ToddlerBot: Open-Source ML-Compatible Humanoid Platform for Loco-Manipulation

Haochen Shi[*] **Weizhuo Wang**[*] **Shuran Song**[†] **C. Karen Liu**[†]
Stanford University
[*]Equal contribution [†]Equal advising
https://toddlerbot.github.io

Abstract: Learning-based robotics research driven by data demands a new approach to robot hardware design—one that serves as both a platform for policy execution and a tool for embodied data collection. We introduce ToddlerBot, a low-cost, open-source humanoid robot platform designed for robotics and AI research. ToddlerBot enables seamless acquisition of high-quality simulation and real-world data. The plug-and-play zero-point calibration and transferable motor system identification ensure a high-fidelity digital twin and zero-shot sim-to-real policy transfer. A user-friendly teleoperation interface streamlines real-world data collection from human demonstrations. With its data collection ability and anthropomorphic design, ToddlerBot is ideal for whole-body loco-manipulation research. Additionally, ToddlerBot's compact size (0.56 m, 3.4 kg) ensures safe operation in real-world environments. Reproducibility is achieved with entirely 3D-printed, open-source design and off-the-shelf components, keeping the total cost under $6,000$ USD. This allows assembly and maintenance with basic technical expertise, as validated by successful independent replications of the system. We demonstrate ToddlerBot's capabilities through arm span, payload, endurance tests, loco-manipulation tasks, and a collaborative long-horizon scenario where two robots tidy a toy session together. By advancing ML-compatibility, capability, and reproducibility, ToddlerBot provides a robust and scalable platform for policy learning and execution in robotics research.

Keywords: Humanoid, Mechanisms & Design, Robot Modeling & Simulation

1 Introduction

Conventional robot design prioritizes actuator strength, sensor accuracy, and mechanical precision. While these characteristics are essential for deploying robots in real-world environments, they fall short in addressing the needs of robotics research—where affordability, rapid repairability, and full-stack ownership with no black boxes are crucial for faster innovation, easier debugging, and total system understanding. Additionally, traditional robot platforms are often not aligned with modern machine learning (ML) paradigms driven by embodied data. A robot platform compatible with ML must possess the innate ability to collect observation and action data, both in simulation and the real world, as these complementary data sources are essential for scalable policy learning.

In response to these challenges, we introduce **ToddlerBot, an open-source humanoid robot platform** for the robotics and AI research community (Figure 1), specifically developed to facilitate policy learning for locomotion and manipulation skills. ToddlerBot is designed to resemble a miniature human to minimize construction and maintenance costs and specialized expertise, lowering the barrier for researchers seeking to build self-sufficient, fully transparent humanoids free from critical reliance on third-party components. Although ToddlerBot lacks the payload capacity of full-sized humanoids, it offers more degrees of freedom in both the upper and lower body than any existing miniature humanoids—sufficient for advancing research in manipulation and locomotion.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

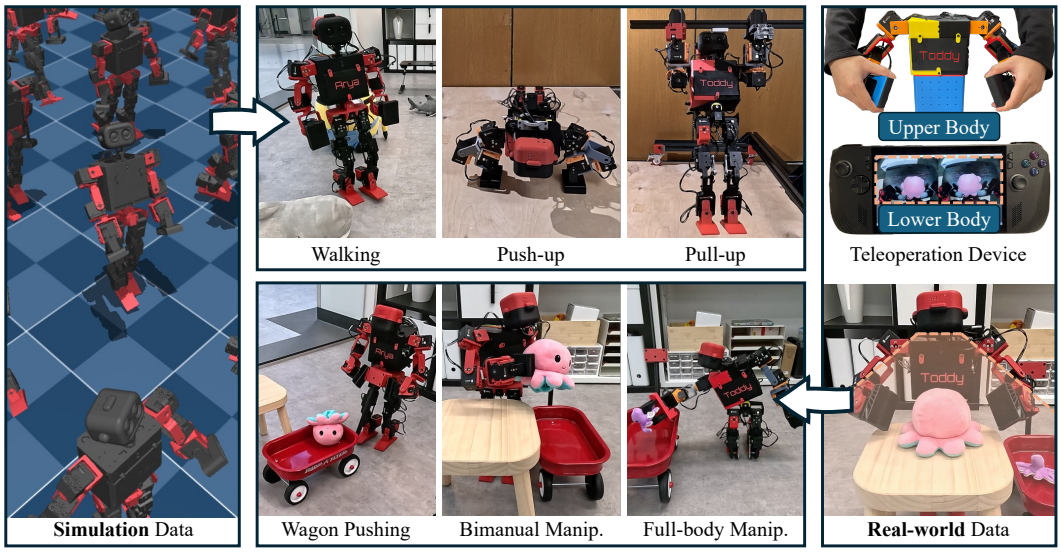


Figure 1: **ToddlerBot** is an open-source humanoid platform for large-scale, high-quality data collection in both simulation and the real world. It combines massive parallel simulation and an accurate digital twin for simulation, and an intuitive teleoperation device for whole-body control in the real world. ToddlerBot enables diverse loco-manipulation skills, including walking, push-ups, pull-ups, wagon pushing, bimanual, and full-body manipulation, learned from both data sources.

Inspired by recent quadrupeds [1, 2] and robotic manipulators [3, 4, 5, 6, 7], ToddlerBot is designed with ML-compatibility in mind. Researchers can use ToddlerBot to not only execute policies but also to serve as a robust data collection platform. It enables the acquisition of **high-quality simulation data** through a plug-and-play zero-point calibration procedure and transferable motor system identification (sysID) results. These tools ensure a high-fidelity digital twin. We validate the quality of simulation data with keyframe-interpolated motions (e.g., push-ups and pull-ups) and reinforcement learning (RL) policies (e.g., walking), demonstrating the capability for zero-shot sim-to-real transfer. ToddlerBot can also acquire **scalable real-world data**. We design an intuitive teleoperation interface that allows simultaneous control of ToddlerBot's upper and lower body to collect whole-body manipulation data and develop effective visuomotor policies. Additionally, ToddlerBot's small size and weight (0.56 m, 3.4 kg) ensure safe and accessible operation in real-world environments.

Beyond ML-compatibility, ToddlerBot is designed with a focus on capability and reproducibility. A humanoid has the potential to utilize large scale human demonstrations by leveraging its anatomical similarity to the human body. As such, ToddlerBot features an anthropomorphic design with 30 active degrees of freedom (excluding end effectors). We demonstrate ToddlerBot's capabilities through arm span, payload, endurance tests, and various open-loop and closed-loop loco-manipulation tasks.

Reproducibility is achieved through low-cost, open-source designs and readily accessible hardware components. ToddlerBot uses commercially available motors and is completely 3D-printed, with a total cost under $6,000$ USD (90% of the cost is for motors and computers). We release digital twin, learning algorithms, hardware designs, and comprehensive tutorials to ensure that ToddlerBot can be built at home with basic knowledge of hardware and software, without requiring specialized equipment for manufacturing or repair. To validate reproducibility, we enlisted a CS-major student not involved in this project to independently build another instance of ToddlerBot. and successfully zero-shot transfer loco-manipulation policies between the two instances. We also received five reports of independent replications of ToddlerBot within a week by teams worldwide.

We summarize our contributions as follows: **(1)** As a system paper, we integrate existing components to create the first miniature humanoid with 30 DoFs and a superhuman range of motion, offering a practical humanoid platform with new capabilities for the community. **(2)** We develop a comprehensive sysID pipeline to ensure a high-fidelity digital twin in simulation. **(3)** We design a whole-body teleoperation solution to collect loco-manipulation data in the real world.

Table 1: Comparison with Other Popular Humanoid Research Platforms.

