WHERE TO STAND FOR MANIPULATION: AFFORDANCE PREDICTION AND FEASIBILITY-GUIDED STANDING REFINEMENT FOR LAST-METER MOBILE MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Last-meter mobile manipulation requires a robot not only to navigate toward a task-relevant object, but also to terminate at a standing pose that enables successful downstream manipulation. However, existing navigation objectives primarily guide the robot to the target vicinity, without explicitly accounting for whether the resulting standing pose is feasible for the subsequent manipulation. We propose StandReady, which predicts manipulation-ready standing affordances from a single egocentric observation and a task instruction. To incorporate downstream manipulation feasibility, StandReady introduces manipulation-feasibility-guided standing refinement, which evaluates candidate standing poses using reference end-effector trajectories during training and uses the resulting trajectory-level rewards to refine affordance supervision. Experiments on the RoboCasa benchmark demonstrate that StandReady selects more manipulation-ready terminal poses and consistently improves downstream manipulation success over existing baselines. Ablation studies confirm that both feasibility-guided refinement and the trajectory-level reward design contribute to the performance gains. Overall, StandReady bridges the gap between navigation-oriented terminal-pose selection and the requirements of downstream manipulation.

1 INTRODUCTION

Mobile manipulation tasks require robots to perform a sequential pipeline of perception, navigation, and manipulation operations in complex indoor environments. Recent advances have enabled robots to identify task-relevant objects from high-level instructions and navigate to their vicinity Yenamandra et al. (2023); Yokoyama et al. (2024); Zhang et al. (2024b); Li et al. (2020); Wong et al. (2022); Sun et al. (2022); Uppal et al. (2024); Ehsani et al. (2024). However, reaching the interaction area does not by itself ensure successful manipulation, because the robot's terminal mobile-base pose, which we refer to as its standing pose, defines the initial spatial conditions for arm execution. Selecting a manipulation-ready standing pose is therefore essential for reliable last-meter mobile manipulation.

The terminal standing pose is both the endpoint of base motion and the starting condition for downstream manipulation. The robot's standing pose simultaneously affects the target's visual appearance, the relative configuration between the object and the robot's manipulation workspace, the spatial relationship between the base and surrounding obstacles, and the execution of subsequent manipulation motions Zhang et al. (2024a); Naik et al. (2024); Makhal & Goins (2017); Zhang et al. (2023); Shao et al. (2024). A standing location may satisfy conventional navigation requirements, such as being sufficiently close to the target, maintaining target visibility, and allowing safe base placement, yet still lead to unfavorable conditions for manipulation execution. As shown in Fig. 1, two navigation-valid terminal standings for the same downstream task can therefore lead to substantially different manipulation conditions. Navigation validity alone is therefore insufficient: a terminal standing pose must also support the downstream manipulation motion.

Recent studies have recognized the importance of terminal standing poses in mobile manipulation and have explored standing quality from different perspectives. Some approaches learn standing

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.

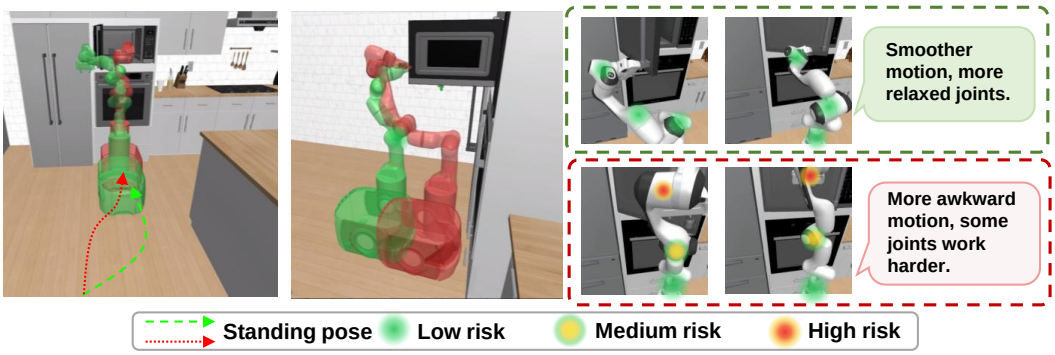


Figure 1: Terminal standing pose selection before downstream manipulation. After standing affordance prediction and navigation, both terminal standing poses reach the target vicinity, while the green standing pose yields smoother motion and lower trajectory-level risk.

affordances from manipulation success outcomes Zhang et al. (2025), while others seek viewpoints or base poses that are more compatible with existing manipulation policies Yang et al. (2025). Other methods incorporate task semantics, target distance, obstacle layouts, or robot models to select feasible standing candidates Lin et al. (2026); Zhang et al. (2024a); Naik et al. (2024); Rosen et al. (2023); Wu et al. (2025). These efforts show that terminal standing selection should go beyond navigation-valid positioning and account for the manipulation process that follows. However, existing approaches do not fully address how embodiment-aware feasibility evaluated over a complete manipulation trajectory can be converted into supervision for direct standing-affordance prediction from a single egocentric observation. The underlying difficulty is that endpoint- or outcome-level signals may overlook poor intermediate configurations along the manipulation motion. Our key insight is that trajectory-level manipulation feasibility can be transferred from offline candidate evaluation into supervision for a direct standing-affordance predictor.

To realize this insight, we propose StandReady, which formulates last-meter standing selection as task-conditioned affordance prediction from a single egocentric RGB-D observation and a manipulation instruction. StandReady is trained in two stages to capture both task relevance and downstream manipulation readiness. The first stage learns standing affordances from VLM-generated targets, providing visual-semantic priors over plausible standing regions. The second stage introduces manipulation-feasibility-guided standing refinement to incorporate downstream execution conditions into affordance learning. Instead of repeatedly executing the downstream policy to evaluate candidate poses, StandReady reuses offline manipulation trajectories to construct trajectory-level feasibility supervision during training. It thereby incorporates downstream manipulation feasibility into standing prediction while retaining single-forward inference at deployment.

Finally, we evaluate StandReady in simulation and on a physical mobile manipulator. Simulation results demonstrate improved downstream manipulation success, while real-world experiments provide preliminary evidence of transfer through terminal-pose readiness evaluation.

The main contributions of this paper are summarized as follows:

- We propose StandReady, a last-meter mobile manipulation framework that predicts standing affordances from a single-frame egocentric observation and a manipulation instruction, enabling feasibility-guided standing selection that provides a suitable initial execution condition for the subsequent manipulation process.

- We introduce manipulation-feasibility-guided standing refinement, which converts feasibility evaluated over complete reference end-effector trajectories into progressively refined affordance supervision using fixed offline manipulation assets.

- We evaluate StandReady on five RoboCasa atomic manipulation tasks, demonstrating improved downstream task success over representative baselines and validating the proposed refinement through ablation studies.

2 Related Work

Mobile Manipulation. Mobile-manipulation systems coordinate perception, navigation, and arm control to complete object-interaction tasks in complex indoor environments. HomeRobot and OK-Robot combine semantic perception, object search, navigation, and manipulation for open-vocabulary household tasks Yenamandra et al. (2023); Liu et al. (2024), while Skill Transformer learns skill selection and whole-body control within a unified policy Huang et al. (2023). LeLaN further studies language-guided last-mile navigation for precise approach to visible target objects Hirose et al. (2025). These methods advance open-world task execution and target approach, but terminal standing quality is generally implicit in navigation or policy execution rather than explicitly predicted according to downstream manipulation requirements.

