A Scalable Whole-body Trajectory Generator for Coordinated Mobile Manipulation

Yida Niu[1,2,3,4] Xinhai Chang[1,2,4,5] Xin Liu[4] Ziyuan Jiao[4] Yixin Zhu[1,2,3,6†]

***Abstract*—Mobile manipulators must seamlessly coordinate their base, arm, and manipulated objects to perform tasks effectively, particularly within confined household environments. While learning-based methods hold promise for discovering robust, generalizable control policies, they critically depend on access to large volumes of high-quality, physically valid training data. Generating such coordinated, whole-body trajectories requires simultaneously satisfying complex robot-scene constraints, yet existing datasets remain limited by the computational complexity of producing physically valid motions across diverse robots, environments, and task configurations. Here we present AutoMoMa, a system that efficiently generates high-quality whole-body trajectories using Virtual Kinematic Chain (VKC) modeling and GPU-accelerated motion planning at a rate of 2.5k valid episodes per hour per consumer-level GPU. Unlike prior approaches that rely on manual demonstrations or are tied to specific robot-scene pairs, AutoMoMa generalizes across diverse household layouts, interactive objects, robot morphologies, and manipulation tasks while ensuring physical feasibility and strict constraint satisfaction. This large-scale, diverse dataset establishes a foundation for advancing learning-based approaches to mobile manipulation in everyday environments. By making our code and dataset publicly available, we enable community-driven research toward autonomous robots that can reliably perform complex manipulation tasks in human-centered spaces. Website: https://automoma.pages.dev**

I. Introduction

Effective mobile manipulation in everyday environments demands seamless coordination between a robot's locomotion and manipulation capabilities. As robots navigate through homes—reaching into cabinets, manipulating furniture, or retrieving objects from cluttered surfaces—they must orchestrate their mobile base and articulated arm as a unified system while accounting for spatial constraints and task requirements; see Fig. 1. This whole-body coordination is fundamental to household service robots [21], requiring simultaneous reasoning about base positioning, arm configuration, and object interaction within dynamic and confined environments [25].

Learning-based approaches offer a promising path toward robust mobile manipulation policies, but they remain severely constrained by the absence of large-scale, diverse datasets capturing effective whole-body coordination. Current data collection methods face critical limitations: Reinforcement Learning (RL) requires prohibitively expensive trial-and-error exploration [10, 36, 24], especially when scaling across object variations and environments; teleoperation [11] is bottlenecked by expert availability and hardware interface limitations; and traditional planners [25, 38] struggle with real-world complexity without extensive task-specific tuning. These constraints have fragmented research efforts, forcing teams to develop narrow-purpose datasets [11, 26, 27, 4, 34] that fail to capture the full spectrum of mobile manipulation scenarios, ultimately impeding progress toward general-purpose household robots.

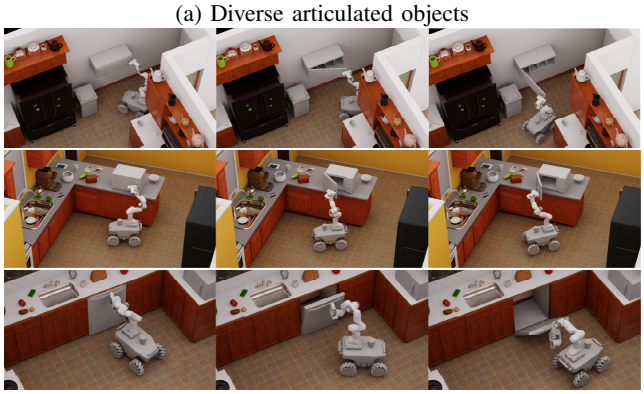
The challenge of generating comprehensive whole-body coordination data stems from the complex, high-dimensional nature of mobile manipulation. Effective datasets must simultaneously model the entangled interactions between base motion, arm articulation, and object manipulation across diverse robots, environments, and tasks. This complexity makes manual trajectory generation infeasible, while automated approaches must efficiently navigate vast configuration spaces subject to multiple interacting constraints. Consequently, existing datasets often oversimplify by focusing on static manipulation scenarios [6, 12], specific robot architectures [26, 1], or limited task classes [34]—failing to capture the rich coordination behaviors needed for generalizable mobile manipulation.

The VKC framework [18] offers a principled foundation to address these challenges by unifying the kinematics of base, arm, and manipulated objects into a cohesive representation. This integrated approach enables coordinated planning in a unified configuration space, naturally generalizing across different robots and scenes with known kinematic structures [19]. However, translating this theoretical advantage into practical data generation faces significant obstacles: conventional VKC implementations rely on computationally intensive optimization that prohibits large-scale dataset creation; they lack diverse, photo-realistic environments and object interactions; and they typically assume fixed grasping poses throughout task execution, limiting their efficacy in confined spaces where switching grasp poses in a task is required to complete operations such as fully opening a dishwasher door.

