MobiGuide: Vocabulary-Guided Online Recovery for Mobile Manipulation

Jia Syuen Lim[1], Zhizhen Zhang[1], Peter Bohm[1], Brendan Tidd[2], Zi Huang[1], and Yadan Luo[*1]

[1]The University of Queensland, Australia [2]CSIRO Robotics, Australia [*]Corresponding Author

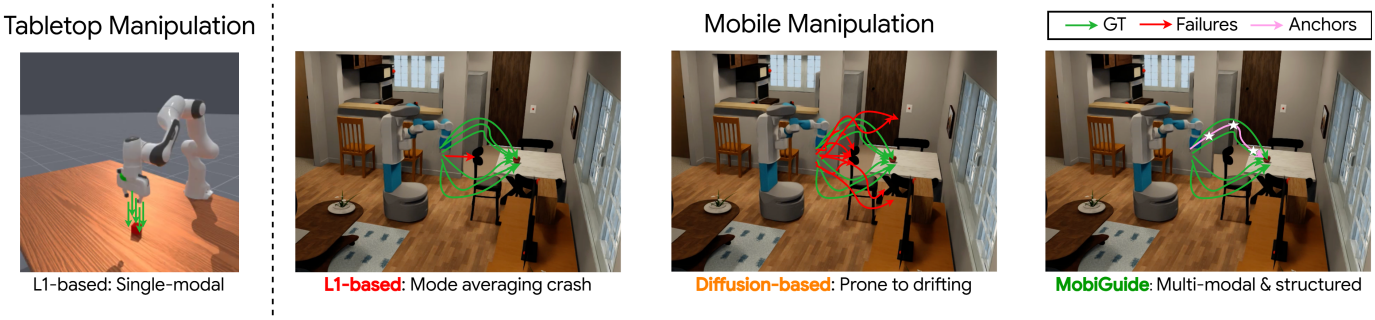


Fig. 1: MobiGuide stabilizes multimodal action generation. Left: Tabletop manipulation is often nearly single-modal, where L_1-based VLA regression is sufficient. Right: Mobile manipulation is inherently *multimodal*; L_1-based policies average across modes and become infeasible, while diffusion-based policies are more expressive but prone to stochastic drift. MobiGuide guides generation with demonstration-derived anchors, preserving multimodal choices while keeping execution stable.

***Abstract*—A central challenge in mobile manipulation is preserving multiple plausible action modes while remaining reactive during execution. For example, grasping a bottle in a cluttered scene can be achieved through multiple physically feasible base–arm motion plans. Deterministic VLA policies trained with L_1 regression are efficient, but they tend to average across valid modes and may produce physically invalid actions. Diffusion-based generative policies can better represent multimodal action distributions, yet their proposals may be weakly constrained when generated from pure noise, and iterative sampling can increase inference cost. Action chunking helps amortize inference, but it also introduces partially open-loop execution, allowing small mismatches to accumulate into drift. We present MobiGuide, a generative VLA model for efficient mobile manipulation, where action generation is guided by a structured trajectory vocabulary. Constructed from expert demonstrations, this vocabulary encodes plausible whole-body motion patterns and provides structured priors for feasible action generation. Given the current observation and instruction, MobiGuide locally refines these reference trajectories into context-compatible action proposals and selects the most suitable trajectory for execution. This preserves multimodal action candidates while keeping generation constrained around feasible expert-like motion modes. To mitigate chunk-induced drift, we further introduce a lightweight residual correction module that performs high-frequency, per-step adjustments during rollout. Across diverse mobile manipulation tasks, MobiGuide improves success and stability under execution disturbances while maintaining low-latency inference.**

I. Introduction

Mobile manipulation is a central challenge in embodied intelligence, requiring semantic reasoning, local base adjustment, and contact-rich arm control near target objects or articulated structures [32, 35]. Its core difficulty lies in the strong temporal coupling between base movement and arm manipulation: early motion decisions shape later viewpoints, object accessibility, and feasible affordances, while small state-estimation or actuation errors can accumulate into task failure. Recent vision-language-action (VLA) models [11, 2, 10, 39, 7, 12, 1, 22, 37, 26, 8] have shown strong promise in mapping visual observations and language instructions directly to actions. However, much of this success has been observed in *tabletop* manipulation settings, where feasible motions are concentrated around a narrow set of solutions. Mobile manipulation is fundamentally less structured: the same instruction may admit multiple valid whole-body executions, differing in local base pose, approach direction, or grasp geometry, making the action distribution inherently *multimodal*. Under such supervision, teacher-forced (L_1) regression often *averages* across valid behaviors rather than selecting a single executable mode, producing smooth but practically *invalid* actions such as collision-prone paths or inaccurate end-effector alignment.

Generative policies such as diffusion-based methods offer a powerful alternative to deterministic regression by representing multiple plausible action modes. However, diffusion-transformer approaches such as AC-DiT [4] still face both *generation-level* and *execution-level* challenges in mobile manipulation. At the generation level, samples from pure noise may be diverse but weakly constrained around feasible expert-like whole-body motion patterns, while iterative sampling increases inference cost. At the execution level, action chunking improves efficiency but leaves future actions partially open-loop, allowing small perception or actuation errors to compound into drift [9, 17]. While receding-horizon execution and reactive updates can mitigate such drift [6, 33, 23, 30, 34], adapting large vision-language backbones for such behavior remains computationally expensive and brittle across embodiments.

These observations motivate a central question: can we design a VLA policy for mobile manipulation that is ① computationally efficient, ② multimodal, and ③ structured