Manipulation-Aware Standing Pose Selection. More closely related methods evaluate base or standing poses according to their suitability for manipulation. MoMa-Kitchen derives floor-level standing affordances from sampled manipulation outcomes, and its NavAff model predicts manipulation-ready regions from egocentric observations Zhang et al. (2025). Mobi-π searches for base poses whose observations are compatible with a trained manipulation policy while satisfying visibility and collision constraints Yang et al. (2025). Semantic-geometric approaches combine task affordances, interaction directions, preferred distances, and collision-aware sampling to generate feasible base positions Lin et al. (2026). Motion-aware approaches further incorporate robot execution constraints through learned pre-grasp approaching behaviors or planning over base placements and arm paths Naik et al. (2024); Zhang et al. (2024a); Makhal & Goins (2017); Zhang et al. (2023); Shao et al. (2024); Rosen et al. (2023); Wu et al. (2025). These methods define standing quality through manipulation outcomes, policy compatibility, semantic-geometric constraints, or explicit motion reasoning. However, these objectives mainly provide outcome-level, endpoint-level, or search-time signals, rather than converting trajectory-level manipulation feasibility into supervision for direct standing affordance prediction. A standing pose may satisfy visibility, collision-free base placement, or endpoint reachability while still inducing unfavorable intermediate arm motions. StandReady addresses this gap by using offline trajectory-level manipulation feasibility evaluation to supervise a direct standing affordance predictor.

3 Method

3.1 Overview

Task Formulation. We formulate last-meter mobile manipulation as a standing prediction problem before executing a fixed downstream manipulation policy. Given a single egocentric RGB-D observation o captured before last-meter approach and a manipulation instruction τ, the robot selects a terminal mobile-base pose $a \in SE(2)$, referred to as a *standing pose*.

Given a prescribed arm configuration q_0 at the navigation-to-manipulation handoff, executing the fixed manipulation policy π from different standings induces different robot–task spatial configurations and downstream execution conditions. The desired standing can therefore be written as

$$a^* = \arg \max_{a \in SE(2)} \mathcal{J}(\pi \mid o, \tau, a, q_0), \tag{1}$$

where $\mathcal{J}(\pi \mid o, \tau, a, q_0)$ denotes the expected downstream manipulation performance when policy π is executed from standing a. Directly estimating $\mathcal{J}$ for every candidate would require repeatedly executing the downstream policy. We therefore use a training-time surrogate that measures whether a reference manipulation motion remains reachable and well-conditioned, while also remaining compatible with the local base geometry. The reference trajectory is associated with the instructed task, so refinement is task- and robot-specific to the extent that the downstream manipulation assets are task- and robot-specific.

Framework. StandReady learns task-conditioned standing affordances from VLM-generated priors, then refines them based on manipulation feasibility. At each refinement round, the current predictor proposes candidate standings, Manipulation Feasibility Validation scores them offline, and the resulting rewards update affordance supervision for fine-tuning. Offline manipulation assets re-

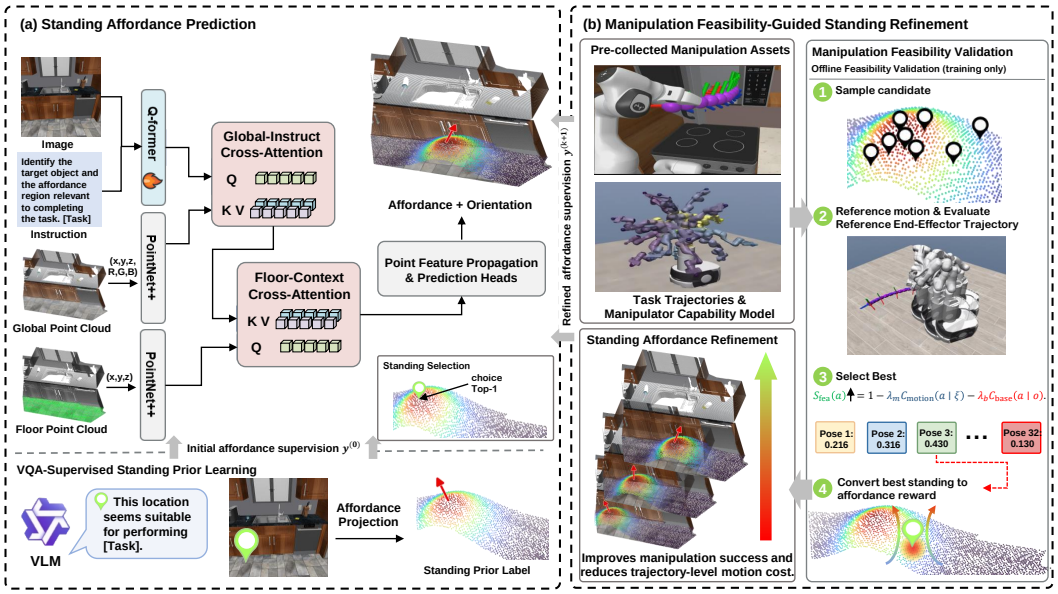


Figure 2: Overview of StandReady. (a) Standing Affordance Prediction learns standing affordances and orientations from RGB-D observations and instructions, with VQA-supervised standing priors providing task-relevant supervision. (b) Manipulation-Feasibility-Guided Standing Refinement uses Manipulation Feasibility Validation with reference end-effector trajectories to refine affordance supervision during training, without introducing additional test-time optimization.

main fixed, while candidate standings and supervision update with the predictor. The evaluator is used only during training; deployment requires a single forward pass.

3.2 Standing Affordance Prediction

Given observed floor points $\mathbf{P}_f = \{p_i\}_{i=1}^{M}$ with $p_i = (x_i, y_i)$, the predictor outputs

$$f_\phi(o, \tau, \mathbf{P}_f) = \{(s_i, \theta_i)\}_{i=1}^{M}, \tag{2}$$

where $s_i \in [0, 1]$ and θ_i denote the standing affordance score and base yaw at p_i. As shown in Fig. 2(a), the Q-Former Dai et al. (2023) extracts instruction-conditioned visual queries from image features, while PointNet++ Qi et al. (2017) encodes global scene and floor point clouds into corresponding features. These three feature streams are fused in stages for point-wise prediction.

The terminal standing is $\hat{a} = (x_{i^*}, y_{i^*}, \theta_{i^*}) \in SE(2)$, where $i^* = \arg\max_{i \in \mathcal{V}} s_i$ and $\mathcal{V}$ indexes the observed valid floor points.

Querying a vision-language model (VLM) offline via visual question answering (VQA) yields responses that are projected onto the observed floor to form initial affordance targets $y^{(0)}(p_i)$. Orientation labels are generated by projecting the target-plane normal onto the ground plane. These targets encode task-relevant standing preferences without explicitly evaluating downstream arm motion.

3.3 Manipulation-Feasibility-Guided Standing Refinement

A standing pose for manipulation should be evaluated by the motion it enables after navigation, not only by where it is located in the observed scene. StandReady therefore uses Manipulation Feasibility Validation during training to refine standing affordance supervision toward candidates that better support the complete reference manipulation motion.

