Tinker 2025 Team Description Paper

Songchuan Lim, Cindy Wang, Langzhe Gu, Xinyao Qin,
Zhiting Zhou, Bingchuan Wei, Yunfei Li, Wang You, Chen Rui

November 25, 2024

Abstract. This paper presents an overview of our competition robot, Tinker, including its structure and the innovative technologies employed in our research. We begin by introducing our research team's interests and focus, which center around the development of dexterous manipulation and grasping techniques in robots. Our approach involves tackling these complex tasks through a learning-based methodology. To facilitate our research, we have developed the SAPIEN simulation platform and conducted extensive investigations in the field of sim-to-real transfer. We provide a detailed system flowchart for task implementation and highlight the core technologies and methods utilized across various tasks. Additionally, we outline the hardware structure of the robot and offer a fully open-source chassis design in the appendix, aiming to contribute to the research community and serve as a valuable reference.

1 Introduction

The Future Robotics Club (FuRoC) is a student-led initiative comprising of undergraduate and graduate students from various academic departments at Tsinghua University, including Electronic Engineering, Software Engineering, Mechanical Engineering, Computer Science, and Automation. FuRoC is dedicated to the advancement of domestic robots with genuine artificial intelligence capabilities.

Our objective is to construct a comprehensive robotic hardware platform with capabilities in vision, manipulation, speech, and navigation. These capabilities are then refined through rigorous work in the relevant domains. In recent years, our research has focused on the integration of learning with robotics. The objective is to equip our robots with the capability to perform an increasing range of complex household tasks. This endeavor primarily revolves around advancements in reinforcement learning and sim-to-real transfer, which have led to significant breakthroughs in enhancing the success rate of object grasping and completing intricate tasks.

In our quest for robustness against complex challenges, we have integrated large language models, 3D navigation technologies, and behavior trees into our

task execution strategies, thereby markedly enhancing the efficacy of our robot. Our dedication to community contribution and collective advancement is evidenced by our comprehensive open-sourcing of simulation environments and hardware platforms, which provide invaluable references for peer teams.

FuRoC, which has a commendable track record in RoboCup@Home, having secured 5th place in 2016 and 7th in 2019, is preparing for its 9th participation in the @Home League of the World RoboCup. Despite a period of reduced membership and resources following the pandemic, our core team has continued to pursue research and development activities with great dedication. We are also pleased to contribute to the organisation of the competition, with one of our team members joining the RoboCup@home organizing committee. Following our return to the competition last year, we have come to recognize the limitations of our approach, and have since re-designed our hardware and are in the process of integrating multiple cutting-edge deep-learning models into our vision, manipulation, audio, and navigation modules. By fervent desire to demonstrate our advancements and to contribute to the ever-evolving field of domestic robotics, we are here again admitting ourselves into World RoboCup.

2 Scientific Research

The following section will highlight the contributions of our team to the field of domestic robotics, with a particular focus on the application of learning and simulation-to-reality methods in the domains of stereo-vision and manipulation.

Learning-based stereo methods typically require extensive datasets with depth information, which is challenging to acquire accurately in real-world settings. In contrast, accurate ground-truth depth is readily available in simulation environments. Additionally, accurately manipulating articulated objects poses a challenging yet essential task for real robot applications. Therefore, our research focus is concentrated on enhancing sim-to-real methods, applying them specifically to stereo vision and manipulation.

2.1 Advancements in 3D Sensing

Existing depth sensors cannot capture the accurate and complete depth information of optically-challenging objects, such as transparent and translucent objects, which limits their applicability. To address this problem, we have conducted research in two directions: improving the depth sensing quality in the real world, and synthesizing realistic noisy depth in simulation.