Humanoid	Size (m)	Weight (kg)	Compute (TFLOPS)[a]	Active DoFs[b] -	Manipulation -	Locomotion -	Sim Data -	Real Data -	Open Source -	Price ($)
BD Atlas [8]	1.50	89.0	-	28	✓	✓	-	-	✗	-
Figure [9]	1.68	70.0	-	26	✓	✓	-	-	✗	-
Optimus [10]	1.73	57.0	-	28	✓	✓	-	-	✗	-
Digit [11]	1.75	65.0	-	16	✓	✓	✓	✗	code	250K
Unitree H1 [12]	1.76	47.0	1.92	19	✓	✓	✓	✓	code	70K
Fourier GR1 [13]	1.65	55.0	2.23	32	✓	✓	-	-	✗	110K
Booster T1 [14]	1.18	30.0	3.33	23	✓	✓	✓	✗	code	34K
Unitree G1 [15]	1.32	35.0	2.50	29	✓	✓	✓	✓	code	57K
iCub [16]	1.04	24.0	1.93	32	✓	✓	✓	✗	code	300K
Berkeley [17]	0.85	16.0	1.92	12	✗	✓	✓	✗	code	10K
MIT [18]	1.04	24.0	-	18	✗	✓	-	-	✗	-
Duke [19]	1.00	30.0	Optional	10	✗	✓	✓	✗	design, code	16K
Berkeley Lite [20]	0.80	16.0	0.29	22	✓	✓	✓	✓	design, code	5K
BRUCE [21]	0.70	4.8	0.1	16	✗	✓	✓	✗	code	6.5K
NAO H25 [22]	0.57	5.2	0.02	23	✓	✓	✓	✗	code	14K
Robotis OP3 [23]	0.51	3.5	0.30	20	✗	✓	✓	✗	code	11K
Zeroth [24]	0.48	3.6	0.01	16	✓	✓	✓	✗	design, code	1.4K
Ours	0.56	3.4	2.50	30	✓	✓	✓	✓	design, code	6K
Average Adult [25]	1.73	70.9	-	32[c]	✓	✓	-	-	-	-

[a] TFLOPS refers to Tera FLoating-point Operations Per Second with single precision (FP32).
[b] The active degrees of freedom actuated by motors, excluding end effectors such as parallel grippers or dexterous hands.
[c] While human body is powered by over 600 muscles, the primary functional movements can be approximated using 32 revolute joints: six DoFs per leg, seven DoFs per arm, three DoFs for the waist, and three DoFs for the neck, excluding fingers and toes.

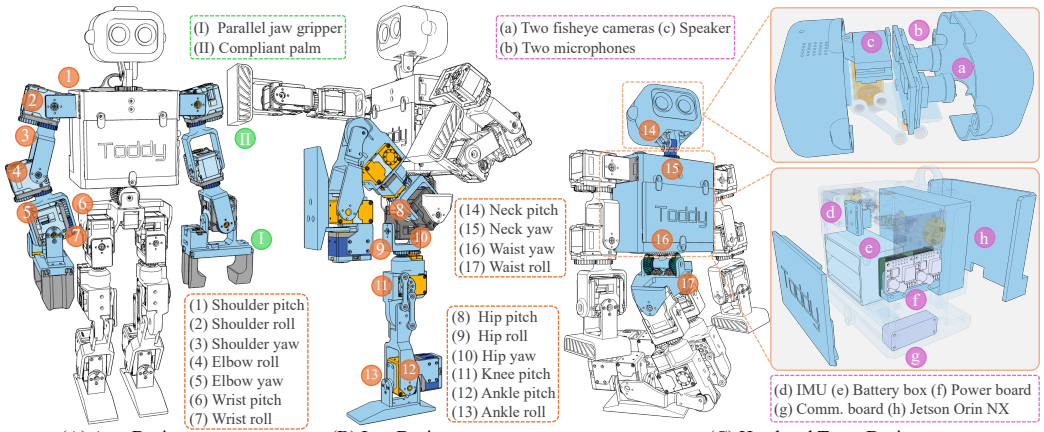
2 Related Works

In recent years, numerous humanoid robots have been developed. Full size industrial humanoids such as Boston Dynamics Atlas [8], Cassie [26], Digit [11], Figure [9], Tesla Optimus [10], Unitree H1 [12] and Fourier GR1 [13]. Half-scaled Booster T1 [14], Berkeley Humanoid Lite [20] and Unitree G1 [15]. Miniature-sized NAO H25 [22], Robotis OP3 [23], and K-Scale Zeroth [24]. Together, they showcase diverse designs and capabilities. On the other hand, humanoids from research institutions, including the Berkeley Humanoid [17], BRUCE [21], Duke Humanoid [19], iCub [16], and MIT Humanoid [18], also explore the humanoid design space with different optimization emphasis. As outlined in Table 1, we identify nine metrics to compare different humanoid platforms.

Size and weight are critical when designing humanoid robots. For manipulation, full-size humanoids are compatible with human-scale objects, whereas reduced-size robots are limited in their ability to manipulate them. However, reduced-size humanoids can still benefit manipulation research when paired with appropriately scaled objects [20]. Furthermore, dynamic whole-body control techniques are likely transferable to larger humanoids. For locomotion, full-sized and small-scale humanoids offer similar research values [27]. Full-size humanoids typically require a substantial engineering team to operate and maintain, along with specialized facilities like gantry cranes for safety. In contrast, a smaller humanoid is inherently cheaper, easier to build and repair, and safer [28, 27, 29]. They can be deployed by a single person and operated in constrained environments with a laptop.

The number of active DoFs is crucial for human-like motion. While the human musculoskeletal system employs over 600 muscles, primary functional movement can be approximated using 32 revolute joints [16, 30]: six DoFs per leg, seven DoFs per arm, three DoFs for the waist, and three DoFs for the neck, excluding fingers and toes. Therefore, humanoid designs aim to achieve a DoF count as close to 32 as possible. The public perception of the limited performance of miniature humanoids is primarily due to fewer DoFs. This limitation often arises from space constraints that restrict the incorporation of many DoFs, a challenge we have successfully addressed in ToddlerBot.

To qualitatively assess the capability, we evaluate the humanoid's ability to perform both manipulation and locomotion tasks. The combination of both is particularly compelling, as it unlocks opportunities for whole-body control research [31, 32, 33, 34, 35]. Furthermore, certain motions, such as push-ups, pull-ups, and cartwheels, require coordinated use of both arms and legs. Apart from physical capabilities, ToddlerBot also has best compute in class. Onboard CUDA accelerator allows concurrent policy inferences to support visual and locomotion capabilities. Large-scale simulation data collection is effective for locomotion [36, 37, 38], while real-world data collection



(A) Arm Design (B) Leg Design (C) Head and Torso Design

Figure 2: **Mechatronic Design.** Orange markers highlights ToddlerBot's 30 active DoFs: 7 per arm, 6 per leg, a 2 on neck, and a 2 on waist. Green markers indicate two end-effector designs—a compliant palm and a parallel-jaw gripper. Purple markers denote the electronics layout with exploded views, featuring 2 fisheye cameras, 1 speaker, 2 microphones, 1 IMU, and 1 Jetson Orin computer.

is more promising for manipulation [39, 40, 41]. Therefore, to be **ML-compatible**, a humanoid research platform should facilitate data collection in both world.

Moreover, being open-source and low-cost is essential for others to reproduce. Without these qualities, research in this field would remain restricted to those with specialized expertise and significant resources. While making no compromise in functionality, ToddlerBot stands out as completely open-source and the most accessible among recent humanoid research platforms.

3 System Design

The design space of humanoid robots is vast; ToddlerBot's key design principles prioritize reproducibility as a hard constraint, and capability and ML-compatibility as key design objectives.

3.1 Reproducibility - A Hard Constraint

Unlike most prior works, we treat reproducibility as a hard constraint, as our humanoid platform holds no value if others cannot reproduce it. We define reproducibility as the ability of a single person to replicate the robot system at home without specialized equipment. 3D printing has become a popular method for reproducing open-source hardware systems due to its accessibility and fast turnaround time [2, 5, 4]. Despite these merits, it's challenging to ensure sufficient strength and precision for printed plastic parts. We discuss our justifications for 3D printing in Appendix 8.1. For performance-critical components like motors and bearings, we limit ourselves to off-the-shelf parts. The total BOM cost of ToddlerBot is $6,000$ USD, with 90% spent on the computer (NVIDIA Jetson) and motors (Robotis Dynamixel). We detail our design considerations in Appendix 8.2.

3.2 Capability - Design Objective

Mechatronic Design. As shown in Figure 2, ToddlerBot replicates the structure of an adult human body to maximize the number of active DoFs, enabling human-like motion and versatile loco-manipulation tasks. Each arm features seven DoFs with spur gears for axis-aligned transmission, ensuring high functionality and reachability. Each leg has six DoFs: three at the hip for a wide range of motion and optimized walking strides, a parallel linkage at the knee to reduce inertia, and two at the ankle for stable locomotion. The neck, with two DoFs, incorporates parallel linkages at the pitch joint for a compact design, allowing expressive motion and full head mobility. The waist, also with two DoFs, uses coupled bevel gears to balance the space budget and effectively transmit power from two motors for yaw and roll actuation, which enables whole-body control. Additionally, ToddlerBot

offers two end-effector designs: a parallel jaw gripper for grasping and a compliant palm for tasks requiring a palm-like posture. The end-effector designs can be switched quickly within two minutes. For each active DoF, we maximize the range of motion by optimizing geometries to prevent self-collisions (Appendix 8.3). To address concerns such as space constraints, axis alignment, and inertia reduction, we integrated a combination of three transmission mechanisms in the design: spur gears, coupled bevel gears, and parallel linkages (Appendix 8.4).