Manipulation Feasibility Validation. The validator converts reusable offline manipulation assets into a standing-level reward. We use the set of successful trajectories collected from training scenes; for a training instance, ξ denotes the task-matched reference trajectory used by the validator,

$$\xi = \{T_{s,e}^t\}_{t=1}^{T}, \tag{3}$$

where $T_{s,e}^t$ denotes the end-effector pose at waypoint t in a common scene frame. We also precompute robot-centric reachability and dexterity workspaces from the robot model, while the current RGB-D observation provides local geometry for base-footprint compatibility.

We evaluate the manipulation-motion conditions at all waypoints of the reference trajectory. This provides waypoint-aggregated trajectory feasibility: the validator evaluates the complete reference motion at discrete waypoints, but does not claim continuous-time dynamic feasibility beyond the implemented checks. Specifically, waypoint-wise IK yields arm configurations q_a^t for reachable waypoints. For each reachable configuration, we compute the local end-effector manipulability $D(q_a^t)$ from the translational Jacobian Yoshikawa (1985) and the singularity-related conditioning $\kappa(q_a^t)$ from the full geometric Jacobian Togai (1986). The two terms are retained because manipulability measures local translational motion capacity, whereas conditioning penalizes near-singular configurations.

We aggregate reachability, manipulability, and singularity-related conditioning over the reference trajectory into a normalized motion penalty $C_{\text{motion}}(a \mid \xi)$, where lower values indicate more favorable manipulation-motion conditions. The base-footprint penalty $C_{\text{base}}(a \mid o)$ is computed from observation-derived local geometry and measures collision or compatibility cost of the mobile base. The feasibility reward is then defined as

$$S_{\text{fea}}(a) = 1 - \lambda_m C_{\text{motion}}(a \mid \xi) - \lambda_b C_{\text{base}}(a \mid o), \tag{4}$$

where λ_m and λ_b balance manipulation-motion feasibility and base-footprint compatibility. This reward serves as a practical training-time surrogate for downstream manipulation, rather than an unbiased estimate of policy success, and only aims to construct refined affordance supervision.

Standing Refinement with Feasibility Reward. At refinement round k, the current predictor, parameterized by ϕ_k, produces point-wise standing affordances and base orientations over the observed floor region. We sample N valid candidate standings from high-affordance regions, subject to floor-validity checks, $\mathcal{C}_{\phi_k}(o, \tau) = \{a_i\}_{i=1}^N$, where each candidate combines a sampled floor location with the orientation predicted by the current model. The feasibility validator evaluates these candidates with S_{fea}, and the most feasible candidate is selected as

$$\tilde{a}_k = \arg \max_{a \in \mathcal{C}_{\phi_k}(o,\tau)} S_{\text{fea}}(a). \tag{5}$$

Writing $\tilde{p}_k$ for the floor location of $\tilde{a}_k$, the refined target is constructed by increasing the affordance response around that location while retaining the Stage I prior:

$$y^{(k+1)}(p) = (1 - \alpha) y^{(0)}(p) + \alpha S_{\text{fea}}(\tilde{a}_k) G_\sigma(p - \tilde{p}_k), \tag{6}$$

where α and σ are the target-mixture coefficient and spatial bandwidth of Gaussian kernel G, respectively. The orientation target at the selected location remains the generated orientation label. The predictor is fine-tuned on the updated targets. Because candidates are induced by the current predictor, both the candidate set and refined supervision evolve across refinement rounds, while the offline manipulation assets remain fixed.

4 Experiments

4.1 Overview

Simulation Setup. We evaluate StandReady in RoboCasa Nasiriany et al. (2026) following the RoboCasa-based mobile manipulation setting used by Mobi-π Yang et al. (2025). Each method selects a terminal standing pose before executing the downstream manipulation policy. We consider five atomic manipulation tasks: Close Door, Close Drawer, Turn on Microwave, Turn on Stove, and Turn on Faucet. For each task, we evaluate each method on 100 rollouts, covering ten initial positions selected with fixed seeds in each of ten RoboCasa test environments. We fine-tune the RoboCasa-provided $\pi_{0.5}$ human300 policy Intelligence et al. (2025) on RoboCasa atomic-task trajectories. The same fine-tuned policy is used across all $\pi_{0.5}$-based methods, so the comparison focuses on terminal standing selection. For Mobi-π and BC w/ Nav, Fig. 3 includes both the original results and variants evaluated with the same fine-tuned $\pi_{0.5}$ used by StandReady and NavAff, so that

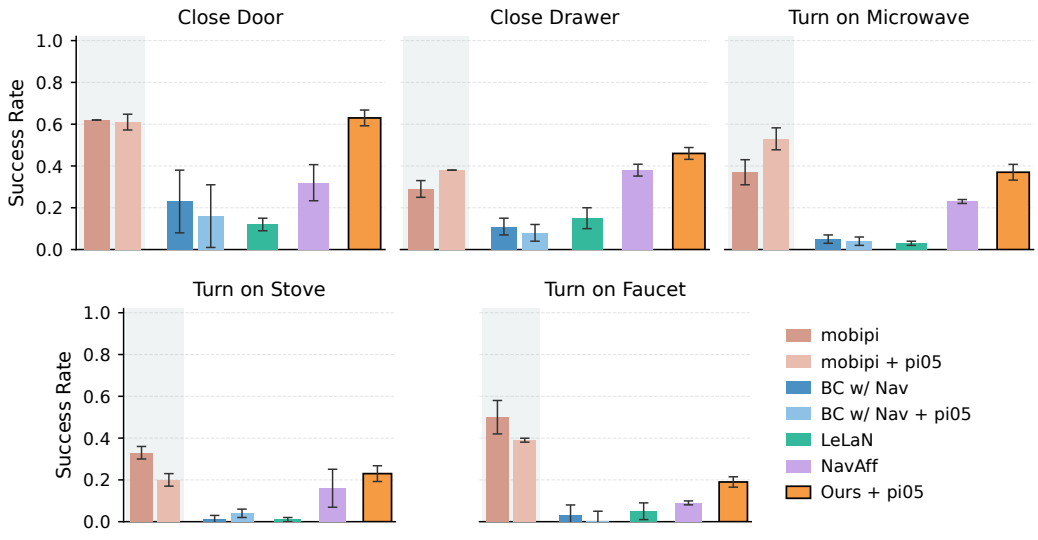


Figure 3: Simulation results on RoboCasa. We evaluate navigation followed by five atomic manipulation tasks: Close Door, Close Drawer, Turn on Microwave, Turn on Stove, and Turn on Faucet. Bars report mean task success rates over three random seeds, and error bars denote standard deviations across seeds.

differences caused by the downstream policy are separated from differences in terminal standing selection. A rollout is counted as successful only when the selected standing pose enables completion of the subsequent manipulation task. We report success rate (SR) averaged over three random seeds, with standard deviation shown as error bars.

