Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

RoboPanoptes: The All-seeing Robot with Whole-body Dexterity

Xiaomeng Xu[1] Dominik Bauer[2] Shuran Song[1,2]
[1]Stanford University [2]Columbia University

https://robopanoptes.github.io

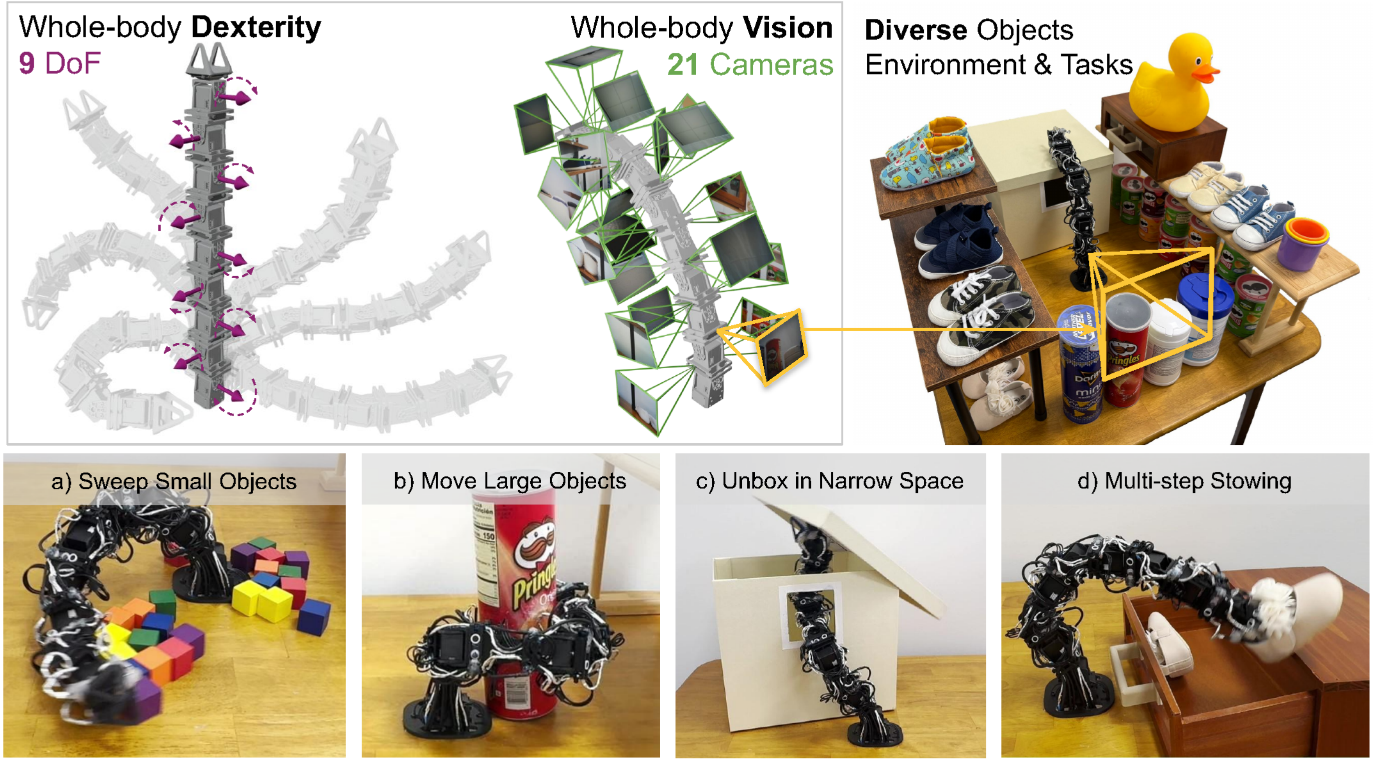

Fig. 1: **RoboPanoptes**, a robot system that utilizes all of its body parts to **sense** and **interact** with its environment. Whole-body vision (via 21 cameras distributed over the robot's body) enables whole-body dexterity, with the robot utilizing its entire surface for manipulation. This design enables new robot capabilities such as a) simultaneously sweeping multiple small objects, b) moving large objects using whole-body contact, c) unboxing in constrained, narrow spaces, and d) executing precise multi-step stowing in cluttered environments.

***Abstract*—We present RoboPanoptes[1], a capable yet practical robot system that achieves whole-body dexterity through whole-body vision. Its whole-body dexterity allows the robot to utilize its entire body surface for manipulation, such as leveraging multiple contact points or navigating constrained spaces. Meanwhile, whole-body vision uses a camera system distributed over the robot's surface to provide comprehensive, multi-perspective visual feedback of its own and the environment's state. At its core, RoboPanoptes uses a whole-body visuomotor policy that learns complex manipulation skills directly from human demonstrations, efficiently aggregating information from the distributed cameras while maintaining resilience to sensor failures. Together, these design aspects unlock new capabilities and tasks, allowing RoboPanoptes to unbox in narrow spaces, sweep multiple or oversized objects, and succeed in multi-step stowing in cluttered environments, outperforming baselines in adaptability and efficiency. Results are best viewed on https://robopanoptes.github.io.**

I. Introduction

The vast majority of today's robot manipulation systems focus on controlling the robot's "end-effector" (*e.g.*, a hand or gripper) using a "centralized camera" (*e.g.*, an environment- or wrist-mounted camera). Despite being the common practice, both designs unnecessarily limit a robot's capability. In this paper, we challenge these conventional designs by introducing **RoboPanoptes**, a novel robot system that achieves **whole-body dexterity** through **whole-body vision**.

By whole-body dexterity, we mean the manipulation skills that utilize the robot's entire body surface to interact with the environment. For example, the robot uses all its links to sweep

[1] *Argos Panoptes* is a many-eyed giant in Greek mythology, with "panoptes" meaning "all-seeing". *RoboPanoptes* is the many-eyed, all-seeing robot.

many objects at once (Fig. 1 a), instead of using only the end-effector to push objects one by one. To demonstrate the benefit of whole-body dexterity, we design a high degree-of-freedom robot, where the redundancy of the degrees of freedom (DoF) allows high motion flexibility and adaptability.

However, achieving whole-body dexterity requires the robot to *coordinate* all its movable body parts, *understand* their spatial relationships with respect to the environment, and *plan* their interactions therewith. Attaining the required level of perception is challenging with a single centralized camera, regardless of its placement. To address this challenge, we introduce the concept of whole-body vision – an integrated camera system that is distributed on the robot's surface, providing comprehensive visual feedback for whole-body dexterity.

Together, these design aspects unlock new capabilities for robotic manipulation systems, including:

- **Multi-contact manipulation** using all body parts, such as simultaneously sweeping multiple objects or moving an object larger than the robot itself (Fig. 1 a, b).
- **Manipulation in constrained spaces**, such as unboxing a container from its side opening or organizing objects in a cluttered environment (Fig. 1 c, d).
- **Omni-directional manipulation**, allowing the robot to perceive and manipulate its environment from any angle without reorienting its body, thereby increasing its efficiency in all tasks by maximizing the share of task-related motion.

However, the increased dexterity and perception complexity create challenges for controlling such a system, and conversely, new opportunities in the system's design:

Control complexity: Programming complex behaviors for such a high degree-of-freedom system is demanding. Previous approaches often rely on manually designed state representations and carefully tuned motion patterns (*e.g.*, sinusoid functions [21, 40]) for every task, which are not transferable for new task requirements. We drastically simplify this process by introducing an intuitive demonstration interface and an efficient policy learning algorithm, allowing even novice users to quickly and easily teach the robot new manipulation skills.

Learning efficiency: For whole-body vision, numerous cameras are distributed on the robot's surface and move in tandem with the robot. Consequently, the policy must efficiently process this complex and high-dimensional input space to infer the appropriate actions. We address this challenge through a novel "whole-body visuomotor policy" architecture based on diffusion transformers [4]. Cross-attention captures semantic correlations between multi-view images and robot actions, while a view-dependent positional encoding preserves crucial spatial information. As we demonstrate for diverse tasks, this architecture enables efficient learning of complex behaviors from limited data.