Sensors, Compute, and Power. To enhance ToddlerBot's capabilities, we integrate a comprehensive set of sensors and computational components. Two fisheye cameras are included to expand the field of view. An Inertial Measurement Unit (IMU) is placed in the chest to provide state feedback. A speaker and microphones facilitate communication with humans and other robots. We provide a conversation example between two ToddlerBot instances in the supplementary video. The onboard computation is powered by a Jetson Orin NX 16GB, enabling real-time inference of ML models. Based on our proposed power metric (Appendix 8.5), we choose Dynamixel motors for their reliability and accessibility (Appendix 8.6). Power management is handled by a custom-designed power distribution board which can be easily ordered online (Appendix 8.7).

3.3 ML-Compatibility - Design Objective

Digital Twin. A high-fidelity digital twin is fundamental for high-quality simulation data collection and zero-shot sim-to-real transfer. We divide the digital twin development into two key components: zero-point calibration for correct kinematics and motor system identification for accurate dynamics. Since Dynamixel motors lack an absolute zero point, a reliable method is needed to calibrate after reassembly. We designed a set of calibration devices that can align the robot to the desired zero point within a minute, defined as standing with both arms parallel to the body (Appendix 8.8). For accurate sysID, we designed a motor test bed and automatic test procedures described in Appendix 8.9 to measure the range of dynamics parameters. Next, we collect actuator position tracking data by commanding a chirp signal [27], and use the position tracking result to fit an actuation model [42] with optimization, as detailed in Appendix 8.10. Testing across instances shows Dynamixel motors of the same model have nearly identical dynamics parameters. This is also validated by successful direct deployment of policies to a second ToddlerBot instance without additional sysID.

Teleoperation Device. Inspired by previous works [4, 43], we develop a second upper body of ToddlerBot as the leader arms to collect high-quality real-world data. Two force-sensitive resistors (FSRs) are embedded in the gripping area of the end effectors to detect compression force from the operator, allowing gripper movement based on force input. We use a handheld gaming computer (either Steam Deck or ROG Ally X) to control the other body parts. The joysticks send velocity commands to walk, turn, and squat. Buttons trigger either programmed or trained policies and provide direct control over neck and waist movements. The detailed mapping is described in Appendix 8.11.

4 System Control

4.1 Keyframe Animation

Keyframe animation is widely used in character animation, but it provides only kinematics data, lacking a guarantee of dynamic feasibility [44]. To address this, we develop a tool integrating MuJoCo [45] with a GUI, enabling real-time tuning and validation of keyframe motion trajectories. Combined with high-fidelity digital twin, we can efficiently generate open-loop trajectories such as cuddling, push-ups, and pull-ups that can be executed zero-shot in the real world.

4.2 Reinforcement Learning

For walking, we train a reinforcement learning (RL) policy, $\pi(\mathbf{a}_t|\mathbf{s}_t)$, which outputs $\mathbf{a}_t$ as joint position setpoints for proportional-derivative (PD) controllers, based on the observable state $\mathbf{s}_t$:

$$\mathbf{s}_t = (\boldsymbol{\phi}_t, \mathbf{c}_t, \Delta\mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, \boldsymbol{\theta}_t, \boldsymbol{\omega}_t)\,, \tag{1}$$

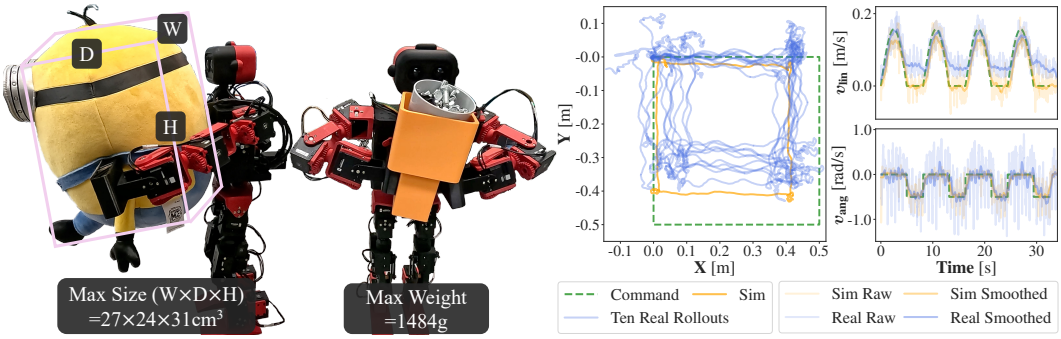


Figure 3: **Arm Span, Payload, and Trajectory Tracking.** On the left, we show that with a torso dimension of $13 \times 9 \times 12$ cm^3, ToddlerBot can grasp objects up to $27 \times 24 \times 31$ cm^3, about 14 times the torso size. Additionally, ToddlerBot can lift weights up to 1484 g, which is 40% of its body weight (3484 g). On the right, we present ten consecutive real-world rollouts of an RL walking policy tracking a square trajectory with a predefined velocity profile. Both raw and smoothed linear and angular velocity tracking are displayed, with real-world results averaged across trials.

where ϕ_t is a phase signal, $\boldsymbol{c}_t$ represents velocity commands, $\Delta \boldsymbol{q}_t$ denotes the position offset relative to the neutral pose $\boldsymbol{q}_0$, $\boldsymbol{a}_{t-1}$ is the action from the previous time step, $\boldsymbol{\theta}_t$ represents the torso orientation, and $\boldsymbol{\omega}_t$ is the torso's angular velocity. During PPO policy training [46], the environment generates the next state, $\boldsymbol{s}_{t+1}$, updates the phase signal, and returns a scalar reward $\mathrm{r}_t = \mathrm{r}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, \boldsymbol{\phi}_t, \boldsymbol{c}_t)$. Following standard practice [42], the reward is decomposed as:

$$\mathrm{r}_t = \mathrm{r}_t^{\text{imitation}} + \mathrm{r}_t^{\text{regularization}} + \mathrm{r}_t^{\text{survival}}. \tag{2}$$

Among these components, $\mathrm{r}_t^{\text{imitation}}$ encourages accurate imitation of the reference walking motion, which is generated using a closed-form ZMP (Zero Moment Point) solution [47]. $\mathrm{r}_t^{\text{regularization}}$ incorporates heuristics of ideal walking motion, penalizes joint torques, and promotes smooth actions to minimize unnecessary movements. A survival reward $\mathrm{r}_t^{\text{survival}}$ prevents early episode termination during training. Additional details are provided in Appendix 8.12.

4.3 Imitation Learning

Real-world data collection involves a human operator guiding the leader's arms to teleoperate the follower's arms while using a joystick and buttons on a handheld game controller to control body movements. During data collection, when the upper body tracks the position commands from the leader arms, ToddlerBot's lower body employs a two-layer PD controller to actively maintain balance. The first layer is a Center of Mass (CoM) PD controller, which keeps the CoM close to the center of the support polygon. The second layer is a torso pitch PD controller, which uses IMU readings to ensure the torso remains upright. The first layer addresses CoM shifts caused by arm movements, while the second layer compensates when lifting heavy objects. With this setup, we can collect 60 trajectories in just 20 minutes for both bimanual and full-body manipulation tasks. The motor positions of the leader's arms are recorded as the actions, while the motor positions of the follower robot, along with the RGB images captured from its camera, are recorded as observations. We train a diffusion policy [48] on this dataset, with further details provided in Appendix 8.13.

5 Experiments

Capability: Arm Span, Payload, and Endurance. To evaluate ToddlerBot's arm span, we teleoperate it to hug a large object using the compliant palm gripper while maintaining balance. We show that, ToddlerBot can grasp objects up to $27 \times 24 \times 31$ cm^3, approximately 14 times its torso volume.