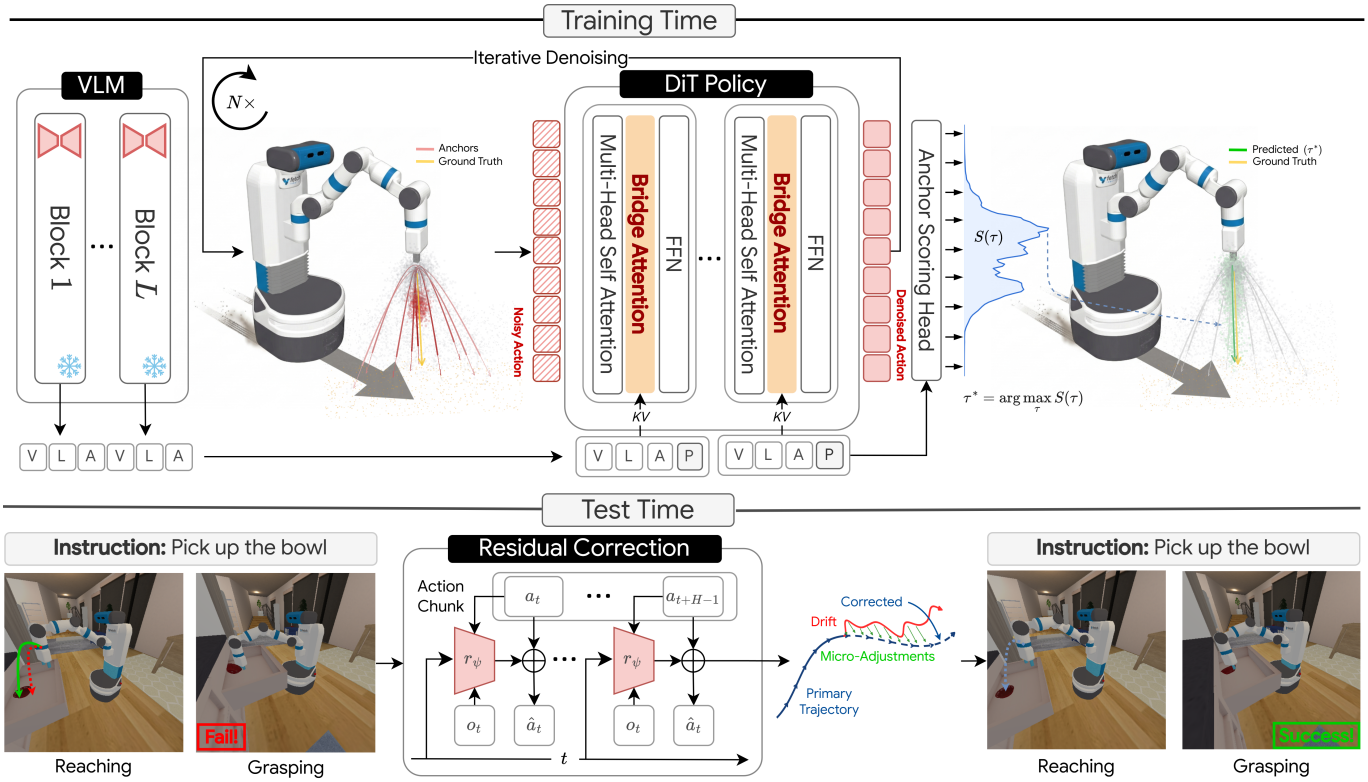


Fig. 3: **Overview of the MobiGuide framework. (Top) Training**: The VLA backbone encodes multimodal context. The action head refines demonstration-derived trajectory anchors into context-conditioned proposals, and a scoring head selects the proposal closest to the expert action chunk. **(Bottom) Inference**: During chunked execution, residual correction predicts state-dependent micro-adjustments to compensate for base drift and arm–object misalignment.

enough to remain executable under chunk-induced drift?

We answer this question with **MobiGuide**, a generative VLA policy that stabilizes long-horizon mobile manipulation through guided trajectory generation and lightweight test-time refinement. The key idea is to construct a vocabulary of *trajectory anchors* from expert demonstrations, where each anchor represents a typical whole-body motion pattern and provides a structured prior over plausible behaviors. Given the current observation and instruction, MobiGuide locally refines these anchors into context-compatible proposals and scores them for execution. This preserves multimodal candidates while constraining generation around demonstration-induced motion modes. To improve robustness during chunked execution, we introduce a lightweight residual correction module that performs per-step adjustments during rollout without expensive re-planning.

We validate MobiGuide in large-scale simulation and real-world deployment. In simulation, we evaluate on ManiSkill-HAB across six pick, place, open, and close tasks; in the real world, we deploy on a Unitree Go2 Edu quadruped with an SO101 arm, where legged-base vibration and drift stress-test execution-time correction. Empirically, MobiGuide achieves a state-of-the-art 64.0% average success rate on ManiSkill-HAB, outperforming leading 3D-aware and RGB baselines by 8.4% and 21.1%, respectively. It remains robust under longer action chunks, maintaining a 61.5% success rate and rivaling computationally heavy reactive baselines. These results show that MobiGuide improves execution stability while preserving the low-latency inference needed for real-world deployment.

II. Methodology

A. Problem Formulation

We consider language-conditioned mobile manipulation, where a robot with a mobile base and a manipulator must execute long-horizon tasks in unstructured environments. At timestep t, the robot observes $o_t = \{I_t^g, I_t^w, s_t\}$, where I_t^g and I_t^w are global- and wrist-view RGB images, and s_t denotes the proprioceptive state of the base, arm, end-effector, and gripper. Given a language instruction ℓ, the policy predicts a future action chunk $A_t = a_{t:t+H-1} \in \mathbb{R}^{H\times d}$, where H is the action horizon and each action $a_t \in \mathbb{R}^d$ jointly controls the mobile base and the manipulator.

B. Framework Overview

Fig. 3 presents an overview of MobiGuide. The framework addresses the trade-off between efficient deterministic regression and multimodal generative modeling. While L_1 action heads are simple and low-latency, they can suffer from *mode averaging* when multiple valid action chunks exist. Generative policies better capture such multimodality, but sampling from broad noise priors can produce *weakly constrained proposals*

and increase inference cost. MOBIGUIDE resolves this through **vocabulary-guided local refinement**: instead of regressing a single action chunk or generating from pure noise, it refines demonstration-derived trajectory anchors into context-compatible proposals and scores them for execution.

Concretely, MOBIGUIDE contains three components. First, a lightweight context encoder combines VLM features from the visual-language inputs with proprioceptive state embeddings:

$$x_t = \Phi_{\text{ctx}}\left(f_{\text{VLM}}(I_t^g, I_t^w, \ell), f_{\text{state}}(s_t)\right), \tag{1}$$

where x_t summarizes the language-conditioned observations and state. Second, the anchor-guided action head conditions on x_t to refine and score multiple trajectory anchors. Third, during rollout, a lightweight residual correction module applies step-wise action adjustments to reduce execution drift during chunked control.

C. Trajectory Anchors as Structured Action Priors

The central idea of MOBIGUIDE is to represent expert demonstrations using a vocabulary of trajectory anchors $\mathcal{V} = \{\bar{A}^{(m)}\}_{m=1}^{M}$, where $\bar{A}^{(m)} \in \mathbb{R}^{H\times d}$. Each anchor $\bar{A}^{(m)}$ corresponds to a representative action-chunk pattern, such as an approach route, base motion, or grasp trajectory. In practice, we segment expert demonstrations into fixed-length action chunks and cluster them in action-chunk space to obtain M representative anchors.

These anchors provide explicit hypotheses over plausible behavior modes. Conceptually, the conditional action distribution can be viewed as an anchor-conditioned mixture:

$$p(A_t \mid x_t) \approx \sum_{m=1}^{M} p_\theta(m \mid x_t)\, p_\theta(A_t \mid x_t, \bar{A}^{(m)}). \tag{2}$$

Here, $p_\theta(m \mid x_t)$ captures which trajectory mode is suitable for the current observation and instruction, while $p_\theta(A_t \mid x_t, \bar{A}^{(m)})$ models local adaptation around that anchor. This separates mode representation from local action prediction: anchors enumerate plausible modes, while the refinement head adapts them to the current context.