Implementation Details. StandReady is trained in two stages. For Stage I, we collect about 100K egocentric RGB-D observations from approximately 2,000 RoboCasa scenes and query Qwen3.5-397B Qwen Team (2026) with VQA-style prompts to annotate task-relevant standing regions. The VQA responses are projected onto the observed floor point cloud to generate standing affordance supervision. Orientation labels are generated geometrically from the target-plane normal projected onto the ground plane. For Stage II, candidate standings are evaluated by the manipulation feasibility reward in Eq. (4); we set $\lambda_m = 2.0$ and $\lambda_b = 1.0$ for all tasks, chosen empirically from preliminary trials and kept fixed. For Manipulation Feasibility Validation, we collect 198 successful task trajectories from training scenes and precompute the robot-centric reachability and dexterity workspaces once. At test time, the feasibility validator is removed, and StandReady selects the terminal standing pose with a single forward prediction.

Baselines. We compare StandReady with four representative baselines covering complementary routes to terminal standing selection. **Mobi-π** Yang et al. (2025) is a policy-aware standing-search method that reconstructs the test scene with 3D Gaussian Splatting and searches for a base pose compatible with the downstream manipulation policy. **BC w/ Nav** Shafiullah et al. (2022) learns base motion and manipulation jointly from demonstrations, representing an end-to-end rollout baseline without explicit terminal standing prediction. **LeLaN** Hirose et al. (2025) is a language-conditioned last-meter approaching method that predicts base actions from egocentric observations and instructions. **NavAff** Zhang et al. (2025) predicts floor-level manipulation affordances from egocentric observations and point-cloud features, making it the closest direct affordance-prediction baseline to StandReady.

4.2 Main Results

The main simulation experiments evaluate whether manipulation-feasibility-guided refinement improves terminal standing selection before downstream manipulation, while retaining direct single-forward deployment. Fig. 3 reports success rates on five RoboCasa atomic manipulation tasks. Among methods without scene-specific standing search at test time, NavAff is the strongest base-

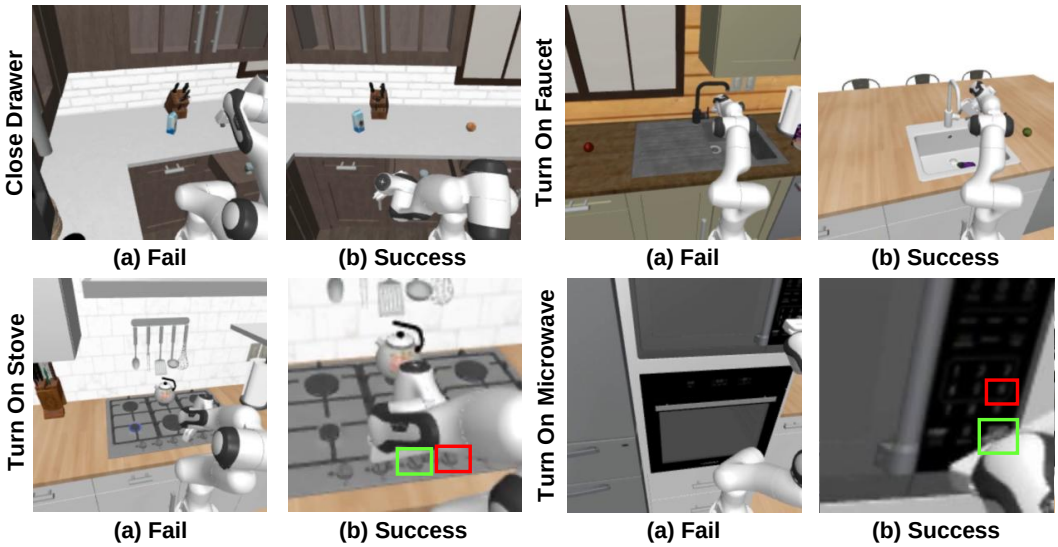


Figure 4: Qualitative analysis of how terminal standing affects downstream execution. Representative cases show that standing selection changes policy-relevant visual evidence, the balance between manipulation-motion feasibility and base-footprint compatibility, and fine-grained grounding among nearby controls. For Stove and Microwave, green boxes mark correct controls and red boxes mark distractors.

line and the closest direct affordance-prediction comparison. StandReady improves the average SR from 0.24 to 0.38, with absolute gains of 0.31 on Close Door, 0.08 on Close Drawer, 0.14 on Turn on Microwave, 0.07 on Turn on Stove, and 0.10 on Turn on Faucet. These improvements indicate that refining standing affordance supervision with trajectory-level manipulation feasibility provides useful motion-conditioned guidance beyond generic floor-level manipulation affordances.

Mobi-π represents a complementary search-based reference that performs policy-aware candidate evaluation at test time. It matches StandReady on Close Door and Turn on Microwave, and achieves higher SR on Turn on Stove and Turn on Faucet, showing the benefit of policy-specific scene search for some precise interaction tasks. StandReady instead moves candidate evaluation into training and selects a standing pose with a single forward prediction at deployment. In our setup, Mobi-π requires 452.24 ± 29.40s, 16.59GB RAM, and 10.47GB VRAM, whereas StandReady requires 0.402s, 3.59GB RAM, and 2.91GB VRAM. This comparison suggests that StandReady preserves competitive manipulation-aware standing selection with substantially lower test-time cost.

Fig. 4 further analyzes how the selected standing pose affects the downstream performance in Eq. 1. In Close Drawer, StandReady selects a more front-facing standing pose, as shown in Fig. 4(b), which better exposes the drawer front and its interaction state to the downstream policy. By contrast, Mobi-π tends to favor a pose close to the policy data-collection viewpoint; this policy-compatible view can become less reliable when the target region is not visually salient, which is also reflected by the Close Drawer results in Fig. 3. In Turn on Faucet, the feasible terminal region lies near areas with high C_{motion} and C_{base}, so success requires balancing arm-motion feasibility with base-footprint compatibility. The stronger performance of Mobi-π on this task suggests that additional test-time candidate optimization can be beneficial when such a balance is narrow. For Turn on Microwave and Turn on Stove, the main challenge is fine-grained semantic alignment: the robot must interact with the correct button or switch among nearby distractors. These cases show that, although the observation o and instruction τ constrain the task, optimizing the standing pose a can substantially improve the execution conditions of the fixed policy π by changing the visual, geometric, and motion-related conditions under which manipulation starts.

4.3 Ablation Study

This ablation study examines whether Stage II refinement can turn the Stage I standing prior into more manipulation-ready supervision. As shown in Table 1, starting from the S1 VQA baseline

Training setup	Success Rate (SR) ↑ Close Door	Close Drawer	Turn on Microwave	Turn on Stove	Turn on Faucet	Overall Average
S1: VQA	0.39	0.32	0.17	0.17	0.09	0.23
S2: R	0.57 (+0.18)	0.32 (+0.00)	0.31 (+0.14)	0.17 (+0.00)	0.15 (+0.06)	0.31 (+0.08)
S2: $R + C_b$	0.53 (+0.14)	0.44 (+0.12)	0.23 (+0.06)	0.21 (+0.04)	0.09 (+0.00)	0.30 (+0.07)
S2: $C_b + C_m$ early	0.37 (-0.02)	**0.48** (+0.16)	0.17 (+0.00)	0.21 (+0.04)	0.07 (-0.02)	0.26 (+0.03)
S2: $C_b + C_m$ final	**0.63** (+0.24)	0.46 (+0.14)	**0.37** (+0.20)	**0.23** (+0.06)	**0.19** (+0.10)	**0.38** (+0.15)

Table 1: Ablation of Stage II reward components and refinement rounds. S1 denotes Stage I VQA supervision, and S2 denotes manipulation-feasibility-guided refinement. R, C_b (C_{base}), and C_m (C_{motion}) denote reachability, base-footprint cost, and trajectory-level manipulation-motion cost, respectively. The early and final rows report checkpoints from the S2 refinement using C_b and C_m. Parenthesized values denote absolute SR changes over the S1 baseline.