The payload test assesses both the upper body's lifting capacity and the lower body's ability to maintain balance. ToddlerBot successfully lifts up to 1484 g, 40% of its total weight (3484 g). To

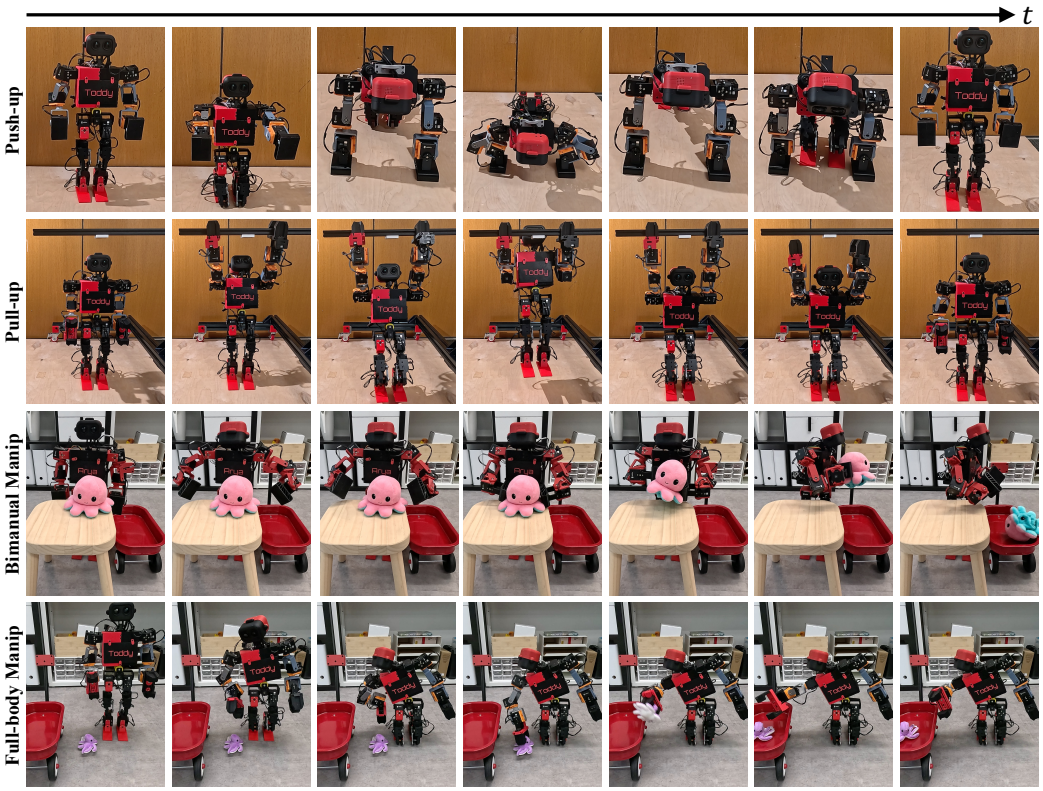


Figure 4: **Experiment Results.** We present four different tasks: push-up, pull-up, bimanual, and full-body manipulation, showing ToddlerBot's capability in challenging loco-manipulation tasks.

eliminate friction effects, we use a 3D-printed cup for the gripper to lock in securely. We incrementally add screws to the cup until ToddlerBot falls over. Results are shown on the left of Figure 3.

In the endurance test, ToddlerBot starts with a fully charged battery, running the walking RL policy while stepping in place. ToddlerBot achieved the longest streak of 19 minutes without falling. Over time, increased motor temperatures gradually pushed it outside the policy's training distribution, leading to more frequent falls. ToddlerBot withstands up to 7 falls before breaking, but even then, repairs are quick—requiring only 21 minutes of 3D printing and 14 minutes of assembly, including removing the damaged part, installing the replacement, and performing zero-point calibration.

Capability: Push-ups and Pull-ups. To demonstrate expressive and dynamic motions, we program push-ups and pull-ups in our keyframe software and perform zero-shot sim-to-real transfer (Figure 4). For pull-ups, we use an AprilTag to help ToddlerBot locate the horizontal bar. Both tasks require strong limbs, balanced upper-lower body strength, and precise coordination, particularly when ToddlerBot transitions from a planking pose to standing after push-ups and when it releases the horizontal bar and lands after pull-ups. These open-loop transfers require only a single motion trajectory designed in simulation, highlighting the fidelity of our digital twin.

ML-Compatibility: Omnidirectional Walking. To demonstrate ToddlerBot's locomotion capabilities, we train RL walking policies to follow a square trajectory with a predefined velocity profile. The right part of Figure 3 presents the results, with real-world tracking data collected via motion capture. Due to RL policy limitations, both simulation and real-world tracking deviate from the command, primarily because the learned walking policy struggles with in-place rotation, causing translation offsets. However, the sim-to-real gap is notably smaller than the tracking gap, supporting a successful zero-shot transfer. Additionally, we report a position tracking error variance of $0.018\ \mathrm{m}$, which demonstrates good repeatability (more details in Appendix 8.14).

ML-Compatibility: Vision-based Manipulation. We show ToddlerBot's ability to perform bimanual manipulation and full-body manipulation by transferring octopus toys from a table and the

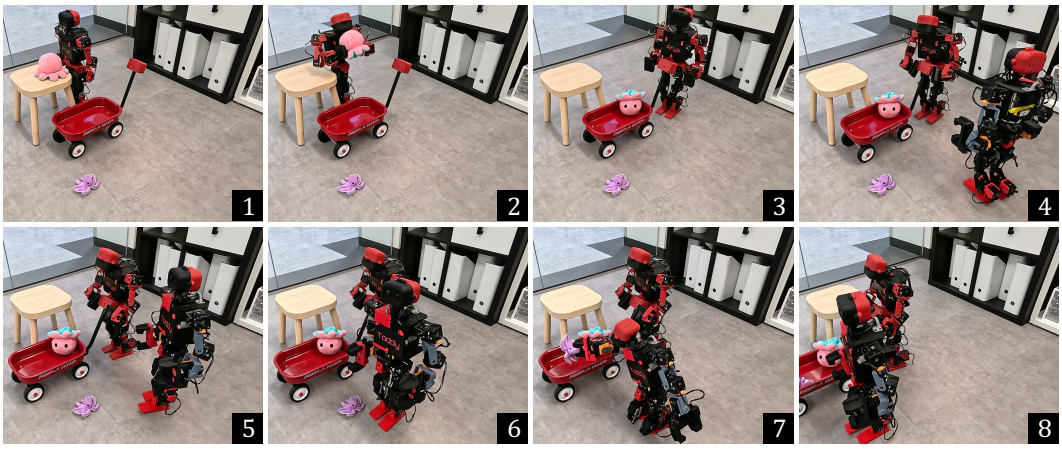


Figure 5: **Long-horizon Collaboration**. In this task, two instances of ToddlerBot, Arya and Toddy, collaborate to clean up a toy session. (1) The task begins with a pink octopus on the table and a purple octopus on the ground. (2) Arya picks up the pink octopus from the table and places it in the wagon. (3) Arya walks to the wagon handle. (4) Arya grasps the handle while Toddy walks over. (5) Arya pushes the wagon toward the purple octopus. (6) Toddy reaches the pickup position. (7) Toddy kneels and picks up the purple octopus. (8) Finally, Arya and Toddy leave side by side.

ground to a wagon. Both tasks are trained with RGB-based diffusion policy [48] with 60 demonstrations. Across 20 test trials, we achieve a 90% success rate for bimanual manipulation and 75% for full-body manipulation. We leverage a combination of open-loop motions and closed-loop policies to enhance data collection efficiency. In the bimanual task, the torso rotating and releasing motions are open-loop, while in full-body manipulation, kneeling down is open-loop. ToddlerBot's onboard computing runs a 300M parameter diffusion policy with about 100 ms latency, enabling real-time operation. Results are shown in Figure 4.

ML-Compatibility: Skill Chaining We test ToddlerBot's ability to combine loco-manipulation skills with wagon pushing. The results are shown in Appendix 8.15. To push the wagon, ToddlerBot first executes a diffusion policy to grasp the handle, while maintaining that pose, switches to the RL policy to walk forward. To enable walking while maintaining the grip, we sample the robot's end pose from the grasping policy's training data (60 demonstrations) during RL training.

Reproducibility: Hardware and Policies To demonstrate hardware reproducibility, we recruit a CS-major student with no prior hardware experience to assemble a second ToddlerBot using our open-source assembly manual and videos. The student independently completes the assembly in three days, including the time for 3D printing. The open-source community has also reported successful reproductions of ToddlerBot, and most of them were completed in one week (Appendix 8.16). For policy reproducibility, we run the manipulation policy trained on data collected with one instance on the other instance, achieving the same success rate of 90% across 20 trials. We also successfully transferred the RL walking policy between both robots. To further showcase the equivalent performance of both ToddlerBot instances, we have them collaborate on a long-horizon toy tidy-up session, as shown in Figure 5.

6 Conclusions

In conclusion, we demonstrate that ToddlerBot is ML-compatible, capable, and reproducible through a series of tests and loco-manipulation tasks. ToddlerBot extends beyond locomotion and supports full-body manipulation, character animation, human-robot interaction, and a range of ML applications, making it a versatile research platform. To bridge the sim-to-real gap, we develop a comprehensive sysID pipeline to create a high-fidelity digital twin, alongside a whole-body teleoperation system to collect loco-manipulation data. Being fully open-source, ToddlerBot empowers researchers to explore new research directions and fosters open collaboration across the community.