System robustness: Multi-sensor systems often become increasingly vulnerable to failure as the number of sensors grows, where a single camera malfunction may generate out-of-distribution observations that compromise the overall system performance. To address this challenge, we develop a "Blink Training" strategy that randomly masks camera inputs during the training process. By simulating these unpredictable sensor disruptions – analogous to random "blinking" of the robot's cameras – we are able to train policies that remain reliable even under partial sensor failure or latency.

In summary, our primary contribution is the **RoboPanoptes** system, demonstrating novel whole-body dexterity capabilities through whole-body vision. Specifically, it is enabled by:

- A **modular hardware design** that integrates distributed vision and actuation in a scalable and reconfigurable system.
- An **intuitive demonstration interface** that streamlines the acquisition of complex whole-body manipulation behaviors, dramatically reducing the barrier to teaching new skills.
- A **whole-body visuomotor policy** that efficiently processes whole-body visual input through cross-attention transformers and view-dependent positional encoding, while improving resilience to sensor failures through blink training.

Our hardware design, training data, and policy code are publicly available on https://github.com/real-stanford/RoboPanoptes.

II. Related work

By discussing prior work on designing high-DoF robots, on leveraging them for whole-body manipulation and the closely related challenge of whole-body sensing, we illustrate the novel capabilities of RoboPanoptes with regards to achieving whole-body *dexterity* by exploiting whole-body *vision.*

A. High Degree-of-Freedom Robot Systems

High-DoF robotic systems are characterized by their ability to achieve complex configurations and motions through a large number of controllable joints. This hyper-redundancy enables them to emulate their biological role models – such as snakes, vines [6, 1], and elephant trunks [46] – to perform tasks in confined and complex environments. Snake robots [21, 40, 27, 31, 22, 39, 17, 18] are especially prominent examples of such high-DoF systems. Leveraging hyper-redundancy, they are able to achieve undulating, side-winding, and tumbling movement strategies. While snake robots demonstrate these impressive locomotion capabilities, their use in manipulation remains limited, often restricted to simple grasping [10, 24, 30, 2], hard-coded control strategies [40, 10], or optimization-based path planning with manual primitive designs [30]. In contrast, our work extends the potential of high-DoF systems by incorporating whole-body vision, intuitive data collection interfaces, and visuomotor policies, enabling robust, dexterous manipulation across diverse and complex tasks.

B. Whole-body Manipulation

Stasse and Righetti [35] define whole-body manipulation as requiring high redundancy, a floating base, and multiple contact points. Previous whole-body manipulation systems typically consist of a quadruped with an arm mounted on top [23, 16, 11, 14], or a bimanual system with legs or a mobile base [29, 12, 32, 42, 33, 37]. However, while existing systems optimize body configurations to achieve desired end-effector trajectories, they still fundamentally **restrict environmental**

contact to the end-effector alone. In contrast, whole-body dexterity actively leverages all available body surfaces for environmental contact and manipulation (with or without a movable base). Moreover, the sensing capabilities of existing whole-body manipulation systems are limited. By commonly using a single environment-, wrist- or head-mounted camera, occlusions are unavoidable and hinder their ability to fully observe the state of cluttered environments and the robot's state relative to it.

C. Whole-body Sensing

Prior work on whole-body sensing has explored range, tactile, and force sensing methods to enhance robot perception and interaction, addressing challenges in collision avoidance, contact detection, and compliant motion. Range sensing enables collision avoidance and spatial awareness in manipulation systems. Tanaka et al. [36] implement a semi-autonomous collision avoidance system for snake robots using range sensors distributed along the body, enabling it to navigate complex terrains. Qi et al. [26] further develop a laser-ranging sensor ring design for human-robot interaction. Kim et al. [19] employ Time-of-Flight sensors distributed on a humanoid robot for collision-free motion planning. Whole-body force sensing, as implemented by Kollmitz et al. [20], leverages force-torque sensors embedded within mobile robots to achieve compliant motion. Tactile sensors allow robots to interact delicately with the environment by detecting pressure, texture, and force [7, 25, 8, 17]. Most noticeably, the Punyo robot [13] uses various contact and force sensors throughout the robot's chest, arm, and hand to enable different resolutions of contact signals across the body, which is crucial for contact-rich whole-body manipulation tasks.

While effective in detecting *nearby* obstacles and contacts, the understanding of the *semantics* of the environment remains limited using range, tactile, and force sensing. In contrast, **vision** provides rich information about objects and environments beyond contact or geometry. Most related to our work, Yamaguchi and Atkeson [45] propose the idea of whole-body vision with an optical skin system; combining embedded cameras with transparent, flexible skin. However, their implementation is constrained by high data transmission requirements and integration complexity such as bulky Raspberry Pi cameras and processors. In contrast, we fully implemented an effective and robust whole-body vision setup capable of autonomously performing dexterous manipulation skills enabled by our modular hardware design (§ IV), intuitive data collection interface (§ V), whole-body visuomotor policy (§ VI), as well as several practical considerations during system implementation (§ VII). Through our experiments, we demonstrate that the whole-body vision data is sufficient for imitation learning of complex manipulation tasks (§ VIII).

III. Design Objectives

We design RoboPanoptes to be **capable yet practical**; able to perform a wide range of whole-body dexterity tasks, while remaining easy to build, extend, and deploy.

Specifically, RoboPanoptes' **capabilities** are characterized by:

- *Motion flexibility*, enabling the system to leverage all body parts for object manipulation and collision avoidance.
- *Visual coverage*, minimizing occlusions caused by the environment or the robot's body itself.
- *Rapid adaptability* to new tasks and applications without explicit reprogramming, instead directly leveraging human demonstration.

The **practicality** of RoboPanoptes is reflected in its:

- *Robustness* to sporadic sensor delays and failures; an issue that compounds as the number of cameras increases.
- *Modular design*, allowing easy customization with varying degrees of freedom (DoFs) and camera configurations.
- *Reproducibility*, as the system is low-cost and built with off-the-shelf and 3D printed components, enabling researchers to easily replicate it.

We achieve these objectives through a series of careful hardware design decisions § IV, an intuitive data collection interface § V, and robust visuomotor policy learning § VI. In § VII, we also discuss important practical considerations for designing whole-body vision systems.

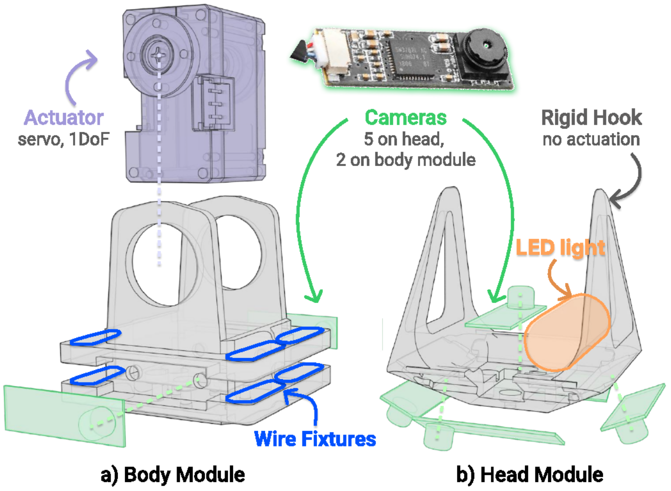

Fig. 2: **Modular Hardware Design** including a) a body module consisting of an actuator, two cameras, and wire fixtures, as well as b) a head module with five cameras and an LED light.

IV. Modular Hardware Design

RoboPanoptes' hardware consists of nine modular body units and one head unit. The design objectives achieved thereby are extendability to higher DoFs, comprehensive visual coverage, and ease of (hardware) reproducibility.

Vision-Actuation Body Module. Each body module (shown in Fig. 2 b) contains an actuator, two cameras, and fixtures for wire management. The actuator is a standard `DYNAMIXEL XC330-M288-T` servo motor with one rotational degree of freedom. This motor is small ($20 \times 34 \times 26$ mm) and lightweight (23 g), with a 288.4:1 gear ratio. These motors are daisy-chained together; a single wire powers the robot.

We control them with current-based position control communicated through the TTL protocol. The body modules are chained together, with neighboring links' DoF perpendicular to each other, as shown in Fig. 1. Each motor can freely rotate up to ±90°, allowing flexible and complex motions of the chain of body modules.