To improve depth sensing quality in the real world, we have proposed ActiveZero [1,2], a mixed domain learning framework for active stereo-vision systems without requiring real-world depth an-

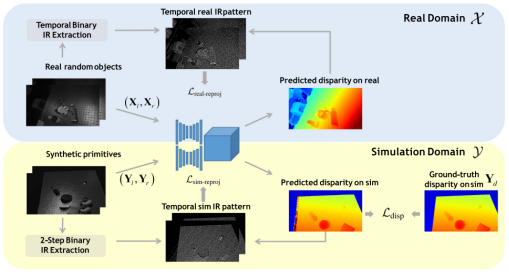

Fig. 1. ActiveZero

notation. It combines supervised and self-supervised losses in both simulated and real domains. This comprehensive approach leads to results that surpass commercial depth sensors, showcasing the effectiveness of each integrated module.

To synthesize realistic noisy depth in simulation, we have developed a physics-grounded simulation pipeline for active stereo-vision depth sensors, producing real-time depth maps with material-dependent error patterns akin to real-world sensors [3]. It effectively transfers perception algorithms and reinforcement learning policies from simulation to real-world applications without additional fine-tuning. Integrated in to the SAPIEN simulator [4], this system is also open-sourced to advance vision and robotics research.

Our two advancements mark a significant leap in 3D sensing, especially in acquiring accurate and complete stereo-vision depth information, enhancing the precision and adaptability of home service robots in diverse and challenging domestic environments.

2.2 Enhancing Articulated Object Manipulation

Recent advancements in domestic robotics have significantly enhanced the manipulation of articulated objects, a key challenge in the field. We have developed two innovative frameworks which leverage the strengths of learning methodologies and sim-to-real approaches.

Sim2Real^2[5] introduces an innovative method for manipulating unseen articulated objects in real scenarios without human guidance. Leveraging advances in physics simulation and learning-based perception, it builds an interactive physics model for long-horizon manipulation trajectory planning. Experimental results show a high success rate in manipulating articulated objects, with less than 30% error, and the ability of advanced manipulation including tool use.

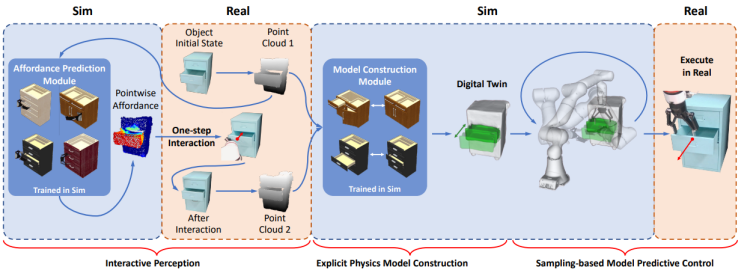

Fig. 2. Sim2Real^2

Parallel to this, our Part-Guided 3D RL framework enhances the manipulation of unseen articulated objects using visual input [6]. It merges 2D segmentation with 3D reinforcement learning, improving RL policy training efficiency. A novel Frame-consistent Uncertainty-aware Sampling strategy improves policy stability on real robots, enabling the training of a single RL policy for multiple tasks and showing strong generalizability in simulated and real-world environments.

Together, these two frameworks mark a significant advancement in the field of domestic robotics, particularly in handling articulated objects. They showcase the potential of integrating learning techniques and sim-to-real approaches to improve the precision and adaptability of robots in executing complex and varied household tasks.

2.3 Revolutionizing Manipulation Skills with ManiSkill2

The development of generalizable manipulation skills stands as a crucial component for domestic robots. Addressing the constraints of current benchmarks, we have introduced ManiSkill2 [7], marking substantial progress.

ManiSkill2 is distinguished by its extensive range of manipulation task families, featuring over 2000 object models and more than 4 million demonstration frames. This wide array encompasses various task types, including stationary and mobile-base tasks, single and dual-arm manipulations, and both rigid and soft-body object interactions. Moreover, our platform significantly boosts the efficiency of visual input learning algorithms, with a CNN-based policy capable of processing about 2000 FPS using a single GPU. This efficiency is complemented by a render server infrastructure, optimizing memory use across environments.

Our approach has propelled learning in robotic manipulation within the realm of home service robots, setting new standards for simulation platforms.

2.4 Large Language Model for Robots

To tackle the challenge of implementing large-scale language models in robotics, we designed OpenChat [8], a streamlined light-weight language model.