D. Anchor-Guided Local Refinement

Given the context representation x_t and the anchor vocabulary $\mathcal{V}$, the action head refines each anchor locally instead of generating an action chunk from pure noise. Let S denote the full number of refinement levels, and let $\rho \in (0, 1]$ be the truncation ratio. We define the truncated refinement horizon as $S_{\text{tr}} = \max\{1, \lfloor \rho S \rceil\}$ and use low-noise refinement levels $\tau \in \{0, \dots, S_{\text{tr}} - 1\}$.

For each anchor $\bar{A}^{(m)}$, we construct an anchor-local perturbed trajectory at refinement level τ:

$$A_t^{(\tau,m)} = \sqrt{\bar{\alpha}_\tau}\bar{A}^{(m)} + \sqrt{1-\bar{\alpha}_\tau}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

The schedule controls the perturbation radius: larger τ explores a broader anchor-local neighborhood, while smaller τ performs fine-grained correction near the anchor. The refinement level is encoded by a timestep embedding $e_\tau = \phi_{\text{time}}(\tau)$ and the perturbed trajectory is processed by a DiT-style action transformer:

$$z_t^{(\tau,m)} = T_\theta\left(A_t^{(\tau,m)}, x_t, e_\tau, \bar{A}^{(m)}\right). \tag{4}$$

From this representation, a refinement head updates the proposal via a residual

$$\hat{A}_t^{(\tau,m)} = A_t^{(\tau,m)} + \Delta A_t^{(\tau,m)}, \qquad \Delta A_t^{(\tau,m)} = r_\theta\left(z_t^{(\tau,m)}\right). \tag{5}$$

A separate scoring head predicts the compatibility score $s^{(\tau,m)} = h_\theta\left(z_t^{(\tau,m)}\right)$. Here, $\hat{A}_t^{(\tau,m)}$ denotes the locally refined action proposal from anchor m at refinement level τ, and $s^{(\tau,m)}$ measures its compatibility with the current context.

At inference time, all anchors are evaluated in parallel. For each anchor, we initialize the trajectory at the largest truncated refinement level, $\tau_{\text{start}} = S_{\text{tr}} - 1$, using Eq. (3). Starting from τ_{start}, we iteratively apply the refinement update toward $\tau = 0$:

$$A_t^{(\tau-1,m)} = \mathcal{U}_\tau\left(A_t^{(\tau,m)}, \hat{A}_t^{(\tau,m)}\right), \qquad \tau = \tau_{\text{start}}, \dots, 1, \tag{6}$$

where $\mathcal{U}_\tau(\cdot)$ denotes the truncated reverse update from level τ to $\tau - 1$.

After the final refinement step, each anchor yields a refined proposal and compatibility score. MOBIGUIDE then selects the highest-scoring anchor-conditioned proposal as the final macro action chunk; see Appendix Fig. 7. Unlike full diffusion-style generation from pure noise, this truncated refinement starts from anchor-local perturbations, constraining the search to plausible expert-like trajectory modes.

E. Test-time Residual Correction for Chunked Execution

After selecting the macro action chunk $\hat{A}_t = \{\hat{a}_t, \dots, \hat{a}_{t+H-1}\}$, the robot executes it over H control steps. Although the chunk provides a temporally coherent trajectory, open-loop execution can drift, especially when small base deviations alter the manipulator–object geometry. To mitigate this drift without querying the full VLA policy at every control step, we introduce a lightweight residual correction module. At execution phase $j \in \{0, \dots, H-1\}$, the applied action is $a_{t+j} = \hat{a}_{t+j} + \Delta a_{t+j}$, where $\Delta a_{t+j} = r_\psi(o_{t+j}, \ell, \hat{a}_{t+j}, j)$ corrects the macro action using the current observation.

F. Learning Objective

During training, each expert action chunk $A_t^\star$ is assigned to the closest anchor under a trajectory-level distance metric: $m^\star = \arg\min_m d\left(A_t^\star, \bar{A}^{(m)}\right)$, where $d(\cdot,\cdot)$ measures the distance between two action chunks. The corresponding anchor label is defined as $y_m = \mathbb{1}[m = m^\star]$. We jointly train the refinement and scoring heads with:

$$\mathcal{L} = \sum_{m=1}^{M}\left[y_m\, \mathcal{L}_{\text{ref}}\left(\hat{A}_t^{(m)}, A_t^\star\right) + \lambda\, \mathcal{L}_{\text{score}}\left(s^{(m)}, y_m\right)\right]. \tag{7}$$

where $\mathcal{L}_{\text{ref}}$ is a trajectory reconstruction loss over the action chunk, and $\mathcal{L}_{\text{score}}$ is a binary classification loss for anchor compatibility.

TABLE I: **Simulation results on ManiSkill-HAB mobile manipulation benchmark.** We report task success rate (%, mean ± std) across six tasks spanning PICK, PLACE, OPEN, and CLOSE behaviors. Columns indicate object–target or articulated-object pairs.

Method	Venue	Input	PICK		PLACE		OPEN	CLOSE	Average
			Apple Fridge	Bowl Counter	Apple Table	Bowl Table	Door Fridge	Drawer Counter	
π_0	arXiv'24	RGB	13.0 ± 1.6	15.7 ± 1.2	23.3 ± 1.7	21.3 ± 2.6	31.3 ± 2.6	70.0 ± 2.2	33.5
ACT	RSS'23	RGB	28.0 ± 2.2	28.0 ± 2.4	8.7 ± 3.3	13.0 ± 0.8	2.0 ± 2.2	85.7 ± 1.2	23.6
RDT	ICLR'25	RGB	12.0 ± 11.3	10.7 ± 6.8	32.0 ± 5.7	18.7 ± 5.0	82.7 ± 10.5	100.0 ± 0.0	42.9
DP	RSS'23	RGB	21.3 ± 3.3	20.7 ± 3.3	28.0 ± 8.0	69.3 ± 3.3	7.3 ± 5.8	55.0 ± 5.7	28.8
DP3	RSS'24	PointCloud	0.0 ± 0.0	20.0 ± 2.4	31.0 ± 0.8	32.0 ± 0.8	0.0 ± 0.0	68.0 ± 0.0	21.6
AC-DiT	NeurIPS'25	PointCloud	**33.3 ± 1.9**	36.0 ± 6.5	33.3 ± 9.4	17.3 ± 6.8	**90.7 ± 5.0**	97.3 ± 1.9	55.6
MOBIGUIDE$_{H=2}$	Ours	RGB	22.7 ± 0.9	**44.5 ± 0.8**	**64.3 ± 0.8**	63.8 ± 2.0	88.9 ± 0.8	**100.0 ± 0.0**	**64.0**
MOBIGUIDE$_{H=5}$	Ours	RGB	16.1 ± 0.5	42.9 ± 0.8	57.7 ± 0.9	**69.6 ± 0.9**	82.5 ± 0.7	**100.0 ± 0.0**	61.5

III. EXPERIMENTS

A. Experimental Setup

We evaluate MOBIGUIDE in ManiSkill-HAB [21], built on SAPIEN [28], using six SetTable mobile manipulation tasks: PICK APPLE, PLACE APPLE, PICK BOWL, PLACE BOWL, OPEN FRIDGE, and CLOSE KITCHEN COUNTER. We further deploy the policy on a Unitree Go2 Edu quadruped equipped with an SO101 arm for two real-world tasks, OPEN DRAWER and LIFT BAG. MOBIGUIDE uses Qwen-0.5B [20] as the VLM backbone and is trained from 1,000 successful expert rollouts per task. The policy takes two RGB views and proprioceptive state as input and predicts a unified 13-DoF continuous base-arm action. We report task success rate as the primary metric. More details on real-world setup can be found in the Appendix–Fig. 4.