To maximize the visual coverage while minimizing the number of cameras, we mount two cameras on each link – one on the front and one on the back – with cameras pointing outwards and perpendicular to the link's DoF. Using eight body modules, the whole robot is thus equipped with 16 cameras – four cameras on each of its four sides –, enabling omnidirectional (*"all-seeing"*) vision. We design the camera slots on the links to hold the cameras mechanically stable to avoid relative movement w.r.t. the link, and caging the cameras inside the body to avoid accidental damage. We use the `Adafruit Ultra Tiny USB Camera` with a `GC0307` image sensor, characterized by an extremely small dimension ($8 \times 25 \times 4.5$ mm) that easily fits on the small links. It offers a sufficient resolution of 640×480 and a 50° field of view. However, each camera requires an adapter cable that converts the camera board's JST connector to a USB-A port, and the cameras cannot be daisy-chained. If not managed properly, the resulting large number of cables would constrain the robot's motion and occlude camera views. To address this, we design wire fixtures that are incorporated on each link. These slots allow wires to pass through on the remaining two sides of the module that are not holding cameras, ensuring that the wires will not occlude the cameras.

Head Module. The final body module is connected to a head module that acts as an end-effector to unlock more manipulation skills and provides additional perception capabilities. We design this module in the shape of a parallel hook that enables the picking and pulling of small extrusions, *e.g.*, hooking and pulling the drawer's handle in Fig. 1 b. Even though the robot's body modules observe all four sides around it, some blind angles (facing top and down) remain, which we address through the head module's camera mounting strategy. Four downwards-facing cameras are placed on the lower part of the unit, providing additional top-down views. Another camera is placed in the center of the head and pointing upwards. Moreover, a battery-powered LED light is placed next to the upward camera, illuminating the area that the head module navigates towards and that its hooks might interact with. This allows the robot to visually perceive even narrow dark spaces, *e.g.*, inside the box in Fig. 1 a. Alternatively, this head design may be flexibly swapped with other types of end-effectors for specific tasks.

V. Data Collection Interface

Directly programming such a high-DoF robot to execute complex (and natural) behaviors is difficult. Inverse kinematics and standard joystick controls, commonly used in manipulation systems, can only control end-effector motions and are hard to solve for high-DoF systems. For locomotion of similarly structured high-DoF robots, previous approaches use carefully coded motion patterns (*e.g.*, sinusoid functions [21, 40, 2]), which do not transfer to complex manipulation tasks. While RL exploration offers an alternative, it can only be safely and scalably implemented in simulation, introducing significant sim2real observation gaps.

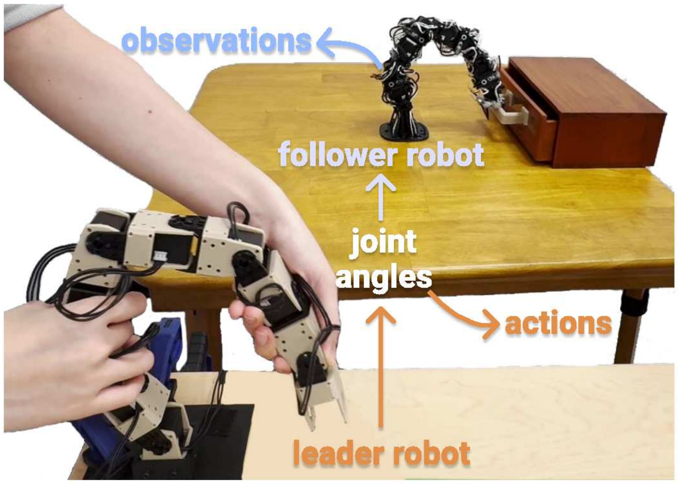

Fig. 3: **Data Collection Interface.** The operator uses both hands to control the leader robot, whose joint angles are sent to the follower robot in real-time as position targets. The joint angles of the leader robot are recorded as target actions, while the images and joint angles of the follower robot are recorded as observations.

Instead, to enable intuitive control and **adaptability** to new skills, we design a leader-follower teleoperation system [41] as shown in Fig. 3. The leader and the follower robot have the exact same structure. During teleoperation, torque is disabled for the leader robot while being enabled for the follower. To demonstrate a task, a human operator uses both hands to move the leader robot. The leader's joint positions are sent to the follower in real time, allowing it to mirror the leader using PID position control at a control rate of 30 Hz. During the demonstration, images from all cameras and robot joint positions are recorded at 10 Hz.

VI. Whole-body Visuomotor Policy

Using the collected demonstrations, we can train a whole-body visuomotor policy that infers whole-body actions (*i.e.*, nine joint angle sequences) given whole-body vision (*i.e.*, images from 21 cameras). Compared to a common manipulation system, RoboPanoptes needs to handle significantly more complex observation spaces due to the following factors:

- **Many cameras:** The policy needs to efficiently extract task-relevant semantic information from the raw pixels captured by a large number of cameras.
- **Non-stationary cameras:** As the robot moves, the cameras are in constant motion. This requires the policy to understand the dynamic spatial relations between the cameras, the robot and its environment.
- **Unreliable cameras:** A system of many cameras is prone to unpredictable failures and delays, requiring the policy to be robust to such disturbances.

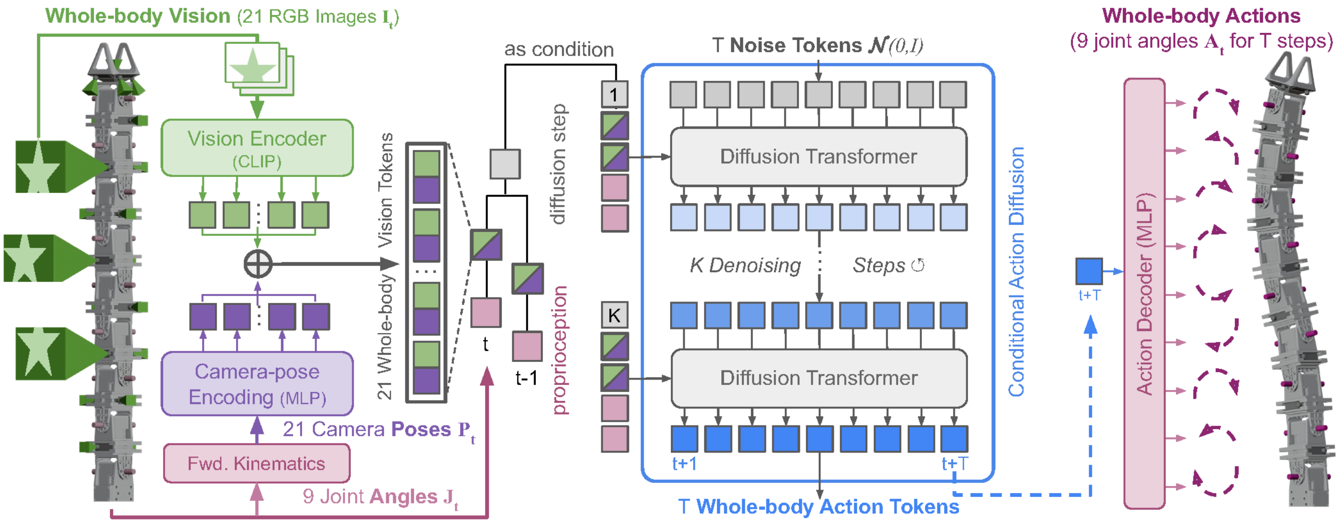

Fig. 4: **Whole-body Visuomotor Policy** leverages whole-body vision for whole-body dexterity. Left: The current robot and environment state is observed via RoboPanoptes' 21 cameras and its 9 joints angles, converted to 21 camera poses using forward kinematics. Each image (green) is represented by the class token of a vision foundation model. Each camera pose (purple) is embedded using our view-dependent positional encoding. The concatenation of each camera's image and pose tokens yields a whole-body vision token; 21 in total. Middle: Our whole-body visuomotor policy consumes these vision tokens, proprioception, and denoising-step tokens as condition via cross attention. We diffuse T whole-body dexterity tokens (blue), each corresponding to an action time step. Right: Per time step, we project the predicted dexterity token to the 9 joint angles to be achieved by the dexterity action.