Traditional models like GPT and Llama possess parameter counts often in the tens of billions, posing significant challenges for real-time inference in robotic applications due to their immense size. OpenChat [8], however, is designed with a notably smaller parameter count of only 7 billion, striking a balance between compactness and performance. This reduction in

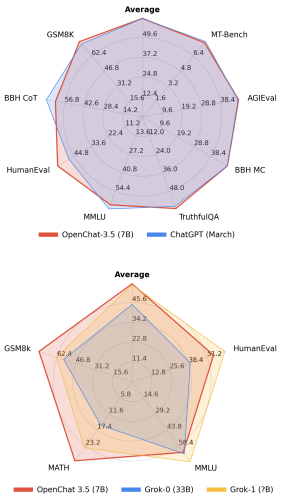

Fig. 3. OpenChat Performance

size enables the model to achieve performance levels comparable to GPT-3.5, yet it remains sufficiently compact for real-time inference on robots, even those equipped with standard graphics cards, such as RTX 4090.

The development of OpenChat represents a crucial step forward in our work, particularly in applying advanced decision-making methods using large models, like function calls, to robotics. This ongoing research has the potential to significantly enhance the capabilities of robots like Tinker in the near future, expanding their computational efficiency and decision-making prowess. The capability of OpenChat is shown in Fig. 3. For more information, please refer to openchat library.

3 Technical Contributions

In this section, we will delve into our technical contributions, emphasizing how we integrate or enhance existing technologies to optimize Tinker's performance in household scenarios. This part is focused on improving task success rates and robustness in the presence of disturbances. Additionally, we provide open-source resources for the community, including all of our decision planning repositories, a mecanum wheel chassis solution and an simulation based on ROS and Gazebo.

3.1 3D Navigation

Our navigation stack based on the ones of Nav2. To achieve customized and intelligent navigation behavior, we have employed behavior trees to orchestrate multiple independent modular servers. In the local planner, we utilize STVL (Spatial-Temporal Voxel Layer), a state-of-the-art 3D perception plugin, within our local static costmap layer. The inclusion of the STVL layer greatly facilitates the modeling of structured building environments, particularly in home scenarios.

In our Tinker project, we incorporate a compact and lightweight 3D lidar, enabling 360-degree perception. This lidar module enhances our obstacle avoidance capabilities by leveraging the power of 3D perception. To effectively utilize this capability, we employ a slicing technique on the point cloud data, based on the z-coordinate, and fuse the sliced data to execute obstacle avoidance. This implementation enables spatial obstacle avoidance, which provides a higher level of security compared to traditional 2D obstacle avoidance, especially for irregularly shaped obstacles. Furthermore, we plan to conduct further research on direct 3D obstacle avoidance for Tinker in the upcoming year.

We are also developing a 3D VSLAM model using rgb-d point cloud information to provide Tinker with spatial and contextual awareness of its surroundings.

3.2 Human Tracking

In the human tracking tasks, we primarily employ the STARK algorithm from the mmtracking library. STARK is an advanced single-object tracking algorithm known for its effectiveness in prolonged tracking of targets in challenging conditions and tracking stability in the presence of disturbances.

To improve tracking in complex scenarios with multiple individuals and enhance the robot's ability to recover tracking after losing the target or experiencing frame drops, we have introduced some multi-modal models. These models adeptly process input image information and extract characteristics of individuals in a textual format, allowing the robot to genuinely understand whom to track. Furthermore, they can locate the bounding box of person based on the extracted textual features. Specifically, we incorporate Grounding Dino [9] that combines text and image modalities. It generates bounding boxes based on input textual content, aligning with the specified features. At the beginning of the tracking task, we input several frames of the target person into a large model resembling GPT-4, requesting the model to output the most prominent features of the person. During tracking, we periodically invoke Grounding Dino to generate bounding boxes corresponding to the specified features and compare them with the current tracking target. In case of target loss, the robot rotates in place to find and resume tracking the person matching the text features.