B. Simulation Results

Table I reports success rates on ManiSkill-HAB. Despite using only RGB observations, MOBIGUIDE$_{H=2}$ achieves the best average success rate of 64.0%, outperforming the strongest PointCloud baseline, AC-DiT, by 8.4 points. The gains are especially large on long-horizon placement tasks, where multimodal base-arm coordination is critical. With a longer execution horizon of $H = 5$, MOBIGUIDE maintains a strong average success rate of 61.5%, suggesting that anchor-conditioned trajectory reconstruction produces stable multi-step action chunks under less frequent replanning.

C. Real-World Results

Table II compares MOBIGUIDE with ACT, Diffusion Policy, and SmolVLA on the real robot. Each task is evaluated over 10 randomized trials. MOBIGUIDE achieves a 60% average success rate, compared with 5%, 10%, and 15% for ACT, Diffusion Policy, and SmolVLA, respectively, demonstrating improved robustness under real-world base vibration and execution drift.

D. Ablation Study

We ablate the two main components on PICK BOWL. Removing trajectory anchors and initializing refinement from Gaussian noise substantially degrades performance under low denoising budgets: with 10 refinement steps, the unanchored model obtains 0.0% success, while the anchored model reaches 42.9%. Removing residual correction reduces success from 42.9% to 33.6%, confirming that lightweight per-step correction mitigates chunk-induced drift. Finally, increasing the execution horizon from $H = 1$ to $H = 5$ reduces per-episode compute from 641.4 TFLOPs to 128.3 TFLOPs, an approximately 80% reduction, while MOBIGUIDE maintains 61.5% average success on ManiSkill-HAB. Detailed ablations and qualitative visualizations are provided in the Appendix E.

TABLE II: Success rate comparison between MOBIGUIDE and baselines on real-world tasks.

Method	Open Drawer	Lift Bag	Average
ACT	10%	0%	5%
DIFFUSION POLICY	20%	0%	10%
SMOLVLA	20%	10%	15%
MOBIGUIDE	**70%**	**50%**	**60%**

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduce MOBIGUIDE, an efficient generative framework for dynamic mobile manipulation. MOBIGUIDE uses anchor-guided diffusion to capture the multimodal structure of expert demonstrations while satisfying the real-time constraints of whole-body control. By refining demonstration-derived anchors with a truncated denoising schedule, it achieves low-latency execution competitive with deterministic policies. A lightweight scoring head further improves reliability by selecting context-compatible anchors, and a residual correction module mitigates open-loop drift during long-horizon execution. As a result, MOBIGUIDE offers a favorable compute-performance trade-off, scaling to longer action chunks without the mode-averaging failures of standard regression. Our current formulation relies on offline imitation learning, so robustness remains bounded by the coverage and diversity of the demonstration dataset. A promising direction for future work is to incorporate reinforcement learning for stronger test-time self-correction and recovery in out-of-distribution settings.

References

[1] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith LLontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734, 2025. doi: 10.48550/ARXIV.2503.14734. URL https://doi.org/10.48550/arXiv.2503.14734.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024. doi: 10.48550/ARXIV.2410.24164. URL https://doi.org/10.48550/arXiv.2410.24164.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[4] Sixiang Chen, Jiaming Liu, Siyuan Qian, Han Jiang, Lily Li, Renrui Zhang, Zhuoyang Liu, Chenyang Gu, Chengkai Hou, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Ac-dit: Adaptive coordination diffusion transformer for mobile manipulation. *CoRR*, abs/2507.01961, 2025. doi: 10.48550/ARXIV.2507.01961. URL https://doi.org/10.48550/arXiv.2507.01961.

[5] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, Yang Tian, Bin Wang, Bolun Wang, Fangjing Wang, Hanqing Wang, Tai Wang, Ziqin Wang, Xueyuan Wei, Chao Wu, Shuai Yang, Jinhui Ye, Junqiu Yu, Jia Zeng, Jingjing Zhang, Jinyu Zhang, Shi Zhang, Feng Zheng, Bowen Zhou, and Yangkun Zhu. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy, 2025. URL https://arxiv.org/abs/2510.13778.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://doi.org/10.15607/RSS.2023.XIX.026.

[7] Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, Han Zhao, Siteng Huang, and Donglin Wang. Openhelix: A short survey, empirical analysis, and open-source dual-system VLA model for robotic manipulation. *CoRR*, abs/2505.03912, 2025. doi: 10.48550/ARXIV.2505.03912. URL https://doi.org/10.48550/arXiv.2505.03912.

[8] Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. Mergevla: Cross-skill model merging toward a generalist vision-language-action agent. *arXiv preprint arXiv:2511.18810*, 2025.

[9] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024. URL https://arxiv.org/abs/2401.02117.

[10] 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. *CoRR*, abs/2504.16054, 2025. doi: 10.48550/ARXIV.2504.16054. URL https://doi.org/10.48550/arXiv.2504.16054.

[11] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

[12] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *CoRR*, abs/2502.19645, 2025. doi: 10.48550/ARXIV.2502.19645. URL https://doi.org/10.48550/arXiv.2502.19645.

[13] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and José M. Álvarez. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *CoRR*, abs/2406.06978, 2024. doi: 10.48550/ARXIV.2406.06978. URL https://doi.org/10.48550/arXiv.2406.06978.

[14] Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zux-

uan Wu, and José M. Álvarez. Hydra-next: Robust closed-loop driving with open-loop training. *CoRR*, abs/2503.12030, 2025. doi: 10.48550/ARXIV.2503.12030. URL https://doi.org/10.48550/arXiv.2503.12030.

[15] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and José M. Álvarez. Generalized trajectory scoring for end-to-end multimodal planning. *CoRR*, abs/2506.06664, 2025. doi: 10.48550/ARXIV.2506.06664. URL https://doi.org/10.48550/arXiv.2506.06664.

[16] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 12037–12047. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01124. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liao_DiffusionDrive_Truncated_Diffusion_Model_for_End-to-End_Autonomous_Driving_CVPR_2025_paper.html.

[17] Min Lin, Xiwen Liang, Bingqian Lin, Liu Jingzhi, Zijian Jiao, Kehan Li, Yuhan Ma, Yuecheng Liu, Shen Zhao, Yuzheng Zhuang, and Xiaodan Liang. Echovla: Robotic vision-language-action model with synergistic declarative memory for mobile manipulation. *CoRR*, abs/2511.18112, 2025. doi: 10.48550/ARXIV.2511.18112. URL https://doi.org/10.48550/arXiv.2511.18112.