As shown in Fig. 4, we base our policy design on Diffusion Policy [4], which infers the robot's joint-space actions through denoising diffusion, conditional on previous observations. Concretely, at each time step t, the policy takes the previous T_o observations $\mathbf{O}_t$ as conditional input. It predicts T_p actions $\mathbf{A}_t$, of which $T_a \leq T_p$ are executed on the robot. In our implementation, we set the observation horizon $T_o = 2$, the action prediction horizon $T_p = 16$, and the action execution horizon $T_a = 8$. The observations $\mathbf{O}_t = \{\mathbf{I}_t, \mathbf{P}_t, \mathbf{J}_t\}$, taken from the follower robot during training, consist of the previous RGB observations $\mathbf{I}_t = \{i_{t-T_o+1}, \dots, i_t\} \in \mathbb{N}_0^{T_o \times 21 \times 640 \times 480 \times 3}$ from 21 cameras, their poses $\mathbf{P}_t = \{p_{t-T_o+1}, \dots, p_t\} \in \mathbb{R}^{T_o \times 21 \times (3+6)}$, and the corresponding joint angles $\mathbf{J}_t = \{j_{t-T_o+1}, \dots, j_t\} \in \mathbb{R}^{T_o \times 9}$. The actions, taken from the leader robot during training, are the following joint angles $\mathbf{A}_t = \{a_{t+1}, \dots, a_{t+T_p}\} \in \mathbb{R}^{T_p \times 9}$.

In the following, we elaborate on the policy design details and outline how these key design decisions address the aforementioned challenges.

Coordinating moving cameras with view-dependent positional encoding. The robot's motions continuously change the camera poses and, thereby, the region of the environment each camera is observing. The observation space of this *dynamic* multi-view camera system is thus significantly larger than that of, *e.g.*, a static top-down camera. To coordinate the moving cameras and make policy learning more efficient, we employ a view-dependent positional encoding strategy. Specifically, we compute the 6D camera poses in the base frame using the forward kinematics based on the current joint angles. Each camera pose is represented by a 3-dimensional vector for position and a 6-dimensional vector for orientation, using the first two columns of rotation matrix [47]. The position and orientation vectors are each projected to a 192-dimensional vector using fully-connected linear layers. Our experiments demonstrate that this additional positional encoding enables the policy to have better data efficiency and 3D awareness.

Learning semantic correspondences using a multi-view cross-attention transformer. To efficiently extract task-relevant information from many cameras, we employ a pretrained vision encoder and a multi-view cross-attention transformer to learn the task-relevant semantic correspondence between different camera observations. To this end, we exploit pretrained vision foundation models such as CLIP [28] or DINO [3] that enable advanced semantic understanding [44] and visually-complex robot manipulation tasks [5, 43]. Specifically, we first resize the images to 224×224 resolution and apply color jitter augmentation. We batch the 21 images and feed them into the (frozen) ViT-B/16 [9] encoder of a CLIP-pretrained model to predict a 768-dimensional class-token feature for each image. These per-image features are projected to 384-dimensional vectors and concatenated with the corresponding camera's position and orientation embeddings, resulting in a 768-dimensional token per camera. The concatenation of these whole-body vision tokens with the projected joint angles yields a $(21+9) \times 768$-dimensional representation of the robot and environment state at time t.

We adopt a time-series diffusion transformer architecture [4]. Noisy actions $\mathbf{A}_t^k$ (where k is the denoising step) are passed in as input tokens for the transformer decoder blocks, while the observations $\mathbf{O}_t$ are passed into the multi-head cross-attention layers of the transformer decoder stack as conditional input. The cross-attention mechanism [38, 4] efficiently leverages the rich information captured by all cameras, learning the correspondences between multi-modal observations (multi-view images and robot proprioception) and robot actions.

Robustness to unreliable cameras with blink training. Low-cost cameras often have unreliable connections, leading to dropouts and variable latencies. On average, the dropout rate (*i.e.*, camera discount) observed for the used cameras is 4.4% and latency ranges from 15 ms to 100 ms. To make the system robust to such camera failure at test time, we employ a "blink training" strategy that randomly drops out camera inputs during training. Concretely, we simulate a 5% failure rate for each camera, *independently* masking out entire images at each time step. This means that there is a 65.9% probability that *at least one* camera is being dropped out at each time step. Consistent with observations in previous work [34], this simple strategy significantly improves the robustness of the policy, enabling it to succeed even when some cameras are completely disabled during test time.

VII. Practical Considerations

This section highlights critical implementation details for developing an effective RoboPanoptes system. Although we do not consider these aspects to be novel technical contributions, they are crucial to ensuring robust and efficient system performance. As such, our insights provided here may serve as guidelines for the future development of whole-body dexterity and vision systems.

Camera Selection. When selecting a camera, we need to consider its size, field of view (FOV), and connectivity. We choose the `Adafruit Ultra Tiny USB Camera` due to its versatility and ease of integration. It offers a compact design ($8 \times 25 \times 4.5$ mm) that fits well within the spatial constraints of the robot while providing a reasonable FOV (50°) for the tasks. Moreover, these USB cameras support standard UVC (USB Video Class) interfaces, making them plug-and-play compatible with various devices. This simplifies the deployment and ensures ease of integration.

We also explored other options, such as wireless cameras and Raspberry Pi cameras, but encountered impractical limitations. Wireless cameras introduce significant latency, limiting the reactivity of the overall system during deployment. The `Raspberry Pi Camera Module` requires a Raspberry Pi single-board computer as the interface, and we find the used ribbon cable connection to be sufficiently flexible but unstable, leading to potential reliability issues. In contrast, USB cameras provide a reliable and standardized interface and, through UVC, are compatible across a wide range of devices without the need for specialized (interface) hardware.

Wire Management. Managing the USB and power wires for all 21 cameras and 9 motors turns out to be a critical engineering challenge. It directly impacts the overall system performance as poor wire management causes a range of issues, including obstructed camera views and restricted robot motion. Tangled wires also cause physical strain and potentially lead to an unreliable, or even damaged, system. To prevent wires from obstructing camera views, we design custom fixtures on the robot's links, as described in § IV, that secure the wires in place. However, fixing the wires to the links introduces a new issue: the stiffness of the wire could reduce the range of the robot's movements. Note that the USB camera's wiring, which consists of four internal wires (power, ground, data+, and data−), requires proper insulation to prevent common-mode interference and ensure stable data transmission. However, we find that we can still increase the wires' flexibility by removing the bulky rubber jacket and retaining only the thin aluminum foil below for insulation.

Data Communication Bandwidth. A common desktop PC often has limited bandwidth, making it challenging to stream data from multiple high-bandwidth devices, such as cameras. For our choice of camera, the main constraint is the number of USB devices and corresponding video streams a PC can support; usually only four or five simultaneously. To address this bottleneck, we use four PCIe USB extension boards, each providing 20 Gbps of additional bandwidth and allowing the system to reliably stream data from 23 cameras simultaneously.

Multiprocessing. To efficiently manage data acquisition and processing from multiple cameras, we use multiprocessing to parallelize tasks and minimize interference between camera streams. Each camera is acquired in its own process at 30 FPS. During demonstration and inference, a shared memory space (with the size of $\mathbb{N}_0^{21\times640\times480\times3}$) across all processes is maintained to store the latest image observations, updated every time a new frame is received. This approach ensures seamless data flow and reduces processing delays, allowing the system to handle large amounts of visual data concurrently.

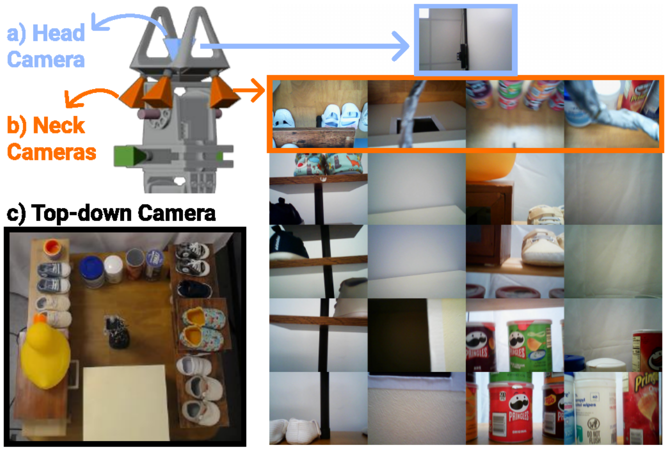

Fig. 5: **Alternative Observation Spaces.** We compare RoboPanoptes with several baselines, including using a) only the head camera, b) the four neck cameras, and c) a top-down camera. Variants using all of RoboPanoptes' cameras but without view-dependent positional encoding or without blink training serve as ablations of our design.