7 Limitations and Future Work

ToddlerBot's performance in more agile tasks is constrained by the off-the-shelf motors' max speed, max torque, and communication speed. Should there be stronger motor choices, the performance can still improve. Rather than achieving superhuman capabilities, ToddlerBot aligns more closely with average human performance in loco-manipulation tasks. In addition, the current actuation model does not consider motor temperature and is less accurate when close to the performance limit. The miniature scale limits its interaction with human-sized objects, though this does not hinder research if appropriately sized objects are used. Compared to metal shells, 3D-printed parts are more likely to break after the event of impact. Despite 3D printing being fast, repairs still require time.

To overcome these limitations, we plan to develop customized communication boards to increase control frequency and improve our actuation model to maximize usable envelope, especially when the motor heats up. We will refine ToddlerBot's design to improve structural strength, as we believe that design has a greater impact than material choice. We also aim to improve sensing capabilities, including stereo vision for depth perception, additional IMUs for improved state estimation, and tactile sensors for richer manipulation feedback.

Acknowledgments

The authors would like to express their gratitude to Kaizhe Hu for assembling the second instance of ToddlerBot and assisting with keyframe animation and demo recording. We also extend our thanks to Huy Ha, Yen-Jen Wang, Pei Xu, and Yifan Hou for their insightful discussions on locomotion, and to Sirui Chen, Chen Wang, and Yunfan Jiang for valuable input on manipulation policy deployment. We are grateful to Albert Wu for his guidance on mathematical formulation and notation. Additionally, we thank João Pedro Araújo for his assistance with the motion capture system. Finally, we appreciate the helpful discussions from all members of TML and REALab. This work was supported by National Science Foundation NSF-FRR-2153854, NSF-2143601, NSF-2037101, NSF-2132519, Sloan Fellowship, Stanford Institute for Human-Centered Artificial Intelligence, and Stanford Wu Tsai Human Performance Alliance.

References

[1] B. Katz, J. D. Carlo, and S. Kim. Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6295–6301, May 2019. doi:10.1109/ICRA.2019.8793865.

[2] N. Kau. Stanford Pupper: A Low-Cost Agile Quadruped Robot for Benchmarking and Education, Feb. 2022.

[3] T. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2. doi:10.15607/RSS.2023.XIX.016.

[4] P. Wu, F. Shentu, X. Lin, and P. Abbeel. GELLO: A General, Low-Cost, and Intuitive Teleoperation Framework for Robot Manipulators. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, Oct. 2023.

[5] K. Shaw, A. Agarwal, and D. Pathak. LEAP Hand: Low-Cost, Efficient, and Anthropomorphic Hand for Robot Learning. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2. doi:10.15607/RSS.2023.XIX.089.

[6] B. Romero, H.-S. Fang, P. Agrawal, and E. Adelson. EyeSight Hand: Design of a Fully-Actuated Dexterous Robot Hand with Integrated Vision-Based Tactile Sensors and Compliant Actuation, Aug. 2024.

[7] R. Bhirangi, A. DeFranco, J. Adkins, C. Majidi, A. Gupta, T. Hellebrekers, and V. Kumar. All the Feels: A Dexterous Hand With Large-Area Tactile Sensing. *IEEE Robotics and Automation Letters*, 8(12):8311–8318, Dec. 2023. ISSN 2377-3766. doi:10.1109/LRA.2023.3327619.

[8] B. Dynamics. Atlas, 2024.

[9] FigureAI. Figure, 2024.

[10] Tesla. AI & Robotics, 2024.

[11] A. Robotics. Agility Robotics, 2024.

[12] Unitree. Unitree H1 - Unitree's First Universal Humanoid Robot - Unitree, 2024.

[13] FOURIER-Robotics. FOURIER-Robotics, 2024.

[14] B. Robotics. Booster, 2024.

[15] Unitree. Unitree G1 - Humanoid Agent AI Avatar - Unitree, 2024.

[16] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. S. Victor, F. Becchi, G. Sandini, and G. Metta. THE DESIGN OF THE iCub HUMANOID ROBOT. *International Journal of Humanoid Robotics*, 09(04):1250027, Dec. 2012. ISSN 0219-8436. doi:10.1142/S0219843612500272.

[17] Q. Liao, B. Zhang, X. Huang, X. Huang, Z. Li, and K. Sreenath. Berkeley Humanoid: A Research Platform for Learning-based Control, July 2024.

[18] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim. The MIT Humanoid Robot: Design, Motion Planning, and Control For Acrobatic Behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8, Munich, Germany, July 2021. IEEE. ISBN 978-1-72819-372-4. doi:10.1109/HUMANOIDS47582.2021.9555782.

[19] B. Xia, B. Li, J. Lee, M. Scutari, and B. Chen. The Duke Humanoid: Design and Control For Energy Efficient Bipedal Locomotion Using Passive Dynamics, Sept. 2024.

[20] Y. Chi, Q. Liao, J. Long, X. Huang, S. Shao, B. Nikolic, Z. Li, and K. Sreenath. Demonstrating Berkeley Humanoid Lite: An Open-source, Accessible, and Customizable 3D-printed Humanoid Robot, Apr. 2025.

[21] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong. Design and Control of a Miniature Bipedal Robot with Proprioceptive Actuation for Dynamic Behaviors. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8547–8553, Philadelphia, PA, USA, May 2022. IEEE. ISBN 978-1-72819-681-7. doi:10.1109/ICRA46639.2022.9811790.

[22] NAO H25. NAO H25 — Aldebaran 2.1.4.13 Documentation, 2018.

[23] ROBOTIS. OP3 E-Manual, 2018.

[24] K. Lab. Kscales Lab Zeroth Bot, 2025.

[25] M. Grimmer, A. A. Elshamanhory, and P. Beckerle. Human Lower Limb Joint Biomechanics in Daily Life Activities: A Literature Based Requirement Analysis for Anthropomorphic Robot Design. *Frontiers in Robotics and AI*, 7, Feb. 2020. ISSN 2296-9144. doi:10.3389/frobt.2020.00013.

[26] A. Robotics. Cassie Sets a Guinness World Record, 2016.

[27] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, M. Bloesch, K. Hartikainen, A. Byravan, L. Hasenclever, Y. Tassa, F. Sadeghi, N. Batchelor, F. Casarini, S. Saliceti, C. Game, N. Sreendra, K. Patel, M. Gwira, A. Huber, N. Hurley, F. Nori, R. Hadsell, and N. Heess. Learning Agile Soccer Skills for a Bipedal Robot with Deep Reinforcement Learning. *Science Robotics*, 9 (89):eadi8022, Apr. 2024. doi:10.1126/scirobotics.adi8022.

[28] F. et al. Improving Human–Robot Interaction by Enhancing NAO Robot Awareness of Human Facial Expression. *Sensors*, Jan. 2021.

[29] Y. et al. Sim-to-Real Transfer for Biped Locomotion. In *IROS*, 2019.

[30] G. Maldonado, F. Bailly, P. Souères, and B. Watier. On the coordination of highly dynamic human movements: An extension of the Uncontrolled Manifold approach applied to precision jump in parkour. *Scientific Reports*, 8(1):12219, Aug. 2018. ISSN 2045-2322. doi:10.1038/s41598-018-30681-6.

[31] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi. OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning. In *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*, July 2024.

[32] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, L. Fan, and Y. Zhu. HOVER: Versatile Neural Whole-Body Controller for Humanoid Robots, Oct. 2024.

[33] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. HumanPlus: Humanoid Shadowing and Imitation from Humans. In *8th Annual Conference on Robot Learning*, Sept. 2024.

[34] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. ExBody2: Advanced Expressive Humanoid Whole-Body Control, Dec. 2024.

[35] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. MobileTeleVision: Predictive Motion Priors for Humanoid Whole-Body Control, Dec. 2024.

[36] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. In *Proceedings of the 5th Conference on Robot Learning*, pages 91–100. PMLR, Jan. 2022.

[37] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, June 2018. ISBN 978-0-9923747-4-7. doi:10.15607/RSS.2018.XIV.010.

[38] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning Quadrupedal Locomotion over Challenging Terrain. *Science Robotics*, 5(47):eabc5986, Oct. 2020. doi:10.1126/scirobotics.abc5986.

[39] O. X. Collaboratoin. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, May 2024. doi:10.1109/ICRA57147.2024.10611477.

[40] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. Srirama, L. Chen, K. Ellis, P. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. Ma, P. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. Walke, B. Wulfe, T. Xiao, J. Yang, A. Yavary, T. Zhao, C. Agia, R. Baijal, M. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. Foster, J. Gao, D. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, July 2024. ISBN 9798990284807. doi:10.15607/RSS.2024.XX.120.