[18] Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment, 2025. URL https://arxiv.org/abs/2511.04555.

[19] 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. *arXiv preprint arXiv:2410.07864*, 2024.

[20] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[21] Arth Shukla, Stone Tao, and Hao Su. Maniskillhab: A benchmark for low-level manipulation in home rearrangement tasks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6bKEWevgSd.

[22] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andrés Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Rémi Cadène. Smolvla: A vision-language-action model for affordable and efficient robotics. *CoRR*, abs/2506.01844, 2025. doi: 10.48550/ARXIV.2506.01844. URL https://doi.org/10.48550/arXiv.2506.01844.

[23] Junhyuk So, Chiwoong Lee, Shinyoung Lee, Jungseul Ok, and Eunhyeok Park. Improving generative behavior cloning via self-guidance and adaptive chunking. *arXiv preprint arXiv:2510.12392*, 2025.

[24] starVLA Contributors. Starvla: A lego-like codebase for vision-language-action model developing. GitHub repository, 1 2025. URL https://github.com/starVLA/starVLA.

[25] Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. *CoRR*, abs/2509.09372, 2025. doi: 10.48550/ARXIV.2509.09372. URL https://doi.org/10.48550/arXiv.2509.09372.

[26] Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic vla foundation model, 2026. URL https://arxiv.org/abs/2601.18692.

[27] Zhenyu Wu, Yuheng Zhou, Xiuwei Xu, Ziwei Wang, and Haibin Yan. Momanipvla: Transferring vision-language-action models for general mobile manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 1714–1723. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00167. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_MoManipVLA_Transferring_Vision-language-action_Models_for_General_Mobile_Manipulation_CVPR_2025_paper.html.

[28] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11094–11104. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01111. URL https://openaccess.thecvf.com/content_CVPR_

2020/html/Xiang_SAPIEN_A_SimulAted_Part-Based_Interactive_ENvironment_CVPR_2020_paper.html.

[29] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 1602–1611. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00157. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xing_GoalFlow_Goal-Driven_Flow_Matching_for_Multimodal_Trajectories_Generation_in_End-to-End_CVPR_2025_paper.html.

[30] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. *arXiv preprint arXiv:2503.02881*, 2025.

[31] Sixu Yan, Zeyu Zhang, Muzhi Han, Zaijin Wang, Qi Xie, Zhitian Li, Zhehan Li, Hangxin Liu, Xinggang Wang, and Song-Chun Zhu. M2diffuser: Diffusion-based trajectory optimization for mobile manipulation in 3d scenes. *CoRR*, abs/2410.11402, 2024. doi: 10.48550/ARXIV.2410.11402. URL https://doi.org/10.48550/arXiv.2410.11402.

[32] Jingyun Yang, Isabella Huang, Brandon Vu, Max Bajracharya, Rika Antonova, and Jeannette Bohg. Mobi-π: Mobilizing your robot learning policy. *CoRR*, abs/2505.23692, 2025. doi: 10.48550/ARXIV.2505.23692. URL https://doi.org/10.48550/arXiv.2505.23692.

[33] Xiu Yuan. Unpacking the individual components of diffusion policy. *arXiv preprint arXiv:2412.00084*, 2024.

[34] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. *arXiv preprint arXiv:2412.13630*, 2024.

[35] 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, 2025. URL https://arxiv.org/abs/2503.11081.

[36] Ziyan Zhao, Ke Fan, He-Yang Xu, Ning Qiao, Bo Peng, Wenlong Gao, Dongjiang Li, and Hui Shen. Anchordp3: 3d affordance guided sparse diffusion policy for robotic manipulation, 2025. URL https://arxiv.org/abs/2506.19269.

[37] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

[38] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10838–10845, 2025. doi: 10.1109/ICRA55743.2025.11128074.

[39] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

APPENDIX

We provide supplementary materials to support the main paper. Appendix A includes an extended discussion of related work. Appendix B provides additional details on model training. Appendix C describes further implementation details of the model architecture. Appendix D presents the full algorithmic procedure. Appendix E reports additional ablation studies. Appendix F illustrates qualitative examples of our method.

A. Related Work

Recent vision-language-action (VLA) models adapt large-scale vision-language backbones for robotic control, either by representing actions as token sequences [3, 39, 11] or by attaching compact action heads to frozen or partially frozen backbones [25, 22, 1, 2, 24, 5, 18]. These approaches improve semantic generalization and parameter efficiency, but long-horizon mobile manipulation additionally requires coordinated prediction of base and arm motions under multimodal task constraints. Existing mobile manipulation systems often address this coordination through structured decomposition, such as base waypoint prediction, viewpoint optimization, or hierarchical planning [27, 32]. Other methods learn whole-body trajectory distributions more directly, including scene-conditioned trajectory optimization in M2Diffuser [31] and diffusion-style whole-body action modeling in AC-DiT [4]. In parallel, diffusion policies have been used to capture multimodal action distributions in robotic manipulation and planning [19, 38], while some trajectory priors have been explored for efficient refinement in autonomous driving and manipulation [16, 29, 15, 14, 13, 36]. Our work differs from these lines by formulating mobile manipulation as guided dense trajectory reconstruction: the policy reconstructs executable base-arm action chunks guided by trajectory vocabularies and further improves closed-loop execution through residual self-correction, reducing the need for prolonged iterative trajectory generation.

Vision-Language-Action (VLA) Models. The paradigm of robot learning has shifted from training specialized policies from scratch to adapting large-scale pre-trained vision-language models (VLMs) for control. Early monolithic VLA models, such as RT-1 [3]/RT-2 [39] and OpenVLA [11], treated action tokens as an extension of language, fine-tuning billion-parameter backbones on robot trajectories. While effective for semantic generalization, this approach suffers from prohibitive computational costs and high inference latency, often running at frequencies insufficient for dynamic mobile manipulation. Recent works have pivoted towards parameter-efficient architectures that bridge frozen VLMs with lightweight policy heads [25, 22, 1, 2, 24, 5, 18]. Yet, many lightweight heads rely on deterministic regression objectives, which can collapse multimodal action distributions into non-executable mean trajectories via mode-averaging when faced with complex, multi-step trajectory planning.

Mobile Manipulation. Mobile manipulation introduces the unique challenge of coordinating a non-holonomic mobile base with a high-degree-of-freedom manipulator. Naive concatenation of action spaces often leads to drift and error accumulation due to the disparate dynamic scales of the base and arm. MoManipVLA [27] addresses this via a bi-level optimization framework, where the VLA predicts sparse waypoints for the base to maximize workspace feasibility, while a lower-level planner solves for the arm trajectory. While effective for navigation-heavy tasks, this hierarchical separation can limit the reactiveness required for simultaneous whole-body interactions. Mobile-π [32] improves mobile manipulation by optimizing base placement via geometry-aware viewpoint selection while reusing a fixed arm policy. This decoupled mobilization mitigates viewpoint shift but does not learn unified visuomotor coordination, limiting simultaneous base–arm coupling in contact-rich interactions. Closer to our end-to-end objective is AC-DiT [4], our primary state-of-the-art baseline. AC-DiT achieves effective coordination through Mobility-to-Body Conditioning and Perception-Aware Multimodal Adaptation, which dynamically adjusts input weighting during navigation and grasping. However, its iterative diffusion-style action generation introduces substantial inference overhead for real-time execution. In contrast, our method reconstructs executable dense base–arm action chunks from task-conditioned anchors and further refines closed-loop behavior through residual self-correction.