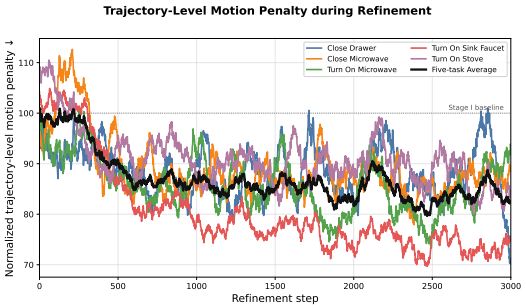


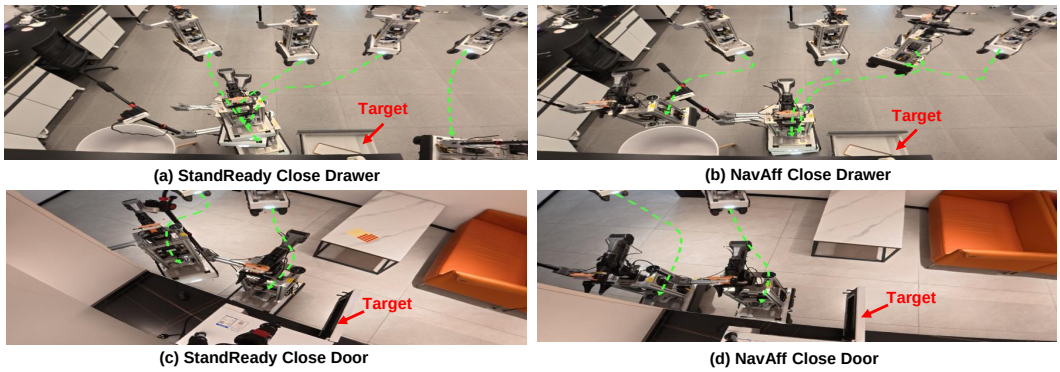
Figure 5: Evolution of trajectory-level motion penalty during manipulation-feasibility-guided refinement. Values are normalized by the Stage I baseline. Lower C_{motion} indicates more favorable manipulation conditions along the reference end-effector trajectory.

with an average SR of 0.23, adding the reachability term in S2 improves the average SR to 0.31, indicating that a basic robot-centric feasibility signal already corrects part of the visually plausible but physically unfavorable standings. Adding the base-footprint cost gives comparable overall performance and improves Close Drawer, where local base placement strongly affects the terminal standing. With both C_b and C_m, the final S2 model achieves the best average SR of 0.38 and improves over S1 on all five tasks, showing that refinement is most effective when base compatibility and trajectory-level manipulation-motion conditions are considered together.

The early and final checkpoints further show that refinement is a progressive supervision-updating process rather than a one-step relabeling procedure. The early checkpoint already shifts the affordance target toward feasibility-preferred regions and gives the strongest Close Drawer result, while the final checkpoint achieves the best overall SR after candidates and refined supervision are updated across rounds. Fig. 5 provides a complementary diagnostic view: the normalized C_{motion} generally decreases during refinement, indicating that the learned affordance increasingly favors standings with lower trajectory-level motion penalty. Together, Table 1 and Fig. 5 show that Stage II improves downstream execution not by changing the fixed policy π, but by optimizing the standing pose a that determines the execution conditions of manipulation.

4.4 Real-World Experiments

We further evaluate StandReady on a real mobile manipulation platform consisting of a four-wheel-steering base and a Piper robotic arm. The robot observes the scene with two RealSense D435 cameras, and its pose is estimated by a MID360 LiDAR with FAST-LIO2 Xu et al. (2022). We choose Close Drawer and Close Door for real-world evaluation, as they are among the more reliable tasks in simulation and can be repeatedly instantiated in our physical scene. We compare



(a) StandReady Close Drawer (b) NavAff Close Drawer

(c) StandReady Close Door (d) NavAff Close Door

Figure 6: Real-world comparison between StandReady and NavAff on Close Drawer and Close Door. Each subfigure overlays multiple terminal standing poses from different trials after segmenting and aligning robot instances to the recorded scene. StandReady performs better on Close Drawer, while Close Door remains challenging under the structurally complex corridor layout.

with NavAff because both methods require simple observations and a single forward prediction at deployment. For each task and method, we evaluate six selected terminal standing poses with five repeated trials per pose, resulting in 30 real-world trials. For both methods, the observed floor region is segmented by Mask2Former Cheng et al. (2021). For NavAff, Mask2Former is also used to provide the target-object segmentation required for constructing its target-distance feature in the global point cloud.

Since our focus is terminal standing selection, we evaluate whether the selected terminal standing pose is manipulation-ready. A trial is counted as successful if the robot stops at a pose from which the interaction region is visible and the arm has a plausible initial condition for downstream manipulation. As shown in Fig. 6, StandReady selects manipulation-ready terminal standing poses in over 60% of Close Drawer trials, while NavAff reaches about 25%. This gap is mainly caused by the difficulty of accurately segmenting the drawer interaction region in real scenes: because NavAff relies on explicit target segmentation to construct its target-distance point-cloud feature, segmentation errors can directly shift its predicted standing affordance. In contrast, StandReady uses language-conditioned visual queries to associate the instruction with the relevant interaction region, which provides better robustness under imperfect real-world target segmentation.

On Close Door, both methods achieve about 40% manipulation-ready terminal standing poses. NavAff performs better than in Close Drawer because the door region is less ambiguous in the real scene. However, both methods are still affected by the structurally more complex corridor layout, which differs substantially from the RoboCasa environments and leaves limited collision-free space for precise base placement. These results suggest that single-forward standing affordance prediction can transfer to real robots, while precise terminal standing selection in complex physical layouts remains an important challenge.

5 Conclusion

In this paper, we study where a mobile manipulator should stand before executing a downstream manipulation policy. We propose StandReady, a task-conditioned standing affordance predictor trained with manipulation-feasibility-guided standing refinement. By transferring manipulation-feasibility-guided candidate evaluation into training supervision, StandReady improves terminal standing selection in RoboCasa while retaining single-forward deployment. Future work will explore stronger 3D scene representation backbones and combine trajectory-level feasibility with policy-aware visual compatibility, which may reduce cross-task interference during multi-task refinement and improve success on precision-sensitive manipulation tasks.

AI Use Statement

In this work, generative AI tools were used for the following aspects:

- **Dataset Annotation:** We utilized Qwen3.5-397B to generate offline VQA-based standing targets for Stage-I supervision, as detailed in the main paper and Appendix B. All candidate annotations were geometrically grounded, filtered, and verified through our proposed pipeline.
- **Writing Assistance:** Large language models were employed solely for grammatical error checking and language refinement of the manuscript text.

The authors reviewed and verified all AI-assisted material and take full responsibility for the final content of this paper.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010.

Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021.