[41] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei. Scaling Robot Supervision to Hundreds of Hours with RoboTurk: Robotic Manipulation Dataset through Human Reasoning and Dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1048–1055, Nov. 2019. doi:10.1109/IROS40897.2019.8968114.

[42] R. Grandia, E. Knoop, M. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer. Design and Control of a Bipedal Robotic Character. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, July 2024. ISBN 9798990284807. doi:10.15607/RSS.2024.XX.103.

[43] J. Aldaco, T. Armstrong, R. Baruch, J. Bingham, S. Chan, D. Dwibedi, C. Finn, P. Florence, S. Goodrich, W. Gramlich, A. Herzog, J. Hoech, T. Nguyen, I. Storz, B. Tabanpour, J. Tompson, A. Wahid, T. Wahrburg, S. Xu, S. Yaroshenko, and T. Z. Zhao. ALOHA 2: An Enhanced Low-Cost Hardware for Bimanual Teleoperation, 2024.

[44] M. Izani, Aishah, A. Eshaq, and Norzaiha. Keyframe Animation and Motion Capture for Creating Animation: A Survey and Perception from Industry People. In *Proceedings. Student Conference on Research and Development, 2003. SCORED 2003.*, pages 154–159, Aug. 2003. doi:10.1109/SCORED.2003.1459684.

[45] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A Physics Engine for Model-Based Control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, Oct. 2012. doi:10.1109/IROS.2012.6386109.

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *ArXiv*, July 2017.

[47] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura. A Closed-Form Solution for Real-Time ZMP Gait Generation and Feedback Stabilization. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 936–940, Seoul, South Korea, Nov. 2015. IEEE. ISBN 978-1-4799-6885-5. doi:10.1109/HUMANOIDS.2015.7363473.

[48] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2. doi:10.15607/RSS.2023.XIX.026.

[49] ROBOTIS. Dynamixel-X E-Manual, 2018.

[50] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem. Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation, 2021.

[51] X. Gu, Y.-J. Wang, and J. Chen. Humanoid-Gym: Reinforcement Learning for Humanoid Robot with Zero-Shot Sim2Real Transfer, Apr. 2024.

[52] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen. Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning. In *Robotics: Science and Systems*, 2024.

[53] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi:10.1109/CVPR.2016.90.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. doi:10.1109/CVPR.2009.5206848.

[55] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, July 2024. ISBN 9798990284807. doi:10.15607/RSS.2024.XX.045.

8 Appendix

8.1 3D Printing

3D printing is a deliberate design choice when combined with miniature-sized parts, not just a cost-saving measure. A quick derivation from Euler-Bernoulli beam bending theory suggests that ToddlerBot's smaller structures experience much less relative deflection with the same material strength—$\delta = \frac{PL^3}{3EI} \rightarrow \frac{\delta}{L} \propto \frac{PL^2}{3EL^4} = \frac{P}{3EL^2}$, where δ is deflection, P is point load at tip, E is material elastic modulus, L is characteristic length, and $I \propto L^4$ is second moment of area. We estimate that our miniature (smaller L) 3D-printed parts (smaller E) offer comparable strength to full-size aluminum ones. Using metal parts would reduce reproducibility with marginal gains. Our gears are carefully tuned for minimal backlash, precise transmission, and direct motor integration, while off-the-shelf parts can't achieve this. We also incorporate various bearings to minimize friction in power transmission components. After over a year of testing on two ToddlerBot instances, we are confident that ToddlerBot's reliability matches that of commercial platforms.

8.2 Ease of Assembly and Maintenance

The ease of assembly and maintenance is crucial yet difficult to optimize, as it requires mentally simulating the assembly and disassembly process. In early iterations, we explicitly optimize for fewer screw types and ensured unobstructed tool access, which allowed a clear assembly direction for the screwdriver. We also prioritized modular design, allowing individual parts to be replaced independently. These considerations significantly improve maintainability and simplify repairs. As we continue our research on this platform, we have addressed several reliability issues, including motor overheating and link durability. We are committed to releasing ongoing fixes and upgrades to improve the humanoid further.

8.3 Range of Motion

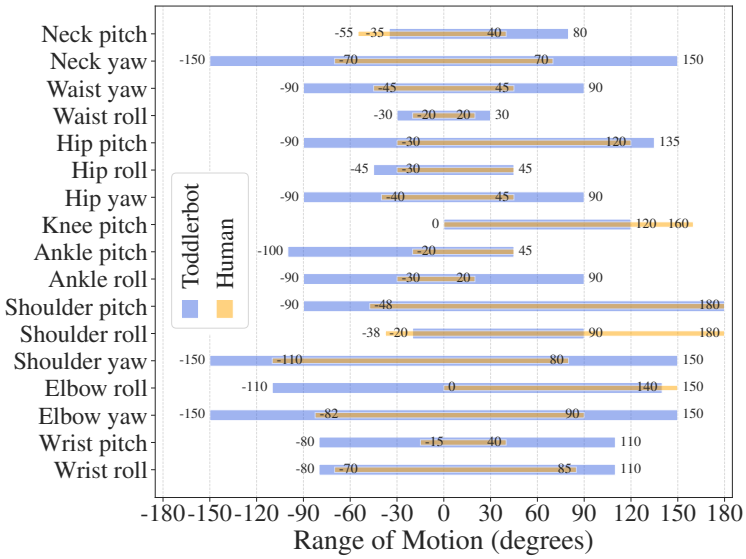
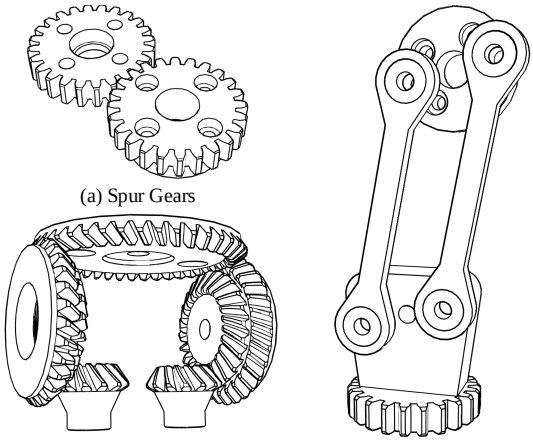


Figure 6: **Range of Motion.** We show that ToddlerBot has near-human or even superhuman mobility in most joints. Negative values represent extensions, adductions, and inversions, while flexions, abductions, and eversions are positive.

We design each joint's range of motion based on human biomechanics, optimizing geometries to prevent self-collisions and achieve near-human or even superhuman mobility in most joints (Figure 6). The additional extension of the ankle pitch partially compensates for the $40°$ gap in the knee



(a) Spur Gears

(b) Coupled Bevel Gears

(c) Parallel Linkages

Figure 7: **Transmission Mechanisms.** We show three design primitives in ToddlerBot's mechanical design: spur gears, coupled bevel gears, and parallel linkages.

pitch and the absence of toes. For shoulder roll, although $90°$ is limited compared to human shoulder abduction, the same hand-up pose can be achieved by actuating the shoulder pitch joint.

8.4 Transmission Mechanisms

Placing motors directly at the joint is often impractical due to the limited space budget. With carefully designed transmission mechanisms, motors can be relocated outside the interference zone, amplify torque output, and offload mechanical stress to the structure. This section highlights key design features, including spur gears, coupled bevel gears, and parallel linkages, as shown in Figure 7.

Each transmission type offers unique benefits. To start with, spur gears provide three advantages:

- **Relocated joint axis:** A 1:1 spur gear set allows repositioning of the joint axis to a more convenient in-plane location. This is widely used in ToddlerBot's arm.
- **Torque modification:** A ratioed spur gear set adjusts the final torque output, which is particularly useful for the parallel jaw gripper.
- **Load distribution:** When a motor's output shaft has significant free play, as in Dynamixel XC330, where it is supported only by a Teflon bushing, using it directly as the joint axis is undesirable. A 1:1 spur gear set enables a reinforced secondary axis with planar bearings and metal shafts to carry the load, protecting the motor from transverse forces. This approach is used in the hip yaw joints, where torque demands are low, but load-bearing capacity is critical.

With precise tolerance tuning, 3D-printed bevel gears provide a highly interlocking design with minimal backlash, while still being structurally strong. They also offer three key advantages:

- **Rotated joint axis:** A coupled bevel gear set enables a parallel waist mechanism, where two motors in the same orientation drive two perpendicular DoFs.
- **Combined torque output:** On each axis, both motors contribute to the driving torque, enhancing power and efficiency. This is critical, as a single Dynamixel XC330 lacks the power to drive the entire upper body, but two motors combined are sufficient.
- **Compact actuation:** In the waist, where space is highly constrained, a coupled bevel gear set allows the compact integration of two DoFs.

