Tinker 2024 Team Description Paper

Qin Xinyao, Zhou Zhiting, Wei Bingchuan,
Li Yunfei, Xie Pengwei, Chen Rui

November 27, 2023

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), comprising undergraduate and graduate students from various departments such as Electrical Engineering, Mechanical Engineering, Computer Science, and Automation at Tsinghua University, is deeply committed to the development of advanced domestic robots with true artificial intelligence capabilities. Our focus extends to crafting a comprehensive robotic hardware platform with capabilities in vision, manipulation, speech, and navigation. We refine these capabilities through rigorous work in these domains. In recent years, our research has shifted towards integrating learning with robotics. The goal is to equip our robots with the proficiency to execute increasingly 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 and 3D navigation technologies into our task execution strategies, significantly augmenting our robot's performance. Our commitment to community contribution and collective progress is demonstrated through the comprehensive open-sourcing of our simulation environments and hardware platforms, providing valuable references for peer teams.

FuRoC, with its commendable track record in RoboCup@Home - securing 5th place in 2016 and 7th in 2019 - is gearing up for its 8th participation in the @Home League of the World RoboCup. Despite a hiatus in recent years due to the pandemic, our team has not waned in its research and development efforts. We are also enthusiastic about assisting with the organization of competition, and one of our team members has joined the Robocup@home organizing committee. As we look forward to rejoining the competition, we are driven by a fervent desire to demonstrate our advancements and to contribute to the ever-evolving field of domestic robotics.

2 Scientific Research

In the subsequent section, we will emphasize our team's contributions to the field of domestic robotics, particularly in the application of learning and simulation-to-real methods in the domains of stereovision and manipulation.

Learning-based stereo methods typically necessitate 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 accurate and complete depth of optical-challenging objects, such as transparent and translucent objects, which limits its 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 the the depth sensing quality in the real world, we have proposed ActiveZero [1,2], a mixed domain learning framework for active stereovision systems without requiring real-world depth annotation. It combines supervised and self-supervised losses in both simulated and real domains. This comprehensive approach

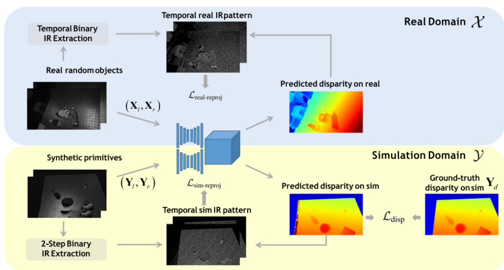

Fig. 1. ActiveZero

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 stereovision 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 into 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 stereovision 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. Two innovative frameworks have been developed by us, leveraging 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.

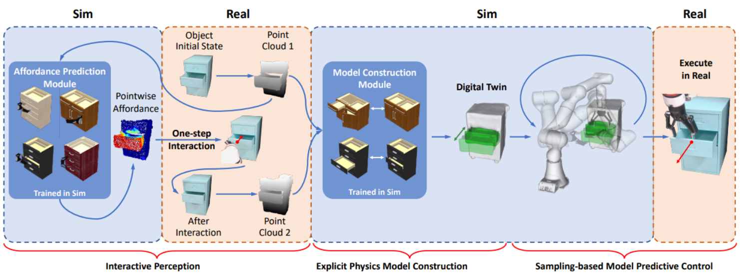

Fig. 2. Sim2Real^2

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 have innovated with the development of OpenChat, a streamlined 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 size enables the model to achieve performance levels comparable to GPT-3.5, yet it remains sufficiently com-

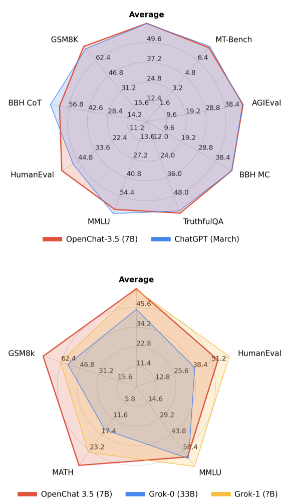

Fig. 3. OpenChat Performance

pact 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 focus on improving task success rates and robustness in the presence of disturbances. Additionally, we provide some open-source resources for the community, including a mecanum wheel chassis solution and an simulation based on ROS and Gazebo.

3.1 3D Navigation

We have implemented the navigation stack based on the Nav2 stack. 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.

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

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`.

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

In the subsequent section, we will present a concise overview of the specific system implementation and relevant technologies employed by the Tinker robot to successfully complete RoboCup tasks. We will refrain from duplicating 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 employ prompts to ensure the robot accurately captures guests' names and preferred beverages under different phrasings.

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 RGB 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 subtasks, montion planning and grasping. With a 6-DOF UR5 arm, Tinker can reach amlost everywhere fexibly in the three-dimensional space. Tinker carries out its montion 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 UR5 with yaml files. Tinker also converts the point clouds from the camera into three-dimensional information to avoid these obstacles in path planning. In the process of grasping, we pre-set the angle of the claw closure and the maximum force to ensure that the grasp force will not damage the objects. The built-in current feedback in Robotiq also plays a role in this process.

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 classification, another image processing method is implemented. We use YOLOv8 [10] for general object type detection, which is a precise while light-weight neural network, designed for general object detection and instance segmentation.

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. G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

Robot Tinker Hardware and Software Description

Mechanical specifications of Robot Tinker are as follows:

- **Base:** Self-built base with a mecanum wheel design, 2m/s max speed.
- **Torso:** Aluminum extrusions.
- **Arm:** Mounted on torso. Universal-Robots UR5 robot arm for accessing objects. Maximum load: 18.4kg.
- **End-Effector:** Mounted on arm. Robotiq-2f140 mechanical gripper.
- **Head:** 2DOF (pan and tilt)
- **Robot dimensions:** height: 1.42m (max), width: 0.58m, depth 0.58m.
- **Robot weight:** 50kg.

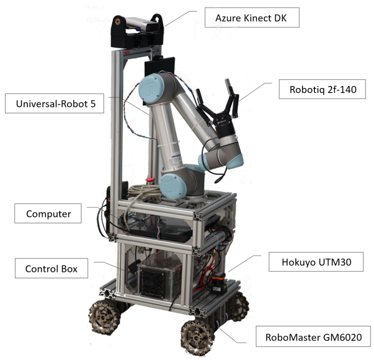

Fig. 4. Tinker

Also our robot incorporates the following devices:

- Dji battery with DCDC transformer for the other equipments
- DJI GM6020 motors
- Azure Kinect DK depth camera
- Realsense D435I camera
- RODE VideoMicro II, Ultra-compact Microphone
- Livox MID-360 laser scanner
- Encoder on motors

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:** YOLOv8
- **Human tracking:** Grounding Dino
- **Speech recognition:** Openai Whisper
- **Speech generation:** ROS2 TTS
- **Manipulation:** MoveIt2
- **Simulation:** SAPIEN
- **LLM:** OpenChat

External Devices

- **ROG laptop** with RTX 4070

Robot software and hardware specification sheet