VIII. Experiments

In the following, we study RoboPanoptes' ability to perform a wide range of real-world manipulation tasks that require whole-body dexterity. To ensure fair comparisons, all methods use the same set of demonstrations for training and identical initial configurations during testing.

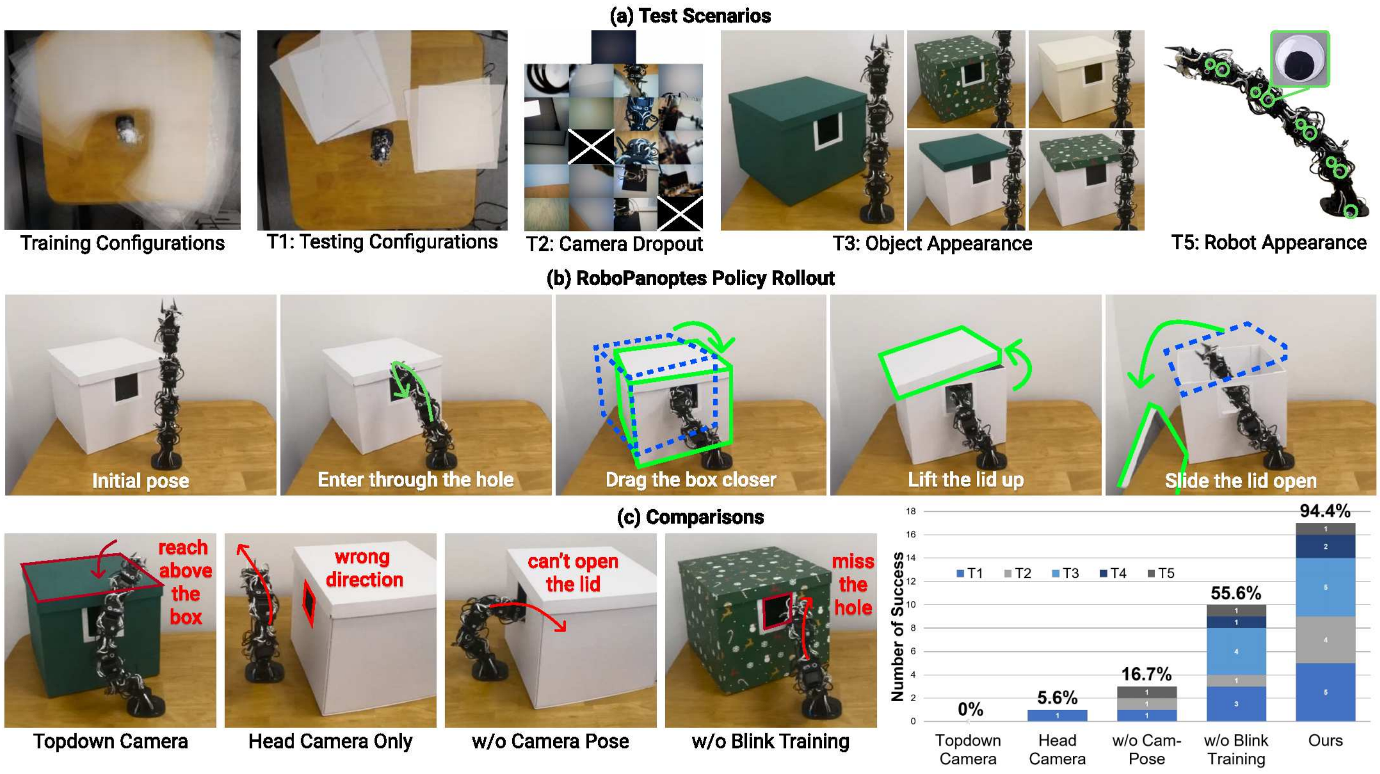

Fig. 6: **Unboxing Task.** a) Different test scenarios. The first two images show all initial configurations overlayed. b) Policy rollout of RoboPanoptes, demonstrating precise manipulation in narrow spaces (inside the box) and utilizing body contacts to drag the box closer. c) Typical failure cases of the baselines. The `Top-down Camera` policy struggles to determine the correct reaching height. The `Head Camera` policy moves in the wrong direction. The `w/o Camera Pose Encoding` policy struggles to open the lid, and at locating and entering the hole. The `w/o Blink Training` policy underperforms in scenarios with camera dropout (T2).

A. *Unboxing Task*

Task: The robot needs to open the lid of a box that is placed in a random location around the robot (Fig. 6). To do so, the robot first needs to locate the small hole on the side of the box, enter the box through the hole, drag the box closer (if it is out of reach), extend its body inside the box, lift the lid, and finally slide the lid aside to fully open the box.

The task success is measured by whether the box is opened in the end. No partial credit is given for this task.

Challenges: Constrained space: The robot must enter the box through a narrow hole and operate within the confined space inside the box. Visual occlusions: The hole's location is often occluded (with respect to a camera at the scene's center) due to the random box placement. Dexterity: Actions like lifting and sliding the lid require fine-grained control over multiple DoFs involving multiple links. Multiple contacts: The robot must drag the box closer, leveraging multiple contacts with its body links and stabilizing the box during lid manipulation.

Test scenarios: We conduct 18 rollouts for each policy. To ensure a fair comparison, we use the same set of initial robot and object configurations for all different methods. We achieve the same object configuration by adjusting it to an overlayed top-down view of the reference configuration. As shown in Fig. 6 (a), the test configurations can be grouped into five categories:

- T1: Variation of the initial object pose (5 rollouts).
- T2: Random camera dropout during execution, simulating for each camera a 5% failure rate at each time step (5 rollouts); *i.e.*, a 65.9% failure rate for at least one camera.
- T3: Generalization to boxes with different appearances, such as different colors and patterns (5 rollouts).
- T4: Introducing random camera latency, with a 10% chance per time step and delays sampled from a uniform distribution $\mathcal{U}(0, 0.5s)$ (2 rollouts).
- T5: Generalization to robot appearance changes by adding googly-eye stickers all over the robot (1 rollout).

Comparisons: We compare RoboPanoptes against the following approaches in this task:

- `Top-down Camera`: Observations from a single, static top-down camera (Fig. 5).
- `Head Camera`: Observations from a single camera mounted at the center of the head (Fig. 5).
- `w/o Camera Pose`: A whole-body visuomotor policy trained without view-dependent positional encoding.
- `w/o Blink Training`: A whole-body visuomotor policy trained without randomized camera dropouts.

Performance: The training dataset contains 147 demonstration episodes, with each demonstration averaging 15 s. We report both qualitative and quantitative results in Fig. 6, and illustrate typical failure cases. RoboPanoptes achieves an