Lastly, parallel linkages allow the motor to be positioned away from the joint axis, as seen in the knee and neck pitch. Despite a slightly smaller range of motion (usually $< 160°$), this design efficiently transfers high torque when paired with ball bearings. They provide three key benefits:

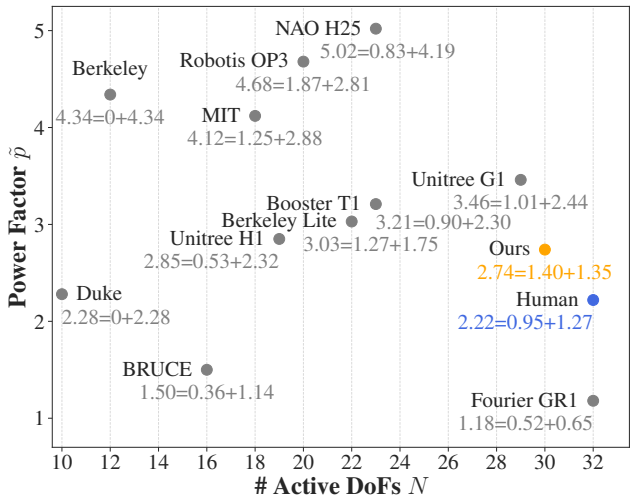


Figure 8: **Humanoid Metrics.** Two key criteria of humanoid capability are the number of active DoFs and power factor $\tilde{p}$ (Equation 4). The total power factor is the sum of the upper and lower body power factor: $\tilde{p}_{\text{total}} = \tilde{p}_{\text{upper}} + \tilde{p}_{\text{lower}}$. ToddlerBot is the closest to human compared with other humanoids, implying potentially comparable loco-manipulation capabilities.

- **Compact design:** This enables a cleaner neck design by placing the motor inside the head.
- **Reduced Inertia:** The knee motor is placed higher to reduce rotational inertia.
- **Structural Efficiency:** In the thigh, the knee motor is bolted to a 3D-printed structure for better load distribution, increased rigidity, and reduced weight.

A potential drawback of these transmission mechanisms is their inaccurate simulation modeling. However, in MuJoCo [45], we mitigate this by using joint equality constraints for spur gears, fixed tendons for coupled bevel gears, and weld constraints for parallel linkages. This approach has empirically shown a small sim2real gap, as demonstrated in Section 5.

8.5 Power Factor

To quantitatively assess a humanoid robot's capability, we propose $\tilde{p}$ as the power factor, representing the total torque (and thus mechanical power) a robot can generate relative to its weight and height. Intuitively, a higher $\tilde{p}$ means that a humanoid can perform energetic, dynamic motions more easily. We argue that $\tilde{p}$ should at least exceed the human threshold $\tilde{p}_{\text{human}}$ to achieve human-like motion, given the inherent gap between robot and human policies, assuming humans operate as an oracle policy that is energy efficient. However, raising $\tilde{p}$ far beyond $\tilde{p}_{\text{human}}$ can have adverse effects: unnatural motion, excessive reliance on motor power, fewer DoFs to accommodate larger motors, reduced battery life, and increased safety concerns. Thus, pushing $\tilde{p}$ past diminishing returns involves a practical trade-off. As shown in Figure 8, ToddlerBot has a $\tilde{p}$ score closest to humans.

When comparing the performance of humanoids with different scales and weights, directly having a full-sized 1.8 m humanoid and a 0.5 m scaled humanoid both jump 0.5 m or run at 3 m/s is not a fair comparison. A more reasonable approach is to normalize performance metrics, for example, by evaluating a jump at 10% of body height or a running speed of twice the body length per second. Formally, we say two humanoids have the **same performance** if they execute the same sequence of joint motions over a time span T, and their total power consumption is the same fraction of their motors' maximum power:

$$\frac{\int_0^T p(t)dt}{\sum_{i=0}^{N} |\tau_i^{\max} \dot{q}_i|} \approx \frac{\Delta h \cdot mg}{\sum_{i=0}^{N} |\tau_i^{\max} \dot{q}_i|} \approx \frac{h \cdot mg}{\sum_{i=0}^{N} |\tau_i^{\max} \dot{q}_i|}, \tag{3}$$

where $p(t)$ is the humanoid's power output at time t. $\tau_i^{\max}$ and $\dot{q}_i$ are the maximum torque and joint velocity of the i-th motor in the humanoid respectively. Moreover, when computing the maximum

power, we take the absolute value to ensure that all motors do positive work. For the numerator, the integral of the power output over T, which is equivalent to the work done by the robot, can be approximated by the gravitational energy gained: $\Delta h \cdot mg$. Since Δh is approximately proportional to the height of the humanoid, we further replace Δh with just the humanoid's height h.

Based on Equation 3, we now define power factor to measure the performance of a humanoid:

$$\tilde{p} = \frac{\sum_{i=0}^{N} |\tau_i^{\max}|}{h \cdot mg}. \tag{4}$$

Note that we flip the fraction from Equation 3 to make the power factor value increase as the utilized torque ratio decreases. We also drop $\dot{q}$ as the same sequence of joint motion would be executed when using power factor to compare the performance of humanoids.

Chi et al. [20] proposes a similar power metric but with an additional normalization by N, the number of active DoFs. However, we find it to be an inappropriate modification. Consider an exaggerated case where two humanoids have the same height and weight: one have a single DoF and the other have 100 equally capable DoF. Normalizing by N would assign them the same power factor, which does not reflect the true actuation capacity of the system at all.

8.6 Motor Selection

We choose Dynamixel motors because of their robustness, reliability, and accessibility. Different types of Dynamixel motors were selected for various joints based on space constraints, torque requirements, and cost considerations. In terms of communication speed, Dynamixel motors communicate via a 5V TTL protocol running at 2M baudrate, providing full-state feedback for all 30 motors at 50 Hz using an off-the-shelf communication board. Precision-wise, our motors have a backlash of about 0.25°, on par with most QDD joints used in full-scale humanoids. Also, the Dynamixel encoder resolution is 4096 pulse/rev, and thus the resolution of the motor position reading is 0.09°.

Given ToddlerBot's size constraint and 30-DoF design, Brushless Direct Drive (BLDC) motors are not a viable option. As BLDC motors shrink, their winding thickness decreases, reducing current capacity and torque constant. Despite their high power density, they still require a high-ratio gearbox, making them less suitable given our limited space budget. We initially explored electric linear actuators, but found them unsuitable due to insufficient power density and low control frequency. After a few iterations, we narrowed our choices to servo motors, ultimately selecting Dynamixel motors for their desirable performance and well-documented support. Given that reproducibility is a hard constraint in our system, we believe a pure Dynamixel design is the most feasible to reproduce, especially for those with limited hardware experience.

For Dynamixel motors, the smallest units start at approximately 50 g. This allows us to estimate the total weight as $3100 = 30 \times 50$ (motors) $+ 600$ (computer, battery, camera) $+$ 1000 (3D-printed structure and metal hardware) g.

To estimate the torque required for each joint to achieve human-like motions, we followed the reference values from [25] and Equation 4 to derive the torque estimation:

$$\boldsymbol{\tau}_{\text{robot}} = \frac{\mathrm{h}_{\text{robot}} \cdot \mathrm{m}_{\text{robot}}}{\mathrm{h}_{\text{human}} \cdot \mathrm{m}_{\text{human}}} \cdot \boldsymbol{\tau}_{\text{human}}. \tag{5}$$

With an estimated height of 0.5 m and weight of 3.1 kg, the required lower limb torque for ToddlerBot to perform the most demanding tasks such as running and slope climbing [25], is estimated as follows: $\tau_{\text{robot}}^{\text{knee}} = 2.35$ Nm, $\tau_{\text{robot}}^{\text{ankle pitch}} = 2.66$ Nm, $\tau_{\text{robot}}^{\text{hip pitch}} = 1.77$ Nm.

As shown in Table 2, XM430 is the only option that provides sufficient torque for the knee and ankle pitch joints. Its metal gears, low backdrive resistance, and high torque output make it ideal for these joints that directly impact walking stability. For the hip, we use 2XC430s for roll and pitch, as they provide sufficient torque while maintaining a compact design, integrating two actuated DoFs in a single housing. This setup allows for a greater range of motion compared to placing two

Table 2: Dynamixel Motor Assignments for ToddlerBot.