Wenliang Dai, Junnan Li, DONGXU LI, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267. Curran Associates, Inc., 2023. doi: 10.52202/075280-2142. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9a6a435e75419a836fe47ab6793623e6-Paper-Conference.pdf.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16238–16250, 2024.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Noriaki Hirose, Catherine Glossop, Ajay Sridhar, Oier Mees, and Sergey Levine. Lelan: Learning a language-conditioned navigation policy from in-the-wild video. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 666–688. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/hirose25b.html.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xiaoyu Huang, Dhruv Batra, Akshara Rai, and Andrew Szot. Skill transformer: A monolithic policy for mobile manipulation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10818–10828, 2023. doi: 10.1109/ICCV51070.2023.00996.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Chengshu Li, Fei Xia, Roberto Martín-Martín, and Silvio Savarese. Hrl4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 603–616. PMLR, 2020.

Tzu-Jung Lin, Jia-Fong Yeh, Hung-Ting Su, Chung-Yi Lin, Yi-Ting Chen, and Winston H Hsu. Affordance-guided coarse-to-fine exploration for base placement in open-vocabulary mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18443–18451, 2026.

Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Abhijit Makhal and Alex K. Goins. Reuleaux: Robot base placement by reachability analysis. *arXiv preprint arXiv:1710.01328*, 2017.

Lakshadeep Naik, Sinan Kalkan, and Norbert Krüger. Pre-grasp approaching on mobile robots: A pre-active layered approach. *IEEE Robotics and Automation Letters*, 9(3):2606–2613, 2024. doi: 10.1109/LRA.2024.3358077.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Eric Rosen, Steven James, Sergio Orozco, Vedant Gupta, Max Merlin, Stefanie Tellex, and George Konidaris. Synthesizing navigation abstractions for planning with portable manipulation skills. In *Proceedings of The 7th Conference on Robot Learning*, volume 229, pp. 2278–2287. PMLR, 2023.

Nur Muhammad (Mahi) Shafiullah, Ziehen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: cloning k modes with one stone. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Beichen Shao, Nieqing Cao, Yan Ding, Xingchen Wang, Fuqiang Gu, and Chao Chen. Moma-pos: An efficient object-kinematic-aware base placement optimization framework for mobile manipulation. *arXiv preprint arXiv:2403.19940*, 2024.

Charles Sun, Jędrzej Orbik, Coline Manon Devin, Brian H. Yang, Abhishek Gupta, Glen Berseth, and Sergey Levine. Fully autonomous real-world reinforcement learning with applications to mobile manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 308–319. PMLR, 2022.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Masaki Togai. An application of the singular value decomposition to manipulability and sensitivity of industrial robots. *Siam journal on algebraic discrete methods*, 7(2):315–320, 1986.

Shagun Uppal, Ananye Agarwal, Haoyu Xiong, Kenneth Shaw, and Deepak Pathak. Spin: Simultaneous perception interaction and navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18133–18142, 2024.

Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and Roberto Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 1367–1378. PMLR, 2022.

Zhenyu Wu, Angyuan Ma, Xiuwei Xu, Hang Yin, Yinan Liang, Ziwei Wang, Jiwen Lu, and Haibin Yan. Moto: A zero-shot plug-in interaction-aware navigation for general mobile manipulation. In *Proceedings of The 9th Conference on Robot Learning*, volume 305, pp. 2933–2948. PMLR, 2025.

Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

Jingyun Yang, Isabella Huang, Brandon Vu, Max Bajracharya, Rika Antonova, and Jeannette Bohg. Mobi-π: Mobilizing your robot learning policy. In *Conference on Robot Learning*, pp. 3516–3536. PMLR, 2025.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin S. Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander Clegg, John M. Turner, Zsolt Kira, Manolis Savva, Angel X. Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. Homerobot: Open-vocabulary mobile manipulation. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1975–2011. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/yenamandra23a.html.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48. IEEE, 2024.

Tsuneo Yoshikawa. Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2):3–9, 1985.

Chenyu Zhang, Haohao Qin, Shiying Sun, Yi Pan, Kuan Liu, Tao Li, and Xiaoguang Zhao. Jpmdp: Joint base placement and multi-configuration path planning for 3d surface disinfection with a uv-c robotic system. *Robotics and Autonomous Systems*, 174:104644, 2024a. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2024.104644. URL https://www.sciencedirect.com/science/article/pii/S0921889024000277.

Huiwen Zhang, Kai Mi, and Zhijun Zhang. Base placement optimization for coverage mobile manipulation tasks. *arXiv preprint arXiv:2304.08246*, 2023.

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. *Robotics: Science and Systems(RSS)*, 2024b.

Pingrui Zhang, Xianqiang Gao, Yuhan Wu, Kehui Liu, Dong Wang, Zhigang Wang, Bin Zhao, Yan Ding, and Xuelong Li. Moma-kitchen: A 100k+ benchmark for affordance-grounded last-mile navigation in mobile manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6315–6326, October 2025.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

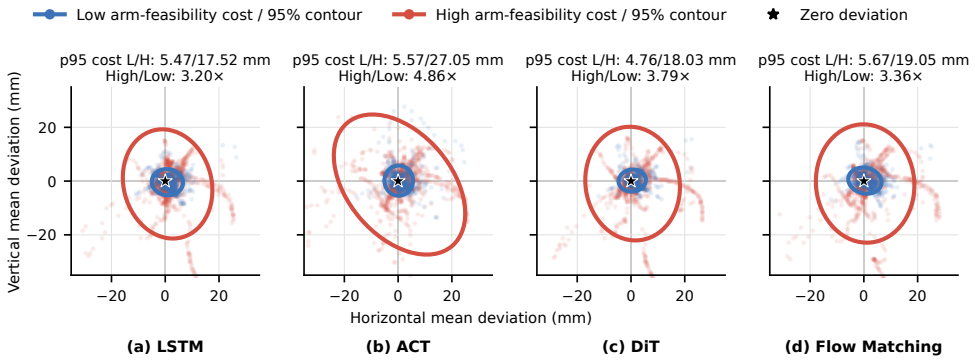


Figure 7: **Policy tracking under low and high arm-feasibility cost.** Points denote mean cross-track deviations over 8-step predicted chunks, projected onto the plane normal to the commanded Cartesian motion. Ellipses show empirical 95% contours, and annotations report p95 deviation with high/low ratios. High-cost trajectories produce larger tail deviation across all four policies.

A Controlled Evaluation of Arm-Feasibility Cost

A.1 Motivation and Scope

Experiments in the main paper evaluate downstream task success after terminal standing selection. However, task success conflates arm-motion feasibility with object appearance, viewpoint variation, language grounding, and the policy training distribution. We therefore conduct a controlled study without visual or language inputs to examine whether trajectory-level arm-feasibility cost is directly associated with policy tracking error when holding the commanded Cartesian motion and training protocol fixed.

This controlled analysis connects to the standing-selection objective in the main paper:

$$a^* = \arg \max_{a \in SE(2)} \mathcal{J}(\pi \mid o, \tau, a, q_0), \tag{7}$$

where different standings induce different execution conditions for the fixed manipulation policy. Rather than evaluating the full task-level objective $\mathcal{J}$, this section isolates the arm-motion component to test whether feasibility cost correlates with policy tracking degradation under identical task demands.