3.3 Open Source Chassis and Simulation

Behavior Tree Building complex decision systems can be a rather daunting task, which, without properly framework, can quickly spiral out of control into a maze of if-else statements which are hard to understand and modify. Having been in this maze ourselves with our previous prototype, we are aiming to open-source all of our decision systems, starting from the one used in the ZhongGuanCun Biomimetic Robot Competition (ZGC-BRC), where Tinker is among 4 teams who made it to the finals (final placement still pending when paper is written).

Tinker's current decision system is a behavior tree implemented using py_trees, it is capable of handling errors reported by child node and processes and take appropriate measures to rectify the mistakes and ensure the show goes on. All of the code is available on public repositories on Github `tinker_ZGC-BRC`, with more to be updated.

Chassis To foster community research, we have open-sourced both the chassis we developed and the simulated environments we constructed. Opting for a mecanum wheel design for the chassis, we aimed to provide the robot with lateral mobility, particularly valuable in confined home environments. This design allows the robot to move left and right, facilitating the grasping of objects at different positions on a tabletop without the need for multiple rotations. The comprehensive mechatronics solution, including the chassis assembly and accompanying components like drivers and odometry, is available on GitHub `Tinker Chassis`

Simulation We have established two distinct simulation platforms, each serving different purposes. The first simulation platform is built on ROS2 Humble and Gazebo, primarily utilized for navigation testing, MoveIt evaluations, and various ROS message, service, and functionality tests. The second simulation platform is based on SAPIEN and ManiSkill2, specifically tailored for learning and training related to manipulation tasks. Both simulation platforms have been open-sourced on GitHub `Tinker_gazebo_ros2_simulation`, `Tinker_sapien_simulation`.

4 Domestic Task

The following section will provide a concise overview of the specific system implementation and relevant technologies employed by the Tinker robot to successfully complete RoboCup tasks. It will not reiterate the detailed methods discussed in the preceding section.

4.1 Receptionist

Speech We have implemented OpenAI's Whisper for offline speech recognition. To enable our robot to comprehend voice commands beyond mere recognition, we have integrated OpenChat to enhance its understanding of varied expressions. In the receptionist task, we use OpenChat to ensure Tinker can accurately captures guests' names and preferred beverages under different phrasings used in natural language.

Face Recognition In order to support human-robot interaction, the robot is required to recognize different masters or guests in domestic service. We established a face recognition system with two steps: enrollment and recognition. During the enrollment section, a person will be asked to stand in front of the camera. The face detector based on Haar feature from OpenCV is applied and the detected feature will be stored.In the recognition section, We employ the ArcFace algorithm, a deep learning-based approach to achieve face recognition by computing the similarity between feature vectors extracted from different facial images. The similarity between different feature vectors (persons) will tell who is the unknown person.

4.2 Serve Breakfast

Manipulation In order to complete the tasks of delivering things, Tinker needs to finish two related sub tasks, motion planning and grasping. With a 7-DOF Xarm7 arm, Tinker has a great range of motion in 3D space. Tinker carries out its motion planning mainly with the help of MoveIt, using the default OMPL algorithm and Rucking algorithm for its path planning and trajectory generation respectively. To avoid the potential collision with other parts, we restrict the work space of Xarm7 with pre-build urdf models, and updates octomap with the realsense depth camera mounted on the tip of the arm.

Once the desired grasping target has been recognized by the vision module, Tinker utilizes AnyGrasp [10], a cutting-edge grasp perception model, to generate grasping poses, which are then filtered for the best reachable position.

Plane Extraction Real-time plane extraction in 3D point clouds plays a crucial role in many robotics applications. For example, in this task, Tinker needs to distinguish the tabletop from other interfering planes in its field of view. We present an innovative algorithm to reliably detect multiple planes in organized point clouds obtained from devices - such as Realsense sensors, in real time. By uniformly dividing such a point cloud into non-overlapping groups of points in the image space, we are able to construct a graph in which the nodes and edges represent a group of points and their neighborhood respectively. We then perform an hierarchical clustering on this graph to systematically merge nodes that belong to the same plane until the squared error of the plane fitting mean exceeds a threshold. Finally we refine the extracted planes using pixel-wise region growing. Our experiments demonstrate that the proposed algorithm can reliably detect all major planes in the scene at a frame rate of more than 15Hz (for point clouds generated by 1096×720 depth images), which is much faster than many other algorithms we know.