We present AutoMoMa, a scalable trajectory generator for diverse, high-quality whole-body mobile manipulation trajectories that overcomes these limitations. By integrating VKC-based modeling with GPU-accelerated motion planning [32], AutoMoMa generates physically valid whole-body coordination trajectories at a rate of 2.5k valid episodes per hour per consumer-level GPU, orders of magnitude faster than previous

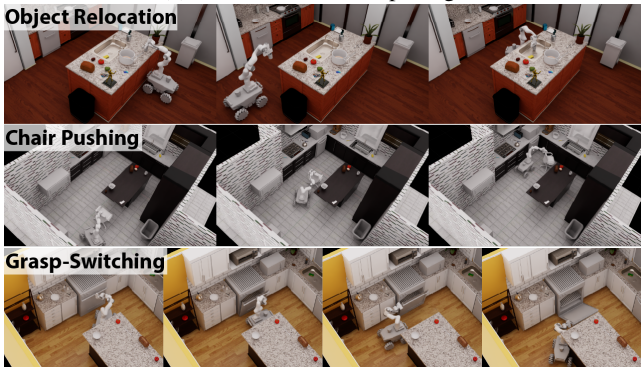
[1] Institute for Artificial Intelligence, Peking University.
[2] School of Psychological and Cognitive Sciences, Peking University.
[3] Beijing Key Laboratory of Behavior and Mental Health, Peking University.
[4] State Key Lab of General AI, Beijing Institute for General AI.
[5] Yuanpei College, Peking University.
[6] Embodied Intelligence Lab, PKU-Wuhan Institute for Artificial Intelligence.
† Corresponding author.
Emails: yniu@stu.pku.edu.cn, changxinhai@stu.pku.edu.cn, liuxin@bigai.ai, jiaoziyuan@bigai.ai, yixin.zhu@pku.edu.cn

(a) Diverse articulated objects



(b) Alternative solutions

(c) Different robot morphologies



(d) Various task setups and grasp-switching

Fig. 1: **AutoMoMa generates whole-body trajectories for coordinated mobile manipulation tasks.** It covers (a) diverse articulated objects, (b) alternative solutions, (c) different robot morphologies, and (d) various task setups and grasp-switching scenarios.

approaches. This efficiency enables the creation of comprehensive datasets spanning diverse household environments, object types, robot morphologies, and manipulation tasks without requiring extensive human demonstration or real-world trials. Our platform extends beyond simple pick-and-place operations to handle complex articulated object interactions and multi-stage manipulation requiring strategic grasp-switching in confined spaces. Through extensive simulation studies and real-world validation on a dual-UR5 Clearpath Ridgeback platform, we demonstrate AutoMoMa's effectiveness in producing trajectories that transfer successfully to physical robots.

Our contributions are threefold:

- **Computationally efficient data generation:** We overcome the VKC's computational bottlenecks through GPU-accelerated motion planning, achieving generation rates of approximately one valid, constraint-compliant whole-body trajectory per second, enabling creation of diverse, large-scale datasets that were previously infeasible.
- **Comprehensive environmental support:** Our platform supports household scenes in USD format, articulated objects with varied kinematic structures in URDF format, robot morphologies with URDF description, and household manipulation tasks defined by start and goal states. These assets are either publicly available or easily constructed using standard simulation pipelines, ensuring compatibility with common robotics frameworks and facilitating rapid adoption.
- **Advanced manipulation capabilities:** We extend beyond fixed-grasp manipulation by implementing a trajectory splicing approach that enables strategic grasp-switching during task execution, allowing robots to complete complex manipulation sequences in highly constrained spaces where traditional approaches fail due to kinematic limitations.

II. Related Work

A. Data Collection for Mobile Manipulation

a) Simulated Embodied AI Platforms: Modern simulation suites like Habitat 2.0 [33], AI2-THOR [22], OmniGibson [24], and RoboHive [23] offer thousands of photorealistic environments with articulated assets and multiple robot embodiments. However, these platforms typically prioritize visual fidelity over physically plausible motion—interactions are often simplified to teleport-to-handle primitives or scripted end-effector trajectories that bypass robot-specific kinematics, collision avoidance, and whole-body coordination challenges. Consequently, despite their environmental richness, no standard dataset of physically valid base-arm-object trajectories has emerged from these platforms. ManiSkill-HAB [28] begins addressing this gap with 8,000 demonstrations of coordinated base-arm motion for table-setting tasks, but its single kitchen layout and limited task complexity constrain generalization to diverse environments and robot morphologies.

b) Teleoperation: Human-guided data collection through teleoperation provides an alternative approach to capturing realistic mobile manipulation behaviors. Systems like MOCA [35] and MOMA-Force [37] enable operators to guide robots through pick-and-place and object manipulation tasks in real environments. However, these systems typically record only end-effector positions and base velocities rather than complete joint-space trajectories, limiting their utility

for learning whole-body coordination. More recent platforms like Mobile ALOHA [11] and TeleMoMa [7] capture full joint-space data with high physical fidelity, but remain constrained by operator fatigue, hardware limitations, and environment accessibility. These practical constraints restrict teleoperation-based datasets to thousands rather than millions of trajectories, with limited object diversity and typically a single robot morphology.

c) Standalone Mobile Manipulation Datasets: Despite growing interest in mobile manipulation, comprehensive datasets remain scarce. BC-Z [17] provides 25,000 demonstrations across 100 tabletop tasks, but mostly involves stationary bases and records only end-effector poses rather than full base-arm trajectories. Mobile ALOHA [11] contributes 276 teleoperated joint-space trajectories coupling a mobile base with a 7-DoF arm, but this dataset is limited to a single robot platform and lacks the scale needed for data-hungry learning methods. The field currently lacks a publicly available dataset offering extensible coverage of diverse scenes, articulated objects, and task variants with verified whole-body motion across multiple robot embodiments. Our AutoMoMa platform addresses this critical gap by enabling scalable, automated generation of diverse, constraint-compliant trajectories that support large-scale learning and benchmarking across a comprehensive range of mobile manipulation scenarios.