Diffusion Policies and Anchored Planning. Diffusion models have become a powerful framework for representing multimodal robotic policies. Recent advancements have demonstrated the scaling potential of Diffusion Transformer (DiT) architectures in this domain. For instance, models such as RDT-1B [19] and ScaleDP [38] leverage transformer backbones to scale to large policy models, enabling the capture of complex, multimodal action distributions necessary for bimanual and whole-body behaviors. Despite their expressivity, standard diffusion policies require iterative denoising at inference time, which can limit their responsiveness in closed-loop control.

Structured trajectory priors provide an alternative way to improve generation efficiency. DiffusionDrive [16] explores anchor-initialized trajectory refinement in autonomous driving, where candidate motion anchors, such as lane-following primitives, provide strong initial trajectories for efficient denoising. In manipulation, AnchorDP3 [36] uses sparse keypose anchors to guide policy prediction. M2Diffuser [31] is also related as a mobile manipulation method that models scene-conditioned whole-body trajectory generation. These works show that structured priors can help constrain large trajectory spaces. Our method follows this broader principle, but differs in its formulation: rather than emphasizing stochastic generation from noise, we reconstruct dense executable base–arm trajectories from task-conditioned anchors and refine them through residual self-correction for closed-loop mobile manipulation.

B. Experimental Details

We summarize all fine-tuning hyperparameters used in our experiments in Table III. Unless otherwise specified, all experiments use the same VLA backbone and training configuration.

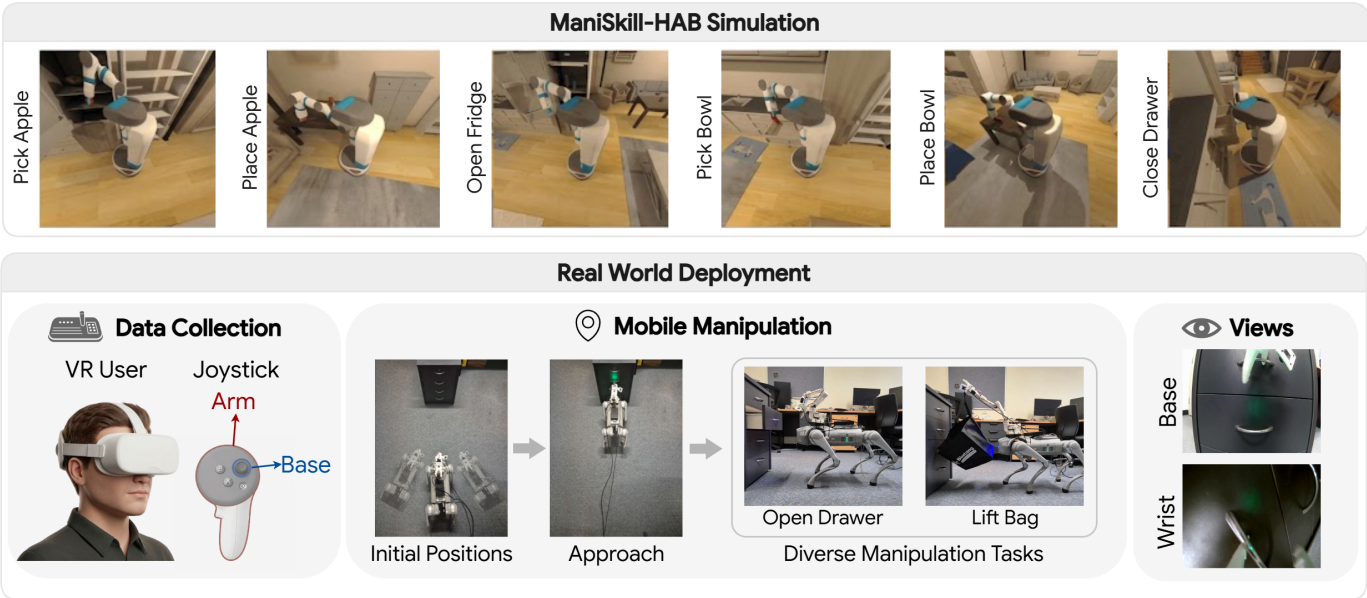


Fig. 4: Experimental Environment. Simulation experiments are conducted in ManiSkill-HAB. Real-world experiments are evaluated on the Unitree Go2 quadruped equipped with an SO101 arm. Each task requires navigation to the target followed by manipulation. The perception module includes two camera views, *i.e.,* base view from Go2 and wrist view from SO101. The data collection process was conducted by teleoperation with the Meta Quest 3 VR device and 50 demonstrations were collected for each task.

TABLE III: Fine-tuning hyperparameters used in all experiments.

Hyperparameter	Value
Backbone	Qwen-2.5 (0.5B)
Batch size	16
Learning rate	2×10^{-4}
LoRA rank	64
Use proprioception	True
Number of images	2
Gradient steps	100k

C. Further Implementation Details

1) Trajectory Vocabulary Construction: To construct the trajectory vocabulary $\mathcal{V} = \{\bar{A}^{(m)}\}_{m=1}^{M}$ from the expert demonstrations, we apply standard K-Means clustering offline on the segmented ground-truth action chunks. In our primary experiments, we set the number of clusters (anchors) to $M = 20$. Before clustering, the unified 13-DoF base and arm actions are normalized to $[-1, 1]$ to prevent any single dimension's dynamic scale from dominating the L_1 distance metric. Using $M = 20$ clusters provides sufficient coverage of the expert manifold, ensuring all ground-truth chunks fall within an easily denoisable distance of an anchor for the truncated diffusion schedule ($S_{tr} = 10$) to successfully converge.

2) Residual Correction Module Training: The residual correction module $r_\psi(\cdot)$ is implemented as a lightweight MLP to avoid bottle-necking the low-latency inference required for test-time self-correction.

Inputs and Execution History: The module takes the real-time observation o_{t+j}, the language instruction ℓ, the primary intended macro-action $\hat{a}_{t+j}$, and the intra-chunk step index j. To preserve the strict low-latency requirements of the system, the module is not conditioned on a long execution history; instead, it relies entirely on the **instantaneous** state mismatch to predict its adjustment.

Phase Encoding: The relative execution phase $j \in \{0, \ldots, H-1\}$ is encoded using a standard 1D sinusoidal positional embedding before being concatenated with the visual and action features and passed through the MLP layers.