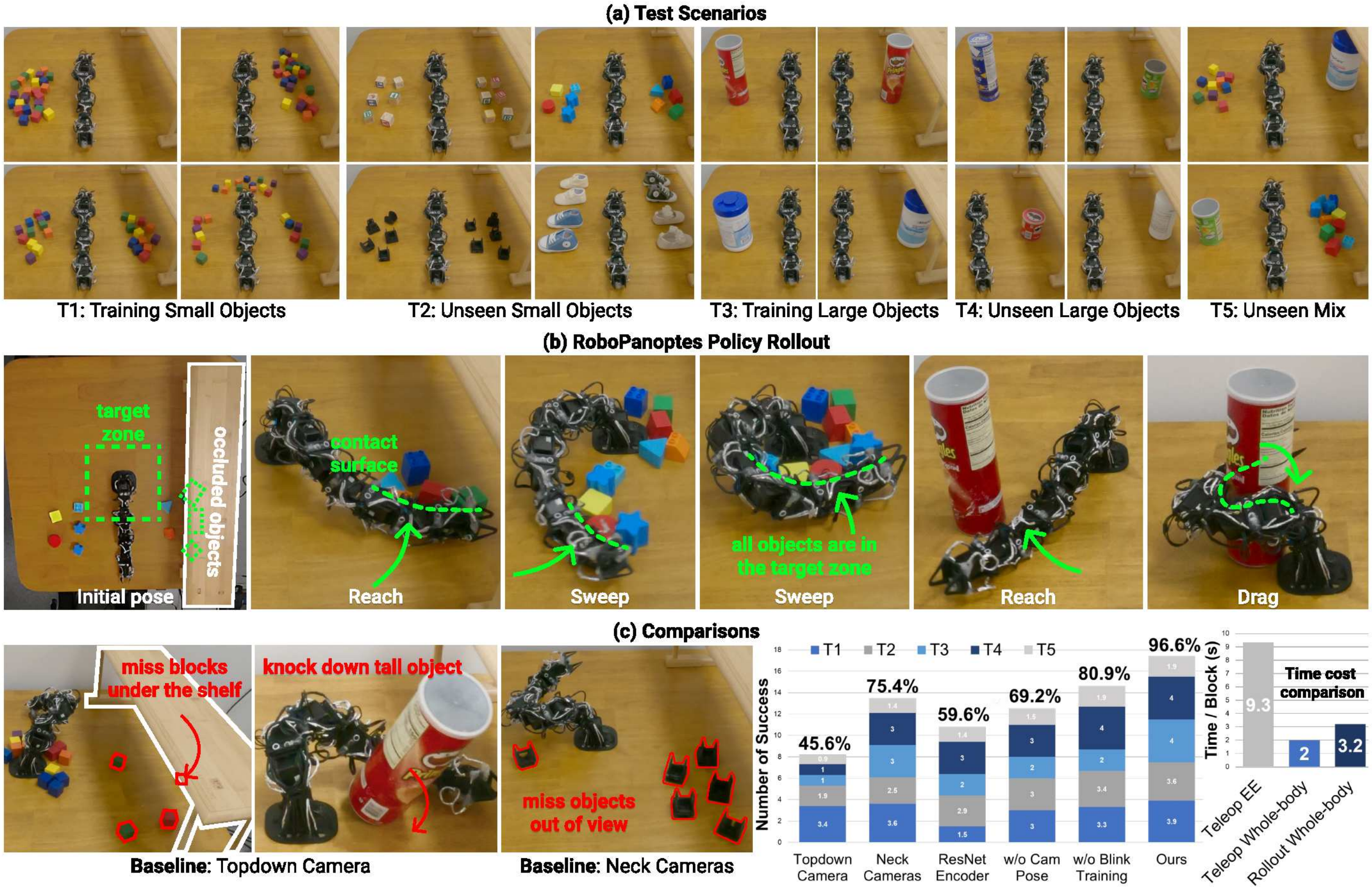

Fig. 7: **Sweeping Task.** a) Different test scenarios. b) RoboPanoptes policy rollouts, highlighting the capability of leveraging multiple whole-body contacts with objects. c) Typical failure cases of baselines. The `Top-down Camera` policy fails to detect objects under the shelf and often knocks down tall objects. The `Neck Cameras` policy struggles with objects located behind the robot due to self-occlusion.

overall 94.4% success rate, outperforming all baselines.

The `Top-down Camera` policy typically fails to locate the hole due to occlusion. While the robot correctly moves towards the box, it often approaches the hole too high or too low. Since the hole cannot be seen from the top-down camera, it is ambiguous at which height the robot should reach forward.

The `Head Camera` policy fails consistently to move towards the box. It always executes a fixed trajectory with the robot moving forward, independent of where the box is located. We hypothesize that this is because the box often cannot be seen from the head camera and, thus, the policy has overfitted to use proprioception instead. During our experiments, the policy only succeeded once, entering the box by coincidence and managing to open the lid smoothly since this action mostly relies on proprioception.

`w/o Camera Pose`, the policy is less accurate in locating and getting inside the hole. This demonstrates that our view-dependent positional encoding design enables better spatial awareness and more effective skill learning. And even if the robot has already gotten inside, it often fails to open the lid. We suspect that this is because vision tokens outnumber proprioception tokens, making it more challenging for the policy to learn how to utilize proprioception information.

`w/o Blink Training` policy performs worse in T2 when we randomly drop out cameras and is slightly less robust than ours in other scenarios as well. This proves that our blink training strategy is critical to the robustness of the policy, especially during unexpected test-time sensor failures.

B. Sweeping Task

Task: The robot needs to sweep all objects (small or large, randomly placed on a table or under a shelf) into a target region around its base. The target zone is 20×20cm and centered around the robot. Training data includes demonstrations with 24 small blocks and two cylindrical objects. The task success rate for sweeping multiple small objects is measured by the ratio of objects inside the target zone. For sweeping a large object, the task success rate is measured by whether the object is dragged into the target zone without being knocked down.

Challenges: Multiple contacts: Efficiently sweeping multiple objects requires the robot to leverage its whole body and additional inter-object contact. The robot can also sweep objects larger than itself by wrapping around them and dragging them with multiple body links. Dexterity requirements: The robot must conform to objects, sweep under narrow spaces (*i.e.* under the shelf), and lift itself to varying heights to sweep tall objects while preventing them from toppling over. Visual occlusions: Objects under the shelf or scattered around the robot are frequently occluded from certain viewpoints.

Test scenarios: We evaluate this task across 18 rollouts, divided into the following scenarios:

- T1: 24 small blocks, seen during training (4 rollouts).
- T2: Unseen small objects (4 rollouts).
- T3: A single, large training object (4 rollouts).
- T4: A single, large but unseen object (4 rollouts).
- T5: A mixture of small and large objects (2 rollouts).

Comparisons: For this task, we compare RoboPanoptes to the following approaches:

- `Top-down Camera`: Observations from a single, static top-down camera (Fig. 5 a).
- `Neck Cameras`: Observations from four cameras, mounted along the robot's neck (Fig. 5 b).
- `ResNet Encoder`: A whole-body visuomotor policy with a ResNet-34 [15] vision encoder trained from scratch instead of using a pretrained vision encoder.
- `w/o Camera Pose`: A whole-body visuomotor policy trained without view-dependent positional encoding.
- `w/o Blink Training`: A whole-body visuomotor policy trained without randomized camera dropouts.
- `Teleop EE`: Teleoperating the robot to move small objects one by one, using only the head link. This approach is used to demonstrate the inefficiency of the end-effector-only manipulation strategy, even under a near-perfect policy (*i.e.*, teleoperation).

Performance: The training dataset contains 153 demonstrations, with each demonstration averaging 23 s. We compare the methods' success rate and time per block in Fig. 7 (bottom right). RoboPanoptes achieves a 96.6% success rate, outperforming all baselines.

The `Top-down Camera` policy typically misses objects under the shelf due to occlusion; especially failing to fetch objects that were not successfully swept out on the first try. It also sweeps tall objects at a low height, toppling them over. As in the unboxing task, we hypothesize this behavior is due to the ambiguous height when observed from the top-down view.

The `Neck Cameras` policy misses out-of-view objects (*e.g.*, when objects are hidden behind the robot itself) and fails to recover in cases when, *e.g.*, the robot pushed objects too far away. Occasionally, we observe the robot getting stuck in a pose without interacting with any object. We suspect this being due to the policy overfitting to the proprioception signal due to occlusion.

The `ResNet Encoder` policy often sweeps toward incorrect or empty regions, which we hypothesize is due to inaccurate semantic understanding in the visual representation.

The `w/o Camera Pose` policy sometimes misses a few blocks and knocks down tall objects due to inaccurate motions, verifying that our view-dependent positional encoding enhances spatial reasoning capabilities.

The `w/o Blink Training` policy is less robust than ours, *e.g.*, the robot fails to retrieve blocks being pushed further away during interactions.

Efficiency: We record the time to sweep 24 blocks from (approximately) the same initial configuration to the target zone, and report the average per-block time. With whole-body sweeping, the robot manipulates object piles much more efficiently, leveraging multiple contacts (2 s/block for teleoperation and 3.2 s/block for our rollouts), as compared to using only the end-effector to move objects one by one (9.3 s/block for teleoperation).

C. Stowing Task

Task: As shown in Fig. 8, in this task, the robot performs a sequence of actions to stow shoes in a drawer. Specifically, it hooks and pulls the drawer handle to open it, picks up a pair of shoes (one by one), places the shoes inside the drawer, and pushes the drawer to close it. The take success rate is measured by whether both shoes are inside the drawer and the drawer is fully closed in the end.