B. Motion Planning for Mobile Manipulation

a) Learning-based Planning: Recent research has explored end-to-end deep RL approaches for coordinated base-arm control in simulated environments [10, 36]. While RL can theoretically discover sophisticated coordination strategies through exploration, it remains notoriously sample-inefficient—requiring millions of interactions to learn even basic skills—and struggles to generalize across novel environments or robot morphologies without extensive retraining [30]. Imitation learning offers a more data-efficient alternative by leveraging human demonstrations [11, 17], but remains inherently constrained by the distribution and quantity of available demonstrations. The effectiveness of both approaches is fundamentally limited by the scarcity of diverse, physically plausible training data that captures the full complexity of whole-body coordination. AutoMoMa addresses this bottleneck by providing a scalable source of high-quality trajectories that can serve as training data for learning-based methods or as initialization for hybrid approaches combining data-driven and model-based components.

b) Model-Based Planning: Classical model-based approaches to mobile manipulation span from specialized controllers for specific tasks—such as impedance and model-predictive controllers for door and drawer manipulation [15, 20, 29]—to more general base-arm optimization frameworks for cluttered environments [2, 13, 3]. While these methods can achieve reliable performance under controlled conditions, they typically require extensive hand-tuning for each new robot-object pair and do not readily scale across diverse environments or object types without significant manual intervention. The VKC framework [18, 19] represents a significant advance by integrating the mobile base, manipulator, and manipulated object into a unified kinematic model. This integration enables constraint-aware whole-body planning within a single configuration space, with natural handling of articulated object constraints. Recent work [38] has demonstrated VKC's feasibility for generating mobile manipulation trajectories and training control policies for indoor pick-and-place tasks. However, the framework's broader potential remains constrained by CPU-based optimization bottlenecks, fixed-grasp assumptions, and limited task variants. AutoMoMa builds upon the VKC foundation while addressing prior limitations through GPU-accelerated planning, support for grasp-switching, an automated environment preprocessing pipeline, and broader support for diverse task constraints. These enhancements establish AutoMoMa as the most scalable and versatile platform for large-scale mobile manipulation data generation currently available.

III. Preliminary

This section briefs VKC-based mobile manipulation planning, illustrating how the VKC formulation enables scalable whole-body trajectory generation for manipulating both rigid and articulated objects. We begin by introducing the VKC modeling, then formulate motion planning problems from the VKC perspective, and finally describe how task and environmental constraints are incorporated into this framework.

A. VKC Modeling

The objective of the modeling is to construct a serial VKC by composing the kinematics of the mobile base, manipulator arm, and the object to be manipulated [19]. This requires three inputs: 1) the robot's kinematic tree, 2) the object's kinematic tree, and 3) the transformation between the robot's end-effector and the object's attachable frame (*i.e.*, the grasping pose). The procedure for constructing the VKC consolidates the robot and object kinematic models as follows.

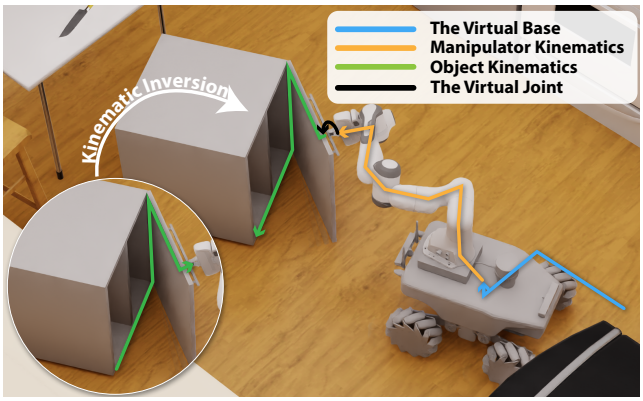


Fig. 2: **Modeling a mobile manipulation task using VKC.** Constructing a VKC involves four key components: the manipulator's kinematics, the object's kinematics, a virtual mobile base, and a virtual joint that connects the manipulator to the object. For articulated objects like cabinets, their kinematic models must be inverted to preserve a valid serial kinematic tree.

The kinematic structures of the robot and the object are each represented as separate kinematic trees (*e.g.*, Unified

Robot Description Format (URDF)), as shown in Fig. 2. To form a serial VKC, we insert a virtual joint, corresponding to the grasping pose, between the robot's end-effector and the object. This requires inverting the object's kinematic model. Importantly, inverting a kinematic tree is not simply a matter of reversing the parent–child relationships; all associated transformations, including those branching structures, must be carefully updated, as revolute and prismatic joints typically define motion relative to the child link's frame. Moreover, the geometry of branching structures must also be considered during trajectory optimization to ensure safety and feasibility.

To jointly optimize locomotion and manipulation, we further insert a virtual base that models the mobile base's planar motion. This is implemented using two orthogonal prismatic joints and one revolute joint between the virtual base and the robot's base, allowing for planar motion while preserving a serial kinematic structure.