Training Objective: To cleanly separate the high-level multimodal planning capabilities of the generative DiT from the high-frequency reactive corrections, we train the residual MLP **separately** after the primary MobiGuide backbone and diffusion head have been fully trained. During this stage, the residual module is conditioned on the nominal action chunk $\hat{a}_{t+j}$ predicted by the frozen MobiGuide policy and predicts the corrective micro-adjustment Δa_{t+j} needed to recover the expert trajectory. We optimize this module using the deterministic L_1 objective

$$\mathcal{L}_{\text{res}} = \mathbb{E}\left[||r_\psi(o_{t+j}, \ell, \hat{a}_{t+j}, j) - \Delta a_{t+j}||_1\right].$$

D. Algorithm Details

In this section, we provide a detailed algorithmic description in Algorithm 1 of how MobiGuide performs test-time inference via anchor-based scoring and truncated diffusion.

Algorithm 1 Test-time Anchored Diffusion and Residual Correction in MobiGuide

Inputs: Observation $o_t = \{I_t^g, I_t^w, s_t\}$, Instruction ℓ
Modules: Anchor vocabulary $\mathcal{V}$, Fine-tuned VLA f_{VLA}, Denoiser f_ϕ, Score head h, Residual corrector r_ψ, Diffusion schedule $\{\alpha_\tau, \bar{\alpha}_\tau\}$
Output: Corrected action chunk $\{a_t, \ldots, a_{t+H-1}\}$
Context extraction:
$x_t \leftarrow f_{\text{VLA}}(o_t, \ell)$ ▷ Extract multimodal latent context
Anchor initialization and scoring:
for $m = 1$ to M **do**
  Sample $\epsilon^{(m)} \sim \mathcal{N}(0, I)$
  $A_t^{(\tau_{\text{start}},m)} \leftarrow \sqrt{\bar{\alpha}_{\tau_{\text{start}}}}\, \bar{A}^{(m)} + \sqrt{1 - \bar{\alpha}_{\tau_{\text{start}}}}\, \epsilon^{(m)}$ ▷ Initialize from anchor neighborhood
  $\tilde{A}^{(m)} \leftarrow A_t^{(\tau_{\text{start}},m)}$
  for $\tau = \tau_{\text{start}}, \tau_{\text{start}} - 1, \ldots, 0$ **do**
    $\hat{\epsilon}^{(m)} \leftarrow f_\phi\Big(\tilde{A}^{(m)}, \text{emb}(\tau), x_t\Big)$
    $\tilde{A}^{(m)} \leftarrow g\Big(\tilde{A}^{(m)}, \hat{\epsilon}^{(m)}, \tau\Big)$ ▷ Reverse denoising update
  end for
  $\hat{A}_t^{(m)} \leftarrow \tilde{A}^{(m)}$
  $s^{(m)} \leftarrow h\Big(\hat{A}_t^{(m)}, x_t, \tau_{\text{start}}\Big)$ ▷ Predict anchor compatibility score
end for
Anchor selection:
$\hat{m} \leftarrow \arg\max_m s^{(m)}$ ▷ Choose best anchor-conditioned trajectory
$\hat{A}_t \leftarrow \tilde{A}_t^{(\hat{m})} = \{\hat{a}_t, \ldots, \hat{a}_{t+H-1}\}$
Chunk execution with residual correction:
for $j = 0$ to $H - 1$ **do**
  Observe current state o_{t+j}
  $\Delta a_{t+j} \leftarrow r_\psi(o_{t+j}, \ell, \hat{a}_{t+j}, j)$ ▷ Predict state-dependent micro-adjustment
  $a_{t+j} \leftarrow \hat{a}_{t+j} + \Delta a_{t+j}$
  Execute a_{t+j}
end for
return $\{a_t, \ldots, a_{t+H-1}\}$

TABLE IV: Anchored prior under different denoising budgets. Speedup is relative to w/o Anchor with 50 steps.

Method	Steps	SR (%) ↑	Freq. (Hz) ↑	Speed
w/o Anchor	50	12.8	60.6	1.00
w/o Anchor	25	5.6	84.5	×1.39
w/o Anchor	10	0.0	**108.0**	×1.78
w/ Anchor	10	**42.9**	53.6	×0.88
w/ Anchor	5	12.6	89.8	×1.48

E. Ablation Study

To isolate the specific contributions of our proposed architectural choices, we conduct targeted ablation studies within the ManiSkill-HAB benchmark suite, specifically focusing our evaluation on the complex Pick Bowl task.

Impact of Anchored Prior. We evaluate the anchored prior by comparing the full model against a standard Diffusion Transformer baseline ("w/o Anchor") initialized from Gaussian noise $\mathcal{N}(0, 1)$. The anchored prior provides two benefits. First, it improves robustness under truncated refinement: with a 10-step budget, the baseline obtains 0.0% success, while MobiGuide achieves 42.9%. Second, it improves inference efficiency without sacrificing quality. To match the baseline's best performance of 12.8% at 50 steps, MobiGuide requires only 5 steps, reaching 12.6% success and increasing inference frequency from 60.6 Hz to 89.8 Hz. These results show that structured trajectory anchors provide an effective initialization for low-latency dense action reconstruction.

TABLE V: Ablation of residual drift correction. Residual correction improves success with modest overhead.

Method	SR (%) ↑	Freq. (Hz) ↑	Params
w/o Residual	33.6	**60.1**	726.19M
w/ Residual	**42.9**	53.6	726.25M

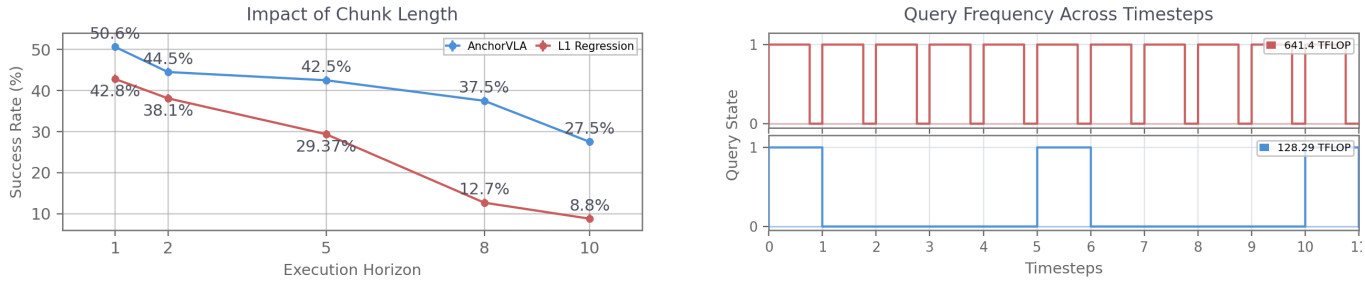
Impact of Residual Action Prediction. We evaluate the residual module by comparing the full model against an open-loop variant without residual correction. During chunked mobile manipulation, small base-tracking errors can accumulate and induce larger arm–object misalignments. Removing the residual module reduces success from 42.9% to 33.6%, showing that state-dependent micro-corrections improve execution robustness. This gain comes with negligible overhead: the module adds only 57K parameters to the 726M backbone and maintains a high inference frequency of 53.6 Hz, compared to 60.1 Hz without it. Thus, residual action prediction provides an efficient mechanism for translating reconstructed macro-trajectories into stable whole-body control.