This analysis complements the main-paper task-success evaluations by isolating policy tracking fidelity under controlled initial arm configurations, rather than measuring end-to-end interaction success for a pre-trained vision–language–action policy.

A.2 Controlled Trajectory Corpus

We construct a controlled trajectory corpus using a 7-DoF Franka Emika Panda arm simulated in robosuite/MuJoCo Zhu et al. (2020); Todorov et al. (2012). To remove visual and semantic confounders, all trajectories execute the same 15 cm Cartesian straight-line motion at 20 Hz for 160 control steps, varying only the initial arm configuration. Candidate initial configurations are sampled from a fixed offline arm-configuration prior and evaluated via an 80-step open-loop preview. The preview assigns a trajectory-level arm-feasibility cost based on local Cartesian dexterity, Jacobian conditioning, joint-rate demand, and tracking error. This cost is used solely to construct evaluation strata and is never provided to the learned policies.

We form two strata from the generated corpus: 80 trajectories from the lowest-cost quartile and 80 trajectories from a feasible, moderately high-cost band. The high-cost stratum is filtered to exclude non-executable previews, focusing the comparison on feasible yet kinematically unfavorable conditions rather than trivial execution failures. Because both strata are sub-sampled from a single corpus rather than generated via randomized intervention, they serve as controlled evaluation cohorts rather than a standalone causal comparison.

A.3 Behavior-Cloning Protocol

To examine whether cost-dependent tracking degradation persists across diverse policy architectures, we evaluate four behavior-cloning models trained on the same mixture of low- and high-cost trajectories: an LSTM Graves (2012), Action Chunking with Transformers (ACT) Zhao et al. (2023), a transformer-based diffusion policy (DiT) Liu et al. (2025), and a conditional flow-matching transformer Black et al. (2025). Each policy observes a history of 16 joint configurations and predicts the next 8 stepwise joint increments. Crucially, the policies receive neither cost-stratum labels nor dexterity- or Jacobian-derived features, ensuring that tracking differences arise without explicit cost supervision.

For each of three random seeds, trajectories within each stratum are partitioned into training and held-out validation splits. Checkpoints are selected by the lowest validation loss on the mixed-stratum set under an identical training protocol across all architectures. We then evaluate task-space tracking performance separately on the low- and high-cost strata without introducing visual inputs, language grounding, or cost cues.

A.4 Task-Space Post-Hoc Evaluation

We map predicted joint increments to task-space end-effector positions using forward kinematics at the `gripper0_right_grip_site` frame. Let $\mathbf{u}$ denote the unit vector along the commanded Cartesian trajectory, $\mathbf{p}_0$ its origin, and $\hat{\mathbf{p}}_t$ the predicted end-effector position at future step t. The cross-track error component is

$$\boldsymbol{\ell}_t = \left(\mathbf{I} - \mathbf{u}\mathbf{u}^\top\right)\left(\hat{\mathbf{p}}_t - \mathbf{p}_0\right). \tag{8}$$

For a prediction window w, the scalar trajectory deviation is

$$d_w = \frac{1}{8}\sum_{t=1}^{8} \left\|\boldsymbol{\ell}_t\right\|_2. \tag{9}$$

We compute the 95th percentile of d_w independently for each seed and report the mean across the three seeds. Thus, the reported p95 metric captures tail deviation over the predicted trajectory chunk rather than simple endpoint error.

Fig. 7 demonstrates consistent cost-dependent degradation: across all four architectures, the high-cost p95 trajectory deviation is 3.20–$4.86\times$ that of the low-cost stratum. This degradation also appears in final end-effector errors, although the ranking among architectures shifts after joint-to-task-space mapping. Specifically, DiT achieves the lowest high-cost mean final error, whereas ACT exhibits the largest high-cost tail error. This discrepancy indicates that aggregate joint-space validation loss fails to capture policy sensitivity to unfavorable arm-motion conditions.

The consistent increase in task-space tail error across diverse policy families supports using trajectory-level arm-feasibility cost as an informative standing-refinement signal. While this controlled study confirms a strong association between initial arm feasibility and Cartesian tracking accuracy in the absence of perception, it does not claim an isolated causal link to end-to-end task completion. System-level efficacy is directly established by the task-success and reward-ablation experiments in the main paper.

B VQA-Based Standing Supervision

StandReady employs visual question answering (VQA) exclusively to construct Stage-I training targets. For each training observation o and manipulation instruction τ, we project a target-conditioned set of candidate standing markers onto the visible floor and query Qwen3.5-397B Qwen Team (2026) to rank them from the robot's first-person perspective. The selected marker provides an initial visual-semantic standing prior, which is subsequently refined based on manipulation feasibility. Crucially, the VLM, visual markers, and VQA prompts operate strictly offline during target construction and are omitted at deployment.

Task-conditioned candidate construction. The RGB-D observation provides the visible floor support and local scene geometry, while the instructed interaction target is highlighted in the image. Candidate markers are anchored to observed floor points in target-facing approach regions.



Figure 8: **Prompt and representative VQA response for offline Stage-I standing supervision.** The prompt clarifies that the image is a first-person observation rather than a top-down metric map, defines target-facing approach constraints, and asks the VLM to rank candidate floor locations using visual-semantic and coarse geometric cues. The right panel shows the annotated input, the selected marker, and the corresponding VLM response for one example; candidate identifiers and scores are preserved. VQA is restricted to offline Stage-I target generation.

The candidates span diverse approach directions, lateral offsets, and stand-off distances, preventing supervision from collapsing to the nearest visible floor point. We filter out non-floor points and kinematically incompatible regions prior to querying the VLM. Each retained marker possesses a unique identifier and a metric 3D floor coordinate, allowing the symbolic VQA response to be mapped back to the observed floor point cloud.

VQA ranking with coarse geometric constraints. The prompt provides the manipulation instruction, identifies the interaction target, and clarifies that the image is a first-person perspective view rather than an orthographic metric map. The VLM ranks candidate floor locations according to target-facing approach, proximity to the interaction region, visible clearance around the base, coarse reachability, and alignment with the required manipulation direction. Candidates that fall outside the approach cone, suffer from visual occlusion, or contradict the intended interaction are penalized or excluded. The response scores the viable candidates and outputs a single highest-scoring marker ID.

Projection to dense standing targets. Let p^{VQA} be the 3D floor location of the selected marker. We convert this sparse selection into an initial point-wise affordance target by applying a normalized spatial kernel over the observed floor point cloud:

$$y^{(0)}(p_i) = K\big(\|p_i - p^{\mathrm{VQA}}\|_2\big), \qquad p_i \in \mathbf{P}_f, \tag{10}$$

where $K(\cdot)$ monotonically decreases with planar distance over a bounded local support. This smooth target mitigates sensitivity to discrete marker spacing and minor viewpoint shifts compared to a one-hot label. The target base orientation is obtained by projecting the interaction-plane normal onto the ground plane, consistent with the standing affordance formulation in the main paper. We retain only responses whose selected marker ID resolves to valid candidate metadata.