Fig. 2 illustrates a constructed VKC for a door-opening task. The resulting VKC begins with a fixed virtual link and ends at the object link connected to the ground (*e.g.*, a door's frame). The mobile base and manipulator are embedded within this serial chain. Consequently, the states of the mobile base, arm, and object are jointly represented within the VKC configuration space. Task goals and kinematic constraints are subsequently imposed during trajectory optimization, as described in the next section.

B. VKC-Based Mobile Manipulation Planning

The mobile manipulation planning problem can be modeled as finding a collision-free trajectory within the configuration space of the VKC. Formally, the resulting VKC state is defined as:

$$\boldsymbol{x} = \left[\boldsymbol{q}_B^\mathsf{T}, \boldsymbol{q}_M^\mathsf{T}, \boldsymbol{q}_O^\mathsf{T}\right]^\mathsf{T} \in \mathcal{X}_{\text{free}}, \tag{1}$$

where $\boldsymbol{q}_B \in \mathbb{R}^3$ is the mobile base pose, $\boldsymbol{q}_M \in \mathbb{R}^n$ is the manipulator joint state (n is the Degree of Freedom (DoF) of the manipulator), $\boldsymbol{q}_O \in \mathbb{R}^m$ is the articulated object's joint state (m is the DoF of the articulated object, 0 for rigid object), and $\mathcal{X}_{\text{free}}$ is the collision-free configuration space. Then, the motion planning problem seeks a collision-free path of length T: $\boldsymbol{x}_{1:T} = \langle \boldsymbol{x}_{[1]}, \ldots, \boldsymbol{x}_{[T]} \rangle \subset \mathcal{X}_{\text{free}}$.

During trajectory optimization (following [18]), we enforce:

$$h_{\text{chain}}(\boldsymbol{x}_{[t]}) = 0, \qquad \forall t = 1, \ldots, T, \tag{2}$$

$$\|f_{\text{task}}(\boldsymbol{x}_{[T]}) - \boldsymbol{g}_{\text{goal}}\|_2^2 \leqslant \xi_{\text{goal}}, \tag{3}$$

$$\boldsymbol{x}_{\min} \leqslant \boldsymbol{x}_{[t]} \leqslant \boldsymbol{x}_{\max}, \qquad \forall t = 1, \ldots, T, \tag{4}$$

$$\|\Delta \boldsymbol{x}_{[t]}\|_\infty \leqslant \Delta \boldsymbol{x}_{\max}, \qquad \forall t = 1, \ldots, T-1, \tag{5}$$

$$\|\Delta \dot{\boldsymbol{x}}_{[t]}\|_\infty \leqslant \Delta \dot{\boldsymbol{x}}_{\max}, \qquad \forall t = 2, \ldots, T-1. \tag{6}$$

Here, Eq. (2) enforces the physical constraints arising from robot–environment interactions (*e.g.*, a door hinged to the ground, or a chair constrained to planar motion along the floor); Eq. (3) bounds the task goal via $f_{\text{task}} : \mathcal{X} \to \mathcal{G}$ with tolerance ξ_{goal}; and Eq. (4)–Eq. (6) impose joint limits, velocity, and acceleration bounds. Collision avoidance is handled by the underlying motion planner's self- and environment-collision checks.

IV. Data Generation Pipeline

Our data generation pipeline consists of four main stages. First, we prepare the planning context by loading a scene, a set of rigid and articulated objects, and a robot embodiment into the simulator. Second, we construct VKC for each robot-object pair and construct environment collision models. Third, we invoke the planner to generate whole-body trajectories for different task setups. Finally, we post-process the optimized trajectories to enforce constraints and render RGB-D images and point clouds in Isaac Sim, producing a multi-modal dataset suitable for downstream learning and evaluation tasks.

A. Planning Contexts

The AutoMoMa pipeline receives a triplet $(\mathcal{S}, \mathcal{O}, \mathcal{R})$ that jointly defines the motion planning context: a static *scene* $\mathcal{S}$, a finite set of *interactive objects* $\mathcal{O}$, and a *robot embodiment* $\mathcal{R}$.

a) Household scene layouts: Each scene $\mathcal{S}$ contains the geometry, appearance, and semantic tags for all static elements such as floors, walls, countertops, and fixed appliances. A world frame is anchored at the scene's geometric center; all scene objects consist of visual and collision meshes for visualization and collision checking, respectively.

b) Interactive objects: The object set $\mathcal{O} = \mathcal{O}_{\text{rigid}} \cup \mathcal{O}_{\text{art}}$ contains *rigid bodies* $\mathcal{O}_{\text{rigid}}$ and *articulated objects* $\mathcal{O}_{\text{art}}$. Every rigid object $o \in \mathcal{O}_{\text{rigid}}$ consists of a watertight mesh and a set of grasp poses expressed in the object's base frame. For each articulated object $o \in \mathcal{O}_{\text{art}}$, we require a URDF specifying joint types, axes, limits, and inertial parameters. Grasp poses are state-dependent—for example, a closed cabinet may afford different grasps than when open. The articulated objects are inverted for VKC modeling, as introduced in Sec. III-A.

c) Robot embodiments: A robot embodiment $\mathcal{R}$ consists of a virtual mobile base and a manipulator. Both are defined following URDF; an auxiliary file provides (i) a spherical approximation of collision geometry, (ii) a self-collision mask, and (iii) a joint-weight vector $\boldsymbol{w} \in \mathbb{R}^{n+m+3}$ used by the trajectory optimization. Any robot embodiment that satisfies the above description can be loaded without further modification. This paper has validated it on a Franka arm mounted on a Summit base, the R1 robot adopted from OmniGibson [24], and the Tiago model from PAL Robotics.