Challenges: Long horizon: This task requires long-horizon sequential actions, involving multiple articulated and deformable objects. Precision: Hooking the handle and placing shoes inside the drawer requires exact manipulation. Cluttered environment: The robot must navigate constrained drawer space and avoid collisions. Also, the robot picks up the cluttered shoes one by one.

Test scenarios: We evaluate the task over 18 rollouts, divided as follows (Fig. 8 (a)):

- T1: Stowing 4 pairs of shoes, seen during training, and each pair in two different initial configurations (8 rollouts).
- T2: Stowing 4 pairs of unseen shoes (8 rollouts).
- T3: Stowing unseen objects (*i.e.*, cups) (2 rollouts).

We compare the RoboPanoptes with two systems described before, one using only a top-down camera and one without view-dependent positional encoding.

Performance: The training dataset contains 90 demonstrations, with each demonstration averaging 20 s. We report the qualitative and quantitative results in Fig. 8, and illustrate typical failure cases. RoboPanoptes achieves an overall success rate of 83.3%, compared to 27.8% for the `w/o Camera Pose` policy and 0% for the `Top-down Camera` policy (Fig. 8). Since our demonstration data contains behaviors of recovering from a sub-goal failure (*e.g.*, failed grasps), we observe that the learned policy is able to capture this kind of robust retry-and-recovery behaviors. These behaviors are best viewed in our supplementary videos.

The `Top-down Camera` policy fails at the first step of opening the drawer, missing the handle. We conjecture that this is due to the handle being occluded when the robot is bending forward, and the ambiguity of the required height when viewed from top down.

The `w/o Camera Pose` policy's actions are less precise, leading to behaviors such as missing the shoes and pulling

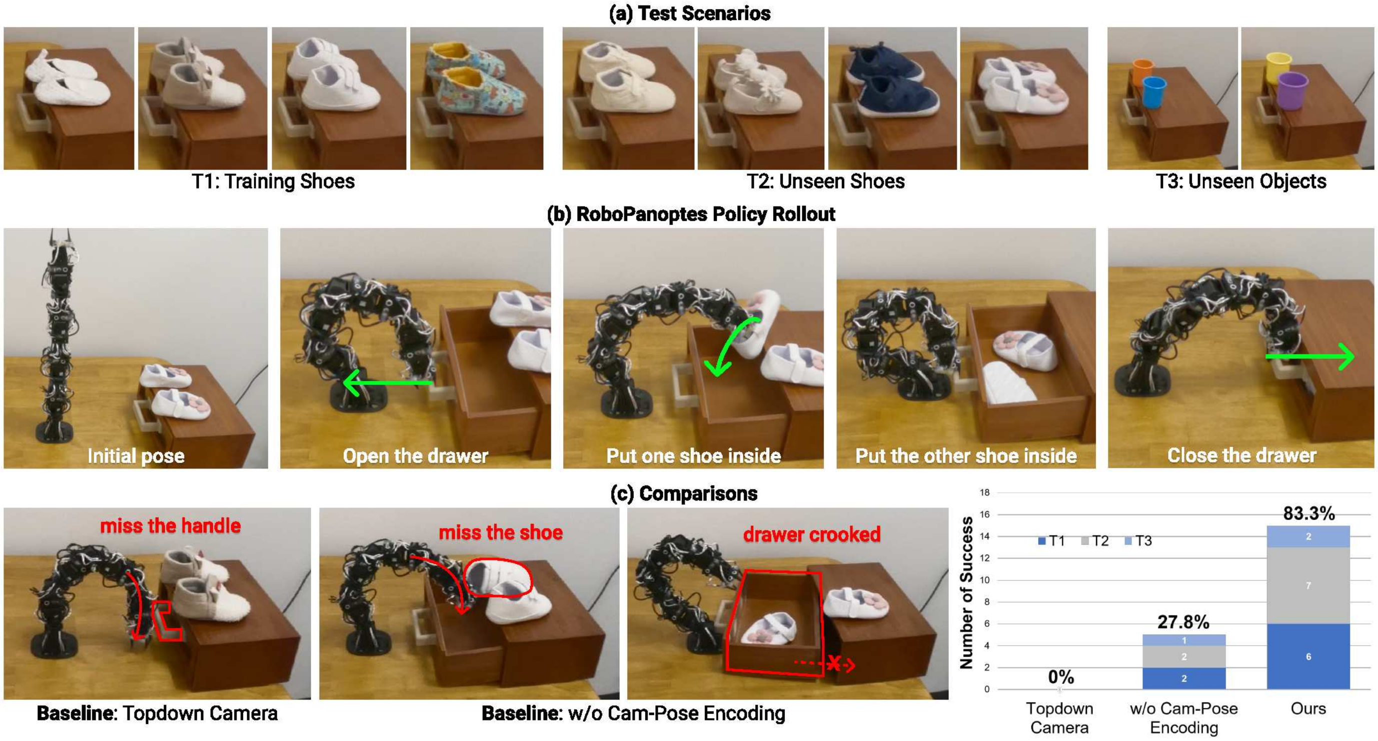

Fig. 8: **Stowing Task.** a) Different test scenarios. b) RoboPanoptes policy rollouts, demonstrating the ability of precise long-horizon manipulation in cluttered environments. c) Typical failure cases of baselines. The `Top-down Camera` policy fails to locate the handle. The `w/o Camera Pose Encoding` policy's actions are less precise, leading to failures like missing the shoe or misaligning the drawer.

the drawer out too much. This might be due to the complexity of learning from blurry observations, resulting from the constantly moving cameras. Supporting this hypothesis, we observe that the policy especially struggles when the robot is going through large whole-body motions, such as bending a large degree to open or close the drawer.

IX. Limitations and Future Work

Our system experiences three most common failure cases: 1) In the unboxing task, certain DYNAMIXEL motors automatically shut down after rapid movement due to overheating, preventing complete lid opening; 2) In sweeping, blocks occasionally move beyond the robot's reach range; 3) When stowing diverse-shaped shoes, motion imprecision can cause misalignment in cluttered spaces, preventing drawer closure. Using stronger and more precise motors could improve system performance.

In our current implementation, the robot's fixed base limits it to table-top manipulation tasks. A future direction is to extend the current setup to a mobile manipulation system, increasing the robot's operation range while enabling capabilities in challenging in-the-wild scenarios. This would require modifications to the data collection system to enable the demonstration of combined locomotion and manipulation, as well as removing the wires that tether the robot to the PC by moving to on-board computation. Future mobile systems could potentially use small-sized camera development boards with WIFI/Bluetooth that can work wirelessly, *e.g.*, `ESP32CAM`. Another improvement would be to enhance the efficiency and robustness of the computational part of the system, opening the opportunity to increase the number of sensors and their resolution.

X. Conclusion

RoboPanoptes demonstrates the effectiveness of whole-body vision and dexterity through an integrated robot system. Our system goes beyond traditional robotic manipulation by leveraging all of its body parts to sense and interact with its surroundings. Using a whole-body visuomotor policy, RoboPanoptes learns to infer complex whole-body actions from high-dimensional camera observations, while remaining robust to potential sensor failures. Extensive real-world experiments suggest that our proposed system increases robotic manipulation capability and efficiency on a wide range of challenging tasks. We hope that the presented hardware design, learning framework, and experimental results will encourage future exploration of unconventional manipulation systems such as whole-body vision and dexterity. Moreover, with all cameras available, we provide an infrastructure for studying optimal viewpoints for specific tasks by selectively disabling the camera. Since both our hardware design and input representation are modular, cameras can be easily added or removed based on the task.

Acknowledgement

The authors would like to thank Han Zhang, Ken Wang, Yifan Hou, Haoyu Xiong, Haochen Shi, Mengda Xu, Cheng Chi, Zhenjia Xu, and Adrian Wong for their advice on the design of the robot and implementation of hardware. We thank Huy Ha for his help with visualizations and video editing. In addition, we would like to thank all REALab members: Mandi Zhao, Zeyi Liu, Max Du, Hojung Choi, Austin Patel, Shuang Li, Chuer Pan, Yihuai Gao, John So, and Eric Liang for fruitful discussions. This work was supported in part by NSF Award #2143601, #2037101, and #2132519. Dominik Bauer is partially supported by the Austrian Science Fund (FWF) under project # J 4683. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

References

[1] Laura H Blumenschein, Margaret M Coad, David A Haggerty, Allison M Okamura, and Elliot W Hawkes. Design, modeling, control, and application of everting vine robots. *Frontiers in Robotics and AI*, 7:548266, 2020.