C Implementation Details

Predictor and input representation. We use the same standing affordance predictor in both training stages. Each observation is represented by an RGB global point cloud with $16{,}384$ points and an observed-floor point cloud with $8{,}192$ points. Two PointNet++ branches encode the global and floor point clouds into 64 latent tokens per branch. The 32 language-conditioned visual queries from

the InstructBLIP Q-Former are 768-dimensional and are projected, together with the point-cloud tokens, to a shared 512-dimensional space. One cross-attention block conditions the global scene tokens on the instruction, and two subsequent blocks condition the floor tokens on the instruction-conditioned global tokens and the visual queries. We use eight attention heads and a dropout rate of 0.1. A feature-propagation decoder predicts, for every observed floor point, an affordance logit, a unit-normalized $(\cos\theta, \sin\theta)$ orientation vector, and an orientation-validity logit.

Manipulation feasibility validation. For each task, the validator uses a task-matched reference end-effector trajectory from fixed offline assets and a precomputed reachability and dexterity workspace for the robot platform in RoboCasa. The reference trajectory is subsampled to at most 30 waypoints while preserving the endpoint. For a candidate standing a, reference waypoints are transformed into the candidate base frame and solved sequentially via damped least-squares inverse kinematics (IK).

Let q_a^t be the arm configuration at waypoint t, and let $J_v(q_a^t)$ and $J(q_a^t)$ denote the translational and full geometric Jacobians, respectively. We compute local translational dexterity Yoshikawa (1985) and singularity-related conditioning Togai (1986) as

$$\begin{aligned} D_t(a) &= \sqrt{\det(J_v(q_a^t) J_v(q_a^t)^\top)}, \\ \kappa_t(a) &= \frac{\sigma_{\max}^t}{\max(\sigma_{\min}^t, \epsilon)}, \end{aligned} \tag{11}$$

where $\sigma_{\min}^t$ and $\sigma_{\max}^t$ are the minimum and maximum singular values of the full geometric Jacobian $J(q_a^t)$, and ϵ prevents division by zero. Larger D_t indicates higher local translational dexterity, while larger κ_t signals ill-conditioning or singularity proximity.

The motion penalty comprises two components. The dexterity term $C_{\mathrm{dex}}(a)$ is computed from normalized workspace dexterity queried along the reference trajectory, assigning higher weights to terminal waypoints to emphasize final manipulation posture. The risk term $C_{\mathrm{risk}}(a)$ aggregates singularity conditioning, IK failures, positional residuals, and discontinuous joint jumps. Both terms are normalized such that larger values indicate less favorable motion conditions.

Reachability serves as a strict eligibility gate before reward computation. Let $u_t^{\mathrm{ws}}(a) \in \{0, 1\}$ indicate whether waypoint t lies within the precomputed reachable workspace. We compute

$$\begin{aligned} \rho_{\mathrm{traj}}(a) &= \frac{1}{T} \sum_{t=1}^{T} u_t^{\mathrm{ws}}(a), \\ \rho_{\mathrm{term}}(a) &= \frac{1}{|\mathcal{T}_{\mathrm{term}}|} \sum_{t \in \mathcal{T}_{\mathrm{term}}} u_t^{\mathrm{ws}}(a). \end{aligned} \tag{12}$$

Here, ρ_{traj} measures full-trajectory workspace coverage, while ρ_{term} evaluates coverage over the terminal segment. A candidate is deemed trajectory-eligible only when the overall trajectory, terminal segment, and endpoint simultaneously satisfy reachability and IK-convergence criteria. This prevents favorable dexterity or base clearance from compensating for an unachievable trajectory segment.

For eligible candidates, the trajectory-level motion penalty is

$$C_{\mathrm{motion}}(a) = 0.7 C_{\mathrm{risk}}(a) + 0.3 C_{\mathrm{dex}}(a), \tag{13}$$

and the feasibility reward used in Stage-II refinement is

$$S_{\mathrm{fea}}(a) = 1 - 2 C_{\mathrm{motion}}(a) - C_{\mathrm{base}}(a). \tag{14}$$

Position-only damped IK uses at most 60 iterations with a $2.5\,\mathrm{cm}$ positional tolerance. The base-footprint term C_{base} is computed as a soft occupancy penalty from the observation depth map over the robot footprint. Thus, the validator balances trajectory-level arm dexterity with base clearance, using reachability gates to filter kinematically invalid candidates.

Standing refinement. Stage-II refinement constructs candidate standings around the current affordance prediction. The highest-scoring valid floor point is retained as the incumbent. The search

additionally evaluates high-affordance floor points within $2\,\mathrm{m}$, prioritizing candidates within $0.5\,\mathrm{m}$ of the incumbent and allocating at most one quarter of the 32-point search budget to farther locations. Points with predicted affordance below 0.05 are excluded.

Each candidate standing pairs a sampled floor location with its model-predicted yaw. The reference trajectory is transformed into the candidate base frame using this full $SE(2)$ pose, evaluating trajectory eligibility, terminal eligibility, and feasibility costs under the predicted heading. Predicted affordance scores define candidate sampling support and do not enter the feasibility reward.

A trajectory-eligible proposal replaces the incumbent if the incumbent fails eligibility checks. If the incumbent is already eligible, a proposal is accepted only when it strictly improves S_{fea}. The planar location of the accepted candidate provides a Gaussian refinement target ($\sigma = 0.15\,\mathrm{m}$) to update standing affordance supervision for that round. If no eligible proposal outperforms the incumbent, supervision remains unchanged.

The orientation target remains the geometrically generated task-facing yaw label. This label supervises the orientation head, while feasibility validation uses the candidate's predicted yaw. The refinement module thus updates the training targets without requiring ground-truth orientation at test time.

Optimization and inference. Stage-I is trained for 10 epochs with AdamW Loshchilov & Hutter (2017), a batch size of 24, two-step gradient accumulation, an initial learning rate of 4×10^{-4}, weight decay of 10^{-4}, one warm-up epoch, cosine decay to 10^{-6}, and gradient clipping at 1.0. Its loss is

$$\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{\mathrm{aff}} + 0.2\mathcal{L}_{\mathrm{valid}} + \mathcal{L}_{\mathrm{yaw}}, \tag{15}$$

where $\mathcal{L}_{\mathrm{aff}}$ is Smooth-L_1 loss on point-wise affordance, $\mathcal{L}_{\mathrm{valid}}$ is binary cross-entropy on orientation validity, and $\mathcal{L}_{\mathrm{yaw}}$ is cosine loss on the unit orientation vector.

Stage-II is initialized from the best Stage-I checkpoint and trained for 4 epochs with a batch size of 20 and four-step gradient accumulation. The predictor learning rate is 4.77×10^{-5}. We adapt only the Q-Former cross-attention query, key, and value projections using LoRA Hu et al. (2022) (rank 8, scale 16, dropout 0.05, learning rate 10^{-5}), leaving the vision encoder and remaining Q-Former parameters frozen. Stage-II loss retains the same three terms with loss weights 1.0, 0.03, and 0.2 for affordance, orientation validity, and yaw, respectively. Training uses 100 warm-up steps followed by cosine decay over $3{,}000$ updates, weight decay of 10^{-4}, gradient clipping at 1.0, and random seed 42.

At inference, the model predicts the highest-scoring valid floor point and its associated yaw in a single forward pass. Standing selection requires no VQA queries, candidate rendering, reference trajectories, workspace lookups, IK feasibility evaluations, base occupancy checks, or downstream policy rollouts.