B. VKC Construction and Collision Processing

Manipulated objects are integrated into robotic manipulation pipelines via the following workflow.

a) Preprocessing: Since standalone datasets typically provide objects at a fixed scale, we resize them to fit the scene and update grasp poses accordingly. In this process, the geometric components of each link are merged into a single mesh and scaled accordingly. Since scaling alters the spatial configuration, joint origins are updated to preserve valid kinematic relationships.

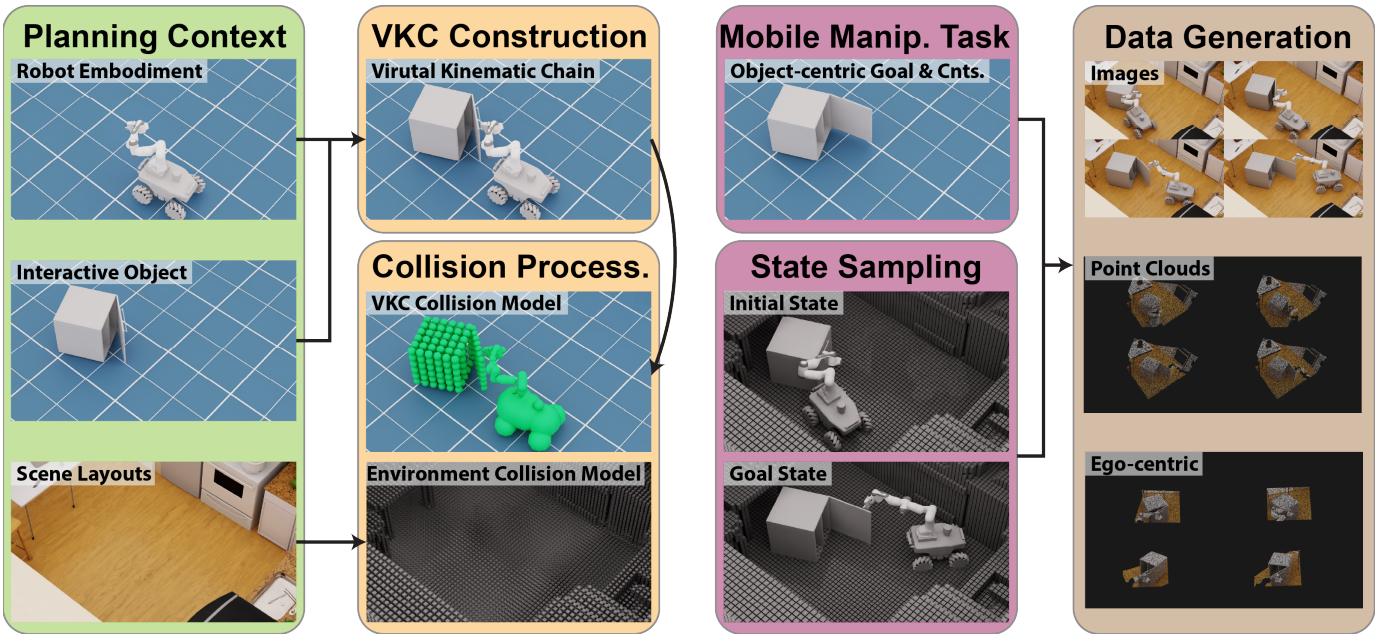


Fig. 3: **AutoMoMa data generation pipeline.** The pipeline begins by preprocessing planning contexts through VKC construction and collision processing. It then models mobile manipulation tasks from the VKC perspective and solves them via trajectory optimization. Finally, the data undergoes postprocessing to enforce constraints and generate multi-modal outputs.

b) VKC construction: To construct a VKC, the post-inversion object model is treated as an extended limb of the robot, The grasp pose defines the transformation between the robot's end-effector and the target object link. This transformation, along with the two associated links, forms a virtual joint that connects the object to the robot, yielding a unified kinematic model $\mathcal{K}_{\text{vkc}}$ for integrated motion planning.

c) Collision Processing: To enable efficient collision checking in the GPU-accelerated motion planner, each VKC's geometry is approximated using a set of fitted spheres. To avoid overestimating the original shape, the merged mesh is slightly downscaled before fitting spheres to its geometry. In rare cases of translational shifts, the sphere cloud's centroid is aligned with that of the original mesh to preserve geometric consistency. Finally, we identify negligible collision pairs (*e.g.*, adjacent links that are always in contact) in the VKC, ensuring efficient collision checking.

d) Environment Collision Models: Each scene is converted into an Euclidean Signed-distance Field (ESDF) to accelerate collision checking. During planning, only the ESDF voxels within an axis-aligned bounding box, defined by the target object's start and goal states, are considered, further limiting collision checks to the local workspace and reducing unnecessary computations.