[2] Yilin Cai, Haokai Xu, Yifan Wang, Desai Chen, Wojciech Matusik, Wan Shou, and Yue Chen. Modular Self-Reconfigurable Continuum Robot for General Purpose Loco-Manipulation. *IEEE Robotics and Automation Letters*, 2025.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[5] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[6] Margaret M Coad, Laura H Blumenschein, Sadie Cutler, Javier A Reyna Zepeda, Nicholas D Naclerio, Haitham El-Hussieny, Usman Mehmood, Jee-Hwan Ryu, Elliot W Hawkes, and Allison M Okamura. Vine robots. *IEEE Robotics & Automation Magazine*, 27(3):120–132, 2019.

[7] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009.

[8] Emmanuel Dean-Leon, J Rogelio Guadarrama-Olvera, Florian Bergner, and Gordon Cheng. Whole-body active compliance control for humanoid robots with robot skin. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5404–5410. IEEE, 2019.

[9] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Belal A Elsayed, Tatsuya Takemori, Motoyasu Tanaka, and Fumitoshi Matsuno. Mobile manipulation using a snake robot in a helical gait. *IEEE/ASME Transactions on Mechatronics*, 27(5):2600–2611, 2021.

[11] 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.

[12] 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.

[13] Aimee Goncalves, Naveen Kuppuswamy, Andrew Beaulieu, Avinash Uttamchandani, Katherine M Tsui, and Alex Alspach. Punyo-1: Soft tactile-sensing upper-body robot for large object manipulation and physical human interaction. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 844–851. IEEE, 2022.

[14] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Seunghun Jeon, Moonkyu Jung, Suyoung Choi, Beomjoon Kim, and Jemin Hwangbo. Learning whole-body manipulation for quadrupedal robot. *IEEE Robotics and Automation Letters*, 9(1):699–706, 2023.

[17] Shuo Jiang and Lawson LS Wong. A Hierarchical Framework for Robot Safety using Whole-body Tactile Sensors. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8021–8028. IEEE, 2024.

[18] Shuo Jiang, Adarsh Salagame, Alireza Ramezani, and Lawson LS Wong. Snake Robot with Tactile Perception Navigates on Large-scale Challenging Terrain. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5090–5096. IEEE, 2024.

[19] Daehwa Kim, Mario Srouji, Chen Chen, and Jian Zhang. ARMOR: Egocentric Perception for Humanoid Robot Collision Avoidance and Motion Planning. *arXiv preprint arXiv:2412.00396*, 2024.

[20] Marina Kollmitz, Daniel Büscher, Tobias Schubert, and Wolfram Burgard. Whole-body sensory concept for compliant mobile robots. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5429–5435. IEEE, 2018.

[21] Pål Liljebäck, Kristin Ytterstad Pettersen, Øyvind Stavdahl, and Jan Tommy Gravdahl. A review on modelling, implementation, and control of snake robots. *Robotics and Autonomous systems*, 60(1):29–40, 2012.

[22] Jindong Liu, Yuchuang Tong, and Jinguo Liu. Review of snake robots in constrained environments. *Robotics and Autonomous Systems*, 141:103785, 2021.

[23] Jingyi Liu, Pietro Balatti, Kirsty Ellis, Denis Hadjivelichkov, Danail Stoyanov, Arash Ajoudani, and Dimitrios Kanoulas. Garbage collection and sorting with a mobile manipulator using deep learning and whole-body control. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 408–414. IEEE, 2021.

[24] Wuji Liu, Zhongliang Jing, Jianzhe Huang, Xiangming Dun, Lingfeng Qiao, Henry Leung, and Wujun Chen. An Inchworm-Snake Inspired Flexible Robotic Manipulator With Multisection SMA Actuators for Object Grasping. *IEEE Transactions on Industrial Electronics*, 70(12): 12616–12625, 2023.

[25] Yun Liu, Xiaomeng Xu, Weihang Chen, Haocheng Yuan, He Wang, Jing Xu, Rui Chen, and Li Yi. Enhancing generalizable 6d pose tracking of an in-hand object with tactile sensing. *IEEE Robotics and Automation Letters*, 2023.

[26] Keke Qi, Zhibin Song, and Jian S Dai. Safe physical human-robot interaction: A quasi whole-body sensing method based on novel laser-ranging sensor ring pairs. *Robotics and Computer-Integrated Manufacturing*, 75: 102280, 2022.

[27] Guodong Qin, Huapeng Wu, Yong Cheng, Hongtao Pan, Wenlong Zhao, Shanshuang Shi, Yuntao Song, and Aihong Ji. Adaptive trajectory control of an under-actuated snake robot. *Applied Mathematical Modelling*, 106:756–769, 2022.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[29] Xiaoqian Ren, Yueyue Liu, Yingbai Hu, and Zhijun Li. Integrated task sensing and whole body control for mobile manipulation with series elastic actuators. *IEEE Transactions on Automation Science and Engineering*, 20 (1):413–424, 2022.

[30] Adarsh Salagame, Kruthika Gangaraju, Harin Kumar Nallaguntla, Eric Sihite, Gunar Schirner, and Alireza Ramezani. Loco-Manipulation with Nonimpulsive Contact-Implicit Planning in a Slithering Robot. *arXiv preprint arXiv:2404.08174*, 2024.

[31] G Seeja, Arockia Selvakumar Arockia Doss, and V Berlin Hency. A survey on snake robot locomotion. *IEEE Access*, 10:112100–112116, 2022.

[32] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

[33] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual Dexterity for Complex Tasks. In *8th Annual Conference on Robot Learning*, 2024.

[34] Skand Skand, Bikram Pandit, Chanho Kim, Li Fuxin, and Stefan Lee. Simple Masked Training Strategies Yield Control Policies That Are Robust to Sensor Failure. In *8th Annual Conference on Robot Learning*.

[35] Olivier Stasse and Ludovic Righetti. Whole-Body Manipulation. In *Encyclopedia of Robotics*, pages 1–9. Springer Berlin Heidelberg, December 2020. doi: 10.1007/978-3-642-41610-1_187-1.

[36] Motoyasu Tanaka, Kazuyuki Kon, and Kazuo Tanaka. Range-sensor-based semiautonomous whole-body collision avoidance of a snake robot. *IEEE Transactions on Control Systems Technology*, 23(5):1927–1934, 2015.

[37] Shagun Uppal, Ananye Agarwal, Haoyu Xiong, Kenny Shaw, and Deepak Pathak. SPIN: Simultaneous Perception, Interaction and Navigation. *CVPR*, 2024.

[38] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[39] Tianyu Wang, Christopher Pierce, Velin Kojouharov, Baxi Chong, Kelimar Diaz, Hang Lu, and Daniel I Goldman. Mechanical intelligence simplifies control in terrestrial limbless locomotion. *Science Robotics*, 8(85): eadi2243, 2023.

[40] Cornell Wright, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-Rivero, Ross Hatton, and Howie Choset. Design of a modular snake robot. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2609–2614. IEEE, 2007.

[41] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.

[42] Haoyu Xiong, Russell Mendonca, Kenneth Shaw, and Deepak Pathak. Adaptive Mobile Manipulation for Articulated Objects In the Open World. *arXiv preprint arXiv:2401.14403*, 2024.

[43] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

[44] Xiaomeng Xu, Yanchao Yang, Kaichun Mo, Boxiao Pan, Li Yi, and Leonidas Guibas. Jacobinerf: Nerf shaping with mutual information gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16498–16507, 2023.

[45] Akihiko Yamaguchi and Christopher G Atkeson. Optical skin for robots: Tactile sensing and whole-body vision. In *Workshop on Tactile Sensing for Manipulation, Robotics: Science and Systems (RSS)*, volume 25, pages 133–134, 2017.

[46] Jie Zhang, You Li, Ziyun Kan, Qiufeng Yuan, Hamed Rajabi, Zhigang Wu, Haijun Peng, and Jianing Wu. A

preprogrammable continuum robot inspired by elephant trunk for dexterous manipulation. *Soft Robotics*, 10(3): 636–646, 2023.

[47] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.