system card. <https://openai.com/index/gpt-4o-system-card/>, 2024.
- [43] R. Yang, H. Chen, J. Zhang, M. Zhao, C. Qian, K. Wang, Q. Wang, T. V. Koripella, M. Movahedi, M. Li, H. Ji, H. Zhang, and T. Zhang. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 70576–70631. PMLR, 2025. URL <https://proceedings.mlr.press/v267/yang25f.html>.
- [44] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020.
- [45] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans. ObjectNav revisited: On evaluation of embodied agents navigating to objects. In *Proceedings of the 4th Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*. PMLR, 2021. arXiv:2006.13171.
- [46] OpenAI. GPT-5.5 system card. <https://openai.com/index/gpt-5-5-system-card/>, 2026.
- [47] OpenAI. OpenAI o3 and o4-mini system card. <https://openai.com/index/o3-o4-mini-system-card/>, 2025.
- [48] Google DeepMind. Gemini 3 Flash. <https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/>, 2025.
- [49] Anthropic. Claude Sonnet 4.6 system card. <https://www.anthropic.com/system-cards>, 2026.
- [50] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- [51] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang. FAST-LIO2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022. doi:10.1109/TRO.2022.3141876.
- [52] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. doi:10.48550/arXiv.2511.16719.

A Method Details

Teacher navigation rules. The teacher turns when the target or path-segment bearing exceeds 15° . Heading errors above 40° are split across multiple turns. When aligned, the teacher selects a `nav_to_point` goal. For a visible or recently visible target, it selects floor points that preserve visibility and approach a reachable region around the object or receptacle. Otherwise, it samples floor points near the projected short-horizon path, favoring points away from navigability boundaries.

Trajectory selection and records. For the full AIDA dataset, collection attempts cover 13,608 episodes, of which 11,740 succeed. Removing 1,528 successful episodes longer than 16 steps leaves 10,212 episodes and 67,183 training examples. Each step stores the RGB observation, instruction, scene-key-frame order, task phase, skill, and argument. Additional simulator records—robot pose, target positions, mask statistics, turn mode, and navigation pixel—support offline annotation and diagnostics.

Annotation and output format. The templates use masks for visible-target descriptions and recorded poses for out-of-view direction labels (Section 3.4). The five output fields map to the agent representation as follows: `state_estimation` and `reasoning` together represent the pre-action reasoning r_t ; `action` and `action_information` encode (a_t, u_t) ; and `summarization` stores the step summary m_t .

B Additional Evaluation Details

EmbodiedBench. We use the VAGEN/EmbodiedBench split [39, 43]. Hosted models share the prompt, output schema, parser, invalid-action handling, executor, and 10-step horizon. Scores average two runs; Table 4 lists public model family names [46, 42, 47, 48, 49, 50, 40].

Real-robot perception and navigation. We map the office with FAST-LIO2 [51] and manually select 15 scene key frames. The VLM receives RGB observations from the torso RGB-D camera. The `search_scene_frame` skill navigates to the selected key-frame pose. For `nav_to_point`, the executor back-projects the selected pixel using depth and plans an obstacle-free path with the fast marching method.

Real-robot manipulation. The executor expresses each pick or place target in the robot base frame and sends an end-effector pose command to the Atribot whole-body controller. Grasping uses a predefined pose. Near the object, head-camera visual servoing uses SAM 3 [52] to refine the grasp position. Low-level grasp failures count toward the reported end-to-end failure rate.

C Failure Analysis

Scene-frame retrieval accounts for the largest failure share in both settings (Figure 5). Grasp execution and target acquisition are also substantial failure sources on the real robot, highlighting the need for reliable manipulation skills and visual guidance during approach.

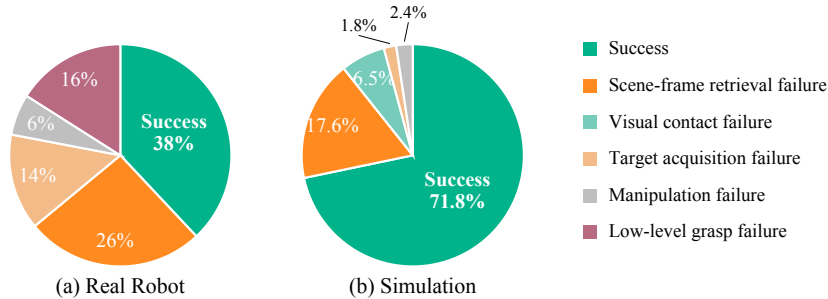


Figure 5: **Task outcomes.** Real-robot success includes low-level grasp failures, shown separately.

D OWMM Training and Evaluation Hyperparameters

Table 5 summarizes the training and episode-evaluation configuration for the AIDA-OWMM policy used in the OWMM experiments.

Table 5: **AIDA-OWMM model, training, and evaluation configurations.** The shared prompt and output format is defined in Section 3.5; evaluation uses the trained LoRA adapter.

Parameter	Value
<i>Base model and adapter</i>	
Base VLM	Qwen3-VL-8B-Instruct
Fine-tuning method	Supervised fine-tuning with LoRA
LoRA rank / alpha / dropout	64 / 128 / 0.05
LoRA target modules	a11 attention and feed-forward linear layers
Rank-stabilized LoRA	Enabled
Trust remote code	Enabled
<i>Training data and prompt schema</i>	
Prompt template	qwen3_vl_nothink
Stage inputs	Search: system prompt, scene key frames, current RGB, task, skill definitions, episode history, output schema. Navigation, pick, and place: system prompt, current RGB, task, skill definitions, episode history, output schema.
Privileged inputs excluded from training and inference	Object mask, target/goal coordinates, projected navigation overlay, privileged distance, and teacher action.
Output fields	state_estimation, reasoning, action, action_information, summarization
Cutoff length	4096 tokens
Image / video max pixels	262144 / 16384
Preprocessing / dataloader workers	16 / 4
<i>Training optimization</i>	
Per-device train batch size	3
Gradient accumulation steps	2
Effective batch size	$6 \times \text{number of training GPUs}$
Learning rate	5.0×10^{-5}
Training epochs	3.0
LR scheduler / warmup	Cosine / 0.03 warmup ratio
Precision	bfloat16
Gradient checkpointing	Enabled
Logging / checkpoint interval	10 / 500 steps
Checkpoint retention	3 checkpoints
Validation during training	Disabled
<i>Episode evaluation</i>	
VLM decoding	512 max output tokens, temperature 0.0, top- p 0.5
Max VLM decisions per episode	10
Episode history cap	30 previous steps; no previous RGB history images
Parallel evaluation	20 workers across four GPUs
Resume unfinished episodes	Enabled
Pick timing at deployment	Selected by the VLM from its current observation
Grasp threshold override	Habitat default