C. Trajectory Generation

This section outlines how mobile manipulation planning problems are formulated using the VKC framework.

a) Defining Task Objectives and Goals: The mobile manipulation planning objective minimizes total traveling distance and trajectory non-smoothness:

$$\sum_{t=1}^{T-1} ||\boldsymbol{w}_v \Delta \boldsymbol{x}_{[t]}||_2^2 + \sum_{t=2}^{T-1} ||\boldsymbol{w}_a \Delta \dot{\boldsymbol{x}}_{[t]}||_2^2, \tag{7}$$

where $\boldsymbol{w}_v$ and $\boldsymbol{w}_a$ are diagonal weight matrices over each DoF, enabling modulation of base–arm coordination strategies. Task goals are object-centric: for rigid-object relocation, the goal is the grasp pose to the object or a target placement pose of the object; for articulated objects, the goal is a desired object state (*e.g.*, a door opened to a specific joint angle).

b) Specifying Task Constraints: Trajectory constraints are defined based on the object–scene relationship and task type. For rigid object relocation, the object is treated as a free joint. For tasks involving large objects or specific task requirements—such as pushing a chair or sweeping a table—we impose planar constraints on the VKC's end-effector (*i.e.*, the object's base link) to ensure stable, planar motion. When manipulating articulated objects fixed to the environment, we enforce a fixed constraint on the VKC's end-effector, penalizing deviations of the object's location from its initial pose via a pose cost.

c) Start and Goal Configurations: To initialize the motion planning problem, we compute start and goal VKC configurations under the assumption of a fixed grasp pose during execution. These configurations are obtained by solving Inverse Kinematics (IK) for both object states. Similar VKC configurations are removed through clustering, yielding a compact yet diverse set of candidate configurations. This reduces planning overhead while maintaining broad workspace coverage, facilitating efficient trajectory optimization.

d) Grasp Switching: Grasp switching is critical when a single grasp cannot maintain stability or reachability, such as opening a dishwasher with a handle positioned near the floor,

making it inaccessible to the robot in one continuous grasp. To address this, we first sample an intermediate object state ϕ_{mid} between the start ϕ_0 and goal ϕ_T. We then solve for two sets of IK solutions: one using the initial grasp for $[\phi_0 \to \phi_{\text{mid}}]$ and the other using the final grasp for $[\phi_{\text{mid}} \to \phi_T]$. A short transition trajectory is planned between the two grasp configurations to enable collision-free detachment and reattachment. The three segments are concatenated into a continuous motion, yielding smooth trajectories with grasp switches executed only when necessary.

D. Data Generation

After trajectory optimization, we refine optimized trajectories to prone constraint-violated trajectories and synthesize realistic sensor observations for downstream tasks.

a) Trajectory Post-Processing: This stage verifies that each trajectory waypoint $x_{[t]}$ satisfies the required motion constraints. For fixed-base tasks, we compute the translational deviation $d = |p(x_{[t]}) - p(x_{\text{ref}})|$ and rotational deviation $\theta = \cos^{-1}\left(2\langle r(x_{[t]}), r(x_{\text{ref}})\rangle^2 - 1\right)$, where $p(\cdot)$ and $r(\cdot)$ denote the translational and rotational components of the VKC forward kinematics, and x_{ref} is the reference configuration. For planar constraints, such as requiring motion constrained to the XY plane, we evaluate the vertical displacement $d_z = |p_z(x_{[t]}) - p_z(x_{\text{ref}})|$ and rool-pitch deviation $\theta_{\text{planar}} = |\psi(x_{[t]}) - \psi(x_{\text{ref}})|$, where $p_z(\cdot)$ is the z-axis translation and $\psi(\cdot)$ denotes roll and pitch. Trajectories violating any thresholded constraint are discarded. This process ensures all retained trajectories satisfy the specified kinematic constraints for stable, physically plausible execution.

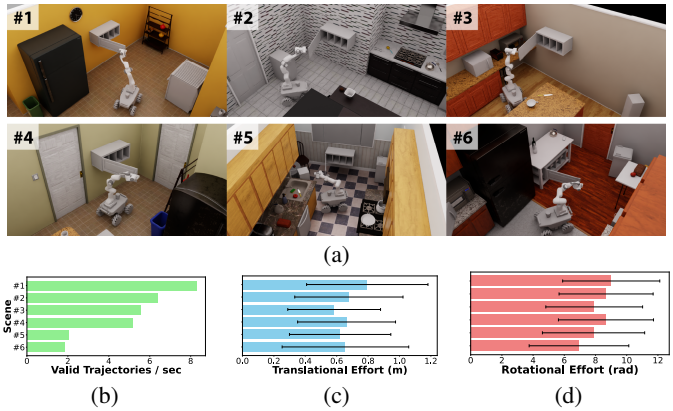
b) Multi-Modal Data Rendering: We integrate both egocentric and fixed RGB-D cameras into each scene using NVIDIA Isaac Sim, configuring synchronized color and depth sensors on the robot and in the environment. At each trajectory waypoint, Isaac Sim renders high-fidelity RGB images and aligned depth maps, which are directly converted into point clouds in the simulation's coordinate frame. Camera placements are fully customizable, and scenes can be re-rendered by replaying the generated trajectories. The resulting dataset supports a wide range of downstream tasks, including imitation learning [9, 14], visual servoing [31, 16], and affordance detection [5, 8].