Object Recognition Tinker uses a two-phase approach to recognize objects and precisely manipulate them. In the first phase, a point cloud is built according to the features collected by the Realsense depth camera, and we use Fast Plane Extraction in Organized Point Clouds inside. In simple terms, we first extract the object from the two-dimensional picture, use the ransac and least square method to fit its shape parameters, and then use the known three-dimensional spatial position information to reproject it into the three-dimensional space.

For object recognition, another image processing method is implemented. We use GroundingDINO [9] with fine-tuned prompts to extract bounding boxes, then use a light-weight version of SAM [11] to get precise segments, allowing for fast and accurate zero-shot object detection.

References

1. I. Liu, E. Yang, J. Tao, R. Chen, X. Zhang, Q. Ran, Z. Liu, and H. Su, "Activezero: Mixed domain learning for active stereovision with zero annotation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 033–13 042.
2. R. Chen, I. Liu, E. Yang, J. Tao, X. Zhang, Q. Ran, Z. Liu, J. Xu, and H. Su, "Activezero++: Mixed domain learning stereo and confidence-based depth completion with zero annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
3. X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han *et al.*, "Close the optical sensing domain gap by physics-grounded active stereo sensor simulation," *IEEE Transactions on Robotics*, 2023.
4. F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
5. L. Ma, J. Meng, S. Liu, W. Chen, J. Xu, and R. Chen, "Sim2real2: Actively building explicit physics model for precise articulated object manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 698–11 704.
6. P. Xie, R. Chen, S. Chen, Y. Qin, F. Xiang, T. Sun, J. Xu, G. Wang, and H. Su, "Part-guided 3d rl for sim2real articulated object manipulation," *IEEE Robotics and Automation Letters*, 2023.
7. J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su, "Maniskill2: A unified benchmark for generalizable manipulation skills," in *International Conference on Learning Representations*, 2023.
8. G. Wang, S. Cheng, X. Zhan, X. Li, S. Song, and Y. Liu, "Openchat: Advancing open-source language models with mixed-quality data," *arXiv preprint arXiv:2309.11235*, 2023.
9. S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.
10. H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics (T-RO)*, 2023.
11. A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: https://arxiv.org/abs/2304.02643
12. C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172
13. H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," 2023. [Online]. Available: https://arxiv.org/abs/2212.08333
14. H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

15. C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Graspness discovery in clutters for fast and accurate grasp detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 964–15 973.

Robot Tinker Hardware and Software Description

Mechanical specifications of Robot Tinker are as follows:

- **Base:** AgileX Tracer, 2m/s max speed.
- **Torso:** Aluminum extrusions.
- **Arm:** Mounted on Base. UFACTORY Xarm7 for accessing objects. Maximum load: 5kg.
- **End-Effector:** Mounted on arm. UFACTORY gripper.
- **Head:** 3D-printed mount
- **Robot dimensions:** Width: 0.69m, Length 0.57m, Height: 1.5m (max).
- **Robot weight:** 50kg.

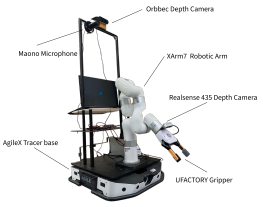

Fig. 4. Tinker

Also our robot incorporates the following devices:

- Femto Bolt- ORBBEC depth camera
- Realsense D435I camera
- MAONO PM461 Microphone
- Livox MID-360 laser scanner

Tinker Software Description

For our robot we are using the following software:

- **Platform:** Ubuntu 22.04 Operating System and ROS2 Humble.
- **Navigation:** ROS2 NAV2
- **Face recognition:** Acrface
- **Object recognition and segmentation:** GroundingDINO
- **Human tracking:** Mediapipe
- **Speech recognition:** Openai Whisper
- **Speech generation:** ROS2 TTS
- **Manipulation:** MoveIt2, AnyGrasp
- **Simulation:** SAPIEN
- **LLM:** OpenChat

External Devices

- **HP OMEN** with RTX 4080, i9-13900HX core