Impact of Action Chunk Length (H). We evaluate MobiGuide against a deterministic L_1 regression baseline across action chunk lengths $H \in \{1, 2, 5, 8, 10\}$ in Fig. 5a. Longer chunks can capture richer temporal structure and reduce policy-query frequency, but they also increase open-loop drift under receding-horizon execution. The L_1 baseline is highly sensitive to this trade-off: as H increases, its performance drops sharply, consistent with mode-averaging over temporally extended multimodal actions. In contrast, MobiGuide degrades more gradually, maintaining a stronger compute–performance trade-off at longer horizons. This suggests that anchor-conditioned trajectory reconstruction better preserves diverse action hypotheses, enabling longer action chunks and improved inference efficiency without catastrophic performance collapse.

Inference Efficiency and Compute Savings. We further quantify the computational benefit of action chunking in Fig. 5b. Querying the VLA backbone at every timestep ($H = 1$) requires 641.4 **TFLOPs** per episode. Increasing the execution horizon to $H = 5$ reduces the number of policy queries and lowers the per-episode compute to 128.3 **TFLOPs**, corresponding to an approximate 80% reduction. Together with the stable performance observed at longer horizons, this shows that MobiGuide can substantially improve inference efficiency while maintaining competitive execution success.

F. Visualization

Fig. 6 visualizes real-time trajectory predictions of MobiGuide during continuous execution on Pick Bowl and



(a) Performance under varying execution horizons (H), comparing MobiGuide and a deterministic L_1 regression head.

(b) Query timing and per-episode compute under different execution horizons (H=1 vs. H=5).

Fig. 5: Effect of chunked execution horizon on robustness and efficiency. Increasing H reduces VLA query frequency and per-episode compute (e.g., from 641.4 **TFLOP** at H=1 to 128.29 **TFLOP** at H=5), but also makes action prediction harder. The deterministic L_1 regression baseline degrades sharply as chunk length increases (42.8% $\rightarrow$ 8.8%), whereas MobiGuide degrades more gradually (50.6% $\rightarrow$ 27.5%) and remains competitive at H=8 and H=10, indicating greater robustness to long-horizon chunked execution.

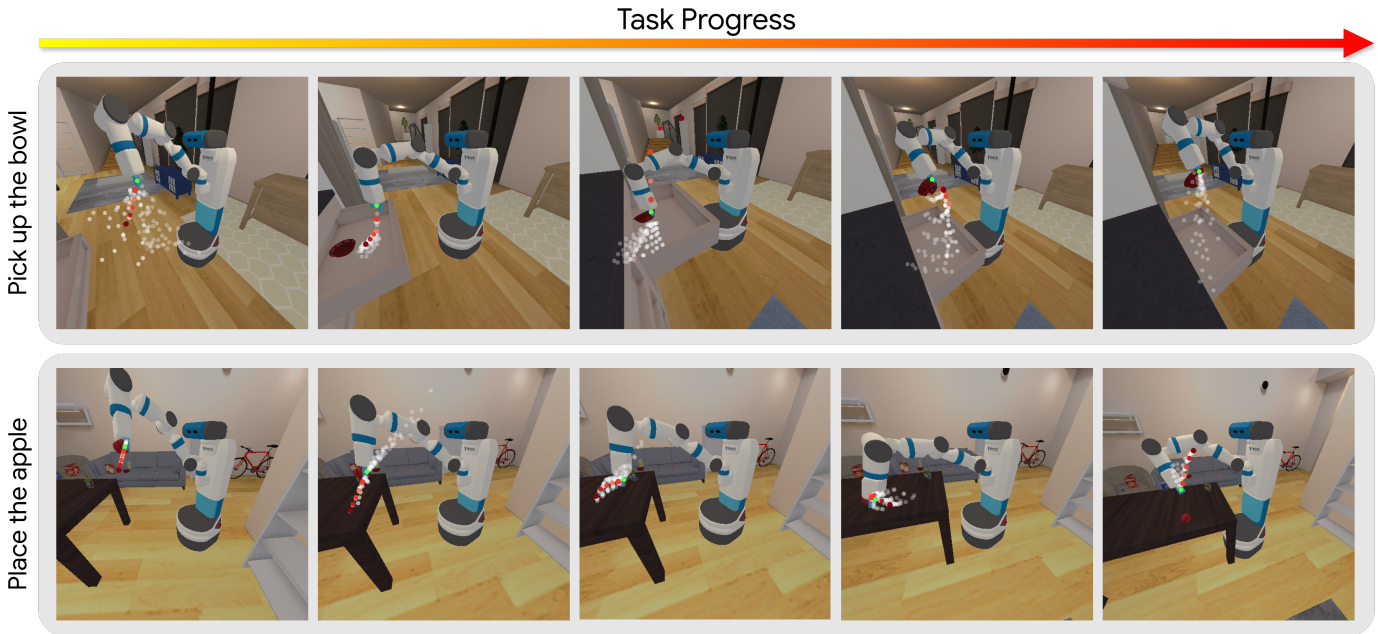


Fig. 6: **Qualitative visualization of MobiGuide in ManiSkill-HAB.** We show task progress from left to right for Pick Bowl and Place Apple. White points denote candidate future end-effector trajectories reconstructed from anchored proposals, while colored points indicate the trajectory selected by the scoring head for execution.

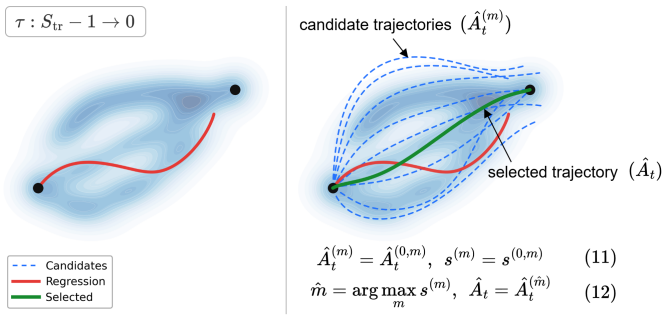


Fig. 7: Comparison of unimodal prediction and vocabulary-guided local refinement around candidate trajectories.

Place Apple. At each timestep, we project the predicted future end-effector poses into the robot's visual workspace. The white points show diverse candidate trajectories reconstructed from anchored proposals, while the colored points trace the trajectory selected by the scoring head for execution. Early in execution, the candidates cover multiple plausible approach directions; as the robot moves closer to the target, the proposals become more concentrated around fine-grained manipulation. This suggests that MobiGuide preserves multimodal action hypotheses while selecting temporally coherent trajectories for closed-loop mobile manipulation.