E. Trajectory Generation Performance

To evaluate the effectiveness and generalizability of our trajectory generation framework, we conduct experiments across six representative kitchen scenes from the data environment. Each scene poses unique spatial constraints and increasing layout complexity (lower complexity means more collision-free IK could be found in that scene) (Fig. 4a). We deploy the Summit Franka mobile manipulator and execute a common articulated object manipulation task, opening a wall-mounted cabinet with an unwieldy door, in each environment.

We evaluate three key metrics:

- **Generation Speed:** Measured as valid trajectories (*i.e.*, trajectories that passed trajectory post processing) generated per second. The results are shown in Fig. 4b. Simpler layouts (*e.g.*, Scene #1 and #2) achieve higher data generation speed, while tightly constrained environments (*e.g.*, Scene #5 and #6) reduce generation speed due to limited spaces that constrain base movement.
- **Translational Effort:** Defined as the average distance traveled by the mobile base per trajectory. As shown in Fig. 4c, variations in base effort result in a diverse set of trajectories within the dataset.
- **Rotational Effort:** Measured by the cumulative angular motion of the arm. Similarly, Fig. 4d illustrates that variation in arm effort also contributes to the diversity of trajectories in the dataset.



(a) (b) (c) (d)

Fig. 4: **Evaluation of trajectory generation performance across six representative household scenes.** (a) Visualizations of test scenes, with increasing confinement for realistic mobile manipulation. (b) Generation speed is measured as valid trajectories per second; simpler layouts result in higher throughput. (c) Average translational effort of the mobile base per trajectory, with error bars indicating standard deviation. (d) Average rotational effort of the manipulator, reflecting the compensatory motion required in constrained environments.

V. Real Robot Experiments

We validate our planning pipeline on a physical UR5-Ridgeback system, which comprises two UR5 manipulators mounted on a Clearpath Ridgeback mobile base. Two representative tasks were tested: opening a drawer and opening a cabinet door. In both tasks, the robot executed the planned trajectories smoothly, accurately reproducing the motion patterns generated in simulation without collisions or constraint violations.

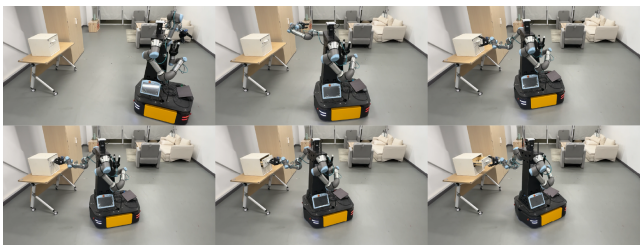


Fig. 5: UR5-Ridgeback executing a planned drawer-opening trajectory.

VI. Limitations and Conclusion

AutoMoMa introduces a scalable whole-body trajectory generator for coordinated mobile manipulation based on VKC-

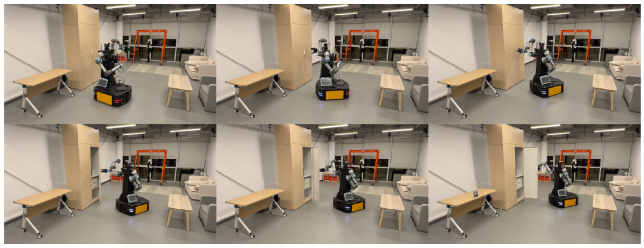


Fig. 6: UR5-Ridgeback opening a cabinet door using the planned motion.

base mobile manipulation planning, generating physically valid trajectories across diverse scenes and robot embodiments. While the use of fixed layouts and known kinematic models limits coverage of highly cluttered, outdoor, or deformable-object scenarios, the framework remains efficient, extensible, and well-suited for scalable data generation. Looking ahead, we aim to incorporate learning-based methods to further automate data generation, and to release community-driven tools for integrating new scenes, robots, and assets into the AutoMoMa ecosystem. Future extensions may also support automatic generation of task and scene assets to further improve scalability and diversity.

References

[1] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 1

[2] Dmitry Berenson, James Kuffner, and Howie Choset. An optimization approach to planning for mobile manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2008. 3

[3] Daniel M Bodily, Thomas F Allen, and Marc D Killpack. Motion planning for mobile robots using inverse kinematics branching. In *International Conference on Robotics and Automation (ICRA)*, 2017. 3

[4] Federico Ceola, Lorenzo Natale, Niko Sünderhauf, and Krishan Rana. LHManip: A dataset for long-horizon language-grounded manipulation tasks in cluttered tabletop environments. *arXiv preprint arXiv:2312.12036*, 2023. 1

[5] Fu-Jen Chu, Ruinian Xu, Landan Seguin, and Patricio A. Vela. Toward Affordance Detection and Ranking on Novel Objects for Real-World Robotic Manipulation. *IEEE Robotics and Automation Letters*, 4(4):4070–4077, October 2019. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2019.2930364. 6

[6] Wenbo Cui, Chengyang Zhao, Songlin Wei, Jiazhao Zhang, Haoran Geng, Yaran Chen, and He Wang. GAPartManip: A large-scale dataset for generalizable and actionable part manipulation with material-agnostic articulated objects. In *IEEE International Conference on Robotics and Automation*. IEEE, 2025. 1

[7] Shivin Dass, Wensi Ai, Yuqian Jiang, Samik Singh, Jiaheng Hu, Ruohan Zhang, Peter Stone, Ben Abbatematteo, and Roberto Martín-Martín. Telemoma: A modular and versatile teleoperation system for mobile manipulation. *arXiv preprint arXiv:2403.07869*, 2024. 3

[8] Thanh-Toan Do, Anh Nguyen, and Ian Reid. AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5, Brisbane, QLD, May 2018. IEEE. ISBN 978-1-5386-3081-5. doi: 10.1109/ICRA.2018.8460902. 6

[9] Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, 3(4):362–369, December 2019. ISSN 2366-5971, 2366-598X. doi: 10.1007/s41315-019-00103-5. 6

[10] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023. 1, 3

[11] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024. 1, 3

[12] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7081–7091, 2023. 1

[13] Kalin Gochev, Alla Safonova, and Maxim Likhachev. Planning with adaptive dimensionality for mobile manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2012. 3

[14] Jiang Hua, Liangcai Zeng, Gongfa Li, and Zhaojie Ju. Learning for a Robot: Deep Reinforcement Learning, Imitation Learning, Transfer Learning. *Sensors*, 21(4): 1278, February 2021. ISSN 1424-8220. doi: 10.3390/s21041278. 6

[15] Advait Jain and Charles C Kemp. Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control. In *International Conference on Robotics and Automation (ICRA)*, 2010. 3

[16] Farrokh Janabi-Sharifi, Lingfeng Deng, and William J. Wilson. Comparison of Basic Visual Servoing Methods. *IEEE/ASME Transactions on Mechatronics*, 16(5):967–983, October 2011. ISSN 1083-4435, 1941-014X. doi: 10.1109/TMECH.2010.2063710. 6

[17] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 3

[18] Ziyuan Jiao, Zeyu Zhang, Xin Jiang, David Han, Song-Chun Zhu, Yixin Zhu, and Hangxin Liu. Consolidating kinematic models to promote coordinated mobile manipulations. In *IROS*, 2021. 1, 3, 4

[19] Ziyuan Jiao, Zeyu Zhang, Weiqi Wang, David Han, Song-Chun Zhu, Yixin Zhu, and Hangxin Liu. Efficient task planning for mobile manipulation: A virtual kinematic chain perspective. In *IROS*, 2021. 1, 3

[20] Yiannis Karayiannidis, Christian Smith, Francisco Eli Vina Barrientos, Petter Ögren, and Danica Kragic. An adaptive control approach for opening doors and drawers under uncertainties. *Transactions on Robotics (T-RO)*, 32 (1):161–175, 2016. 3

[21] Oussama Khatib. Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems*, 26(2-3): 175–183, 1999. 1

[22] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 2

[23] Vikash Kumar, Rutav Shah, Gaoyue Zhou, Vincent Moens, Vittorio Caggiano, Abhishek Gupta, and Aravind Rajeswaran. Robohive: A unified framework for robot learning. *Advances in Neural Information Processing Systems*, 36:44323–44340, 2023. 2

[24] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023. 1, 2, 4

[25] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1647–1654. IEEE, 2022. 1

[26] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021. 1

[27] Giulio Schiavi, Paula Wulkop, Giuseppe Rizzi, Lionel Ott, Roland Siegwart, and Jen Jen Chung. Learning agent-aware affordances for closed-loop interaction with articulated objects. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5916–5922. IEEE, 2023. 1

[28] Arth Shukla, Stone Tao, and Hao Su. ManiSkill-HAB: A benchmark for low-level manipulation in home rearrangement tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[29] Marvin Stuede, Kathrin Nuelle, Svenja Tappe, and Tobias Ortmaier. Door opening and traversal with an industrial cartesian impedance controlled mobile robot. In *International Conference on Robotics and Automation (ICRA)*, 2019. 3

[30] 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 *Conference on Robot Learning*, pages 308–319. PMLR, 2022. 3

[31] Xiaoying Sun, Xiaojun Zhu, Pengyuan Wang, and Hua Chen. A Review of Robot Control with Visual Servoing. In *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 116–121, Tianjin, China, July 2018. IEEE. ISBN 978-1-5386-7057-6. doi: 10.1109/CYBER.2018.8688060. 6

[32] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023. 1

[33] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021. 2

[34] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023. 1

[35] Yuqiang Wu, Pietro Balatti, Marta Lorenzini, Fei Zhao, Wansoo Kim, and Arash Ajoudani. A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant. *IEEE Robotics and Automation Letters*, 4(4):3593–3600, 2019. 2

[36] Fei Xia, Chengshu Li, Roberto Martín-Martín, Or Litany, Alexander Toshev, and Silvio Savarese. Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4583–4590. IEEE, 2021. 1, 3

[37] Taozheng Yang, Ya Jing, Hongtao Wu, Jiafeng Xu, Kuankuan Sima, Guangzeng Chen, Qie Sima, and Tao Kong. Moma-force: Visual-force imitation for real-world mobile manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6847–6852. IEEE, 2023. 2

[38] Zeyu Zhang, Sixu Yan, Muzhi Han, Zaijin Wang, Xinggang Wang, Song-Chun Zhu, and Hangxin Liu. M3Bench: Benchmarking whole-body motion generation for mobile manipulation in 3D scenes. *arXiv preprint arXiv:2410.06678*, 2024. 1, 3