Motor Model	Stall Torque[(a)]	Assigned DoFs
XC330-T288	1.0	Neck PY[(b)], Waist RY, Hip Y, Gripper
XC430-T240BB	1.9	Shoulder P, Ankle R
XM430-W210	3.0	Knee P, Ankle P
2XL430-W250	1.5	Shoulder RY, Elbow RY, Wrist RP
2XC430-W250	1.8	Hip RP

[(a)] The Stall Torque data are measured at 12V as reported on the Dynamixel official website [49]. The unit is Nm.
[(b)] R, P, and Y denote roll, pitch, and yaw respectively.

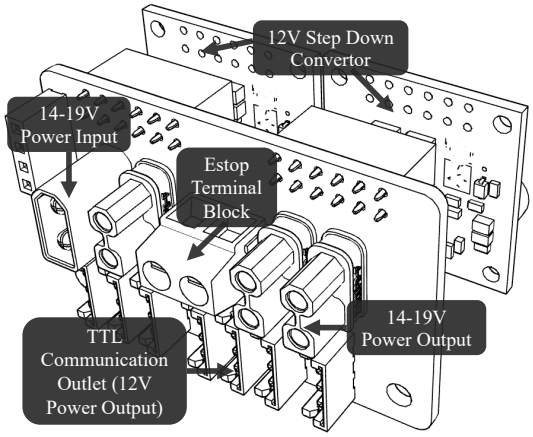


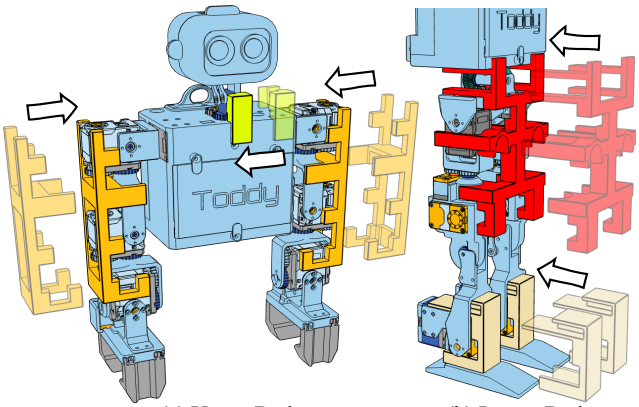
Figure 9: **Power Distribution.** We show the power distribution board design, including four XT30 power plugs, an Estop terminal block, seven JST EH TTL communication outlets, and two 12V step-down convertors.

XC430s sequentially. The XC330 series is the smallest and most cost-effective in the lineup, but has higher tracking errors and backlash. Therefore, we use XC330s on joints with lower torque demands or strict weight and space constraints, such as the neck, waist, hip yaw, and parallel jaw gripper. 2XL430, a lower-cost variant of the 2XC430, is used in the arm to balance performance and cost, ensuring minimal performance loss while maximizing affordability. For the remaining joints, including the shoulder pitch and ankle roll, XC430 is used as a standard choice.

8.7 Power Budget

With the motor selection finalized, we can estimate the power budget. The computer and camera together consume approximately 15 W under typical operating conditions. Actual walking power consumption depends on the energy efficiency of the control policy, but preliminary analysis suggests that the upper body requires minimal power during walking, as it carries no additional load. The lower body operates with alternating support, meaning only one leg is actively working at a time. Assuming a 70% duty cycle, total motor power consumption is estimated at 50 W, requiring a 75 Wh battery for one hour of continuous walking.

For the battery, we offer two options: A 2000 mAh LiPo battery (215 g) available off-the-shelf or a custom-made 4-cell 21700 battery with 5000 mAh, which has higher energy density and weighs 330 g. In practice, the battery lasts 3–5 hours in research settings where the robot walks intermittently. The peak power output from the battery is $14.8\ \text{V} \times 25\ \text{A} = 370\ \text{W}$, sufficient to power all the joints. When debugging without a battery, a 15 V 300 W power supply is a practical alternative—safer and easier to obtain than those required for full-scale humanoids, which often exceed 2 kW and 60 V. In practice, ToddlerBot's battery can last about 2 hours, longer than other commonly available humanoids - Zeroth [24]: 20 min, OP3 [23]: 10-15 min, BRUCE [21]: 20 min, Berkeley Humanoid Lite [20]: 30 min, NAO H25 [22]: 1 hour, Unitree G1 [15]: 2 hours.



(a) Upper Body (b) Lower Body

Figure 10: **Zero-point Calibration.** We 3D-print devices for the plug-and-play zero-point calibration procedure: orange for the arm, yellow for the neck, red for the hip, and beige for the ankle. Arrows indicate the insertion direction, and the zero-point is fixed once the devices click into place.

As shown in Figure 9, the battery provides a $14 - 19\text{V}$ input, regulated to 12V via dual step-down converters to power the motors through TTL communication outlets. An E-stop terminal block controls motor power, enabling emergency reboots. The $14 - 19\text{V}$ output powers the Jetson Orin NX, which remains on when the battery is connected to prevent data loss from abrupt shutdown.

8.8 Zero-point Calibration

Figure 10 shows the zero-point calibration process with the 3D-printed devices.

8.9 Motor Test Bed

Each motor family used on ToddlerBot differs in gearbox material, gear ratio, and core, resulting in distinct stiction, backdrive resistance, damping, and inertia. To enable successful sim-to-real transfer, we must minimize the gap between simulated and real actuator behavior.

Specifically, MuJoCo/MJX models a simplified actuator characteristics with predominantly 3 values: frictionloss, damping, and armature [45]. Frictionloss is the minimum torque $\boldsymbol{\tau_f}$ required for the actuator to start moving, with a unit of Nm. Damping $\mathbf{d}$ refers to the rate at which backdrive resistance increases with speed, measured in Nms/rad. Armature I denotes the effective rotor inertia, accounting for the gearbox, with units of kgm^2. The model in MuJoCo does not consider stiction peak, therefore, the actuator resistance is simply:

$$\boldsymbol{\tau_r} = \boldsymbol{\tau_f} + \mathbf{d} \cdot \dot{\mathbf{q}} \tag{6}$$

Therefore, it is straightforward to design a test platform to measure these physical parameters using a series of test sequences. Specifically, damping and friction loss can be measured by backdriving the motor at a constant RPM and recording the resisting torque via a torque sensor. A linear fit to the torque-speed data yields friction loss as the intercept at 0 RPM and damping as the slope. To estimate armature inertia, the actuator is allowed to spin freely before cutting motor power to observe spin-down behavior. Using the previously measured damping values, the resistance power can be numerically integrated to estimate the initial stored energy E. The armature inertia I is then computed using:

$$E = I \cdot \omega^2 \rightarrow I = \frac{E}{\omega^2} \tag{7}$$

We have developed the test profile and analysis code to automatically identify these parameters in MuJoCo. If a more complex actuator model is needed, our modular test bed can be easily extended with additional sensors. The current setup supports active braking ($5\ \text{Nm}$), active driving ($1\ \text{Nm}$),

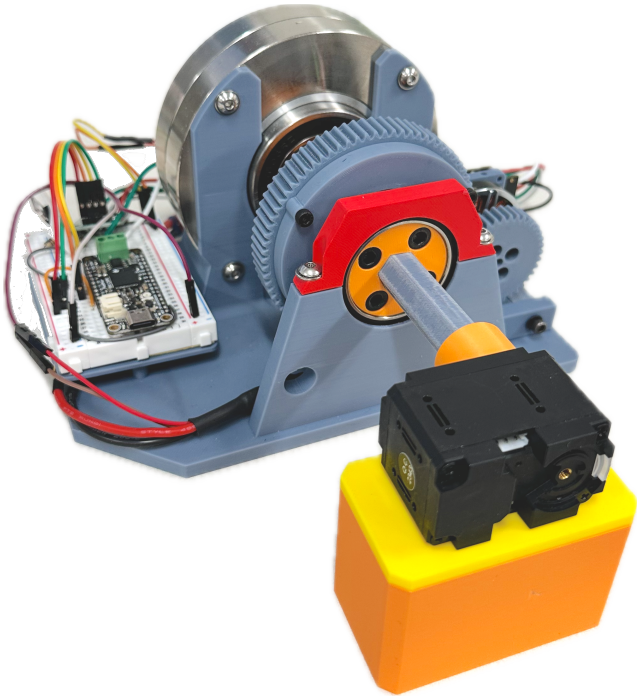


Figure 11: **Motor Test Stand.** We show a photo of the assembled motor test stand. The test motor is mounted via the blue-grey quick-connect shaft, which is directly coupled to a torque sensor. A secondary motor on the side provides active driving torque, while a powder brake at the rear offers up to $5\ \mathrm{Nm}$ of controllable resistance. On the left, a controller MCU handles torque sensing, brake actuation, and CAN communication with the driving motor.

and accurate torque measurements (precision $3e^{-4}$ Nm), meeting most motor SysID requirements. The test bed will be open-sourced with the release. A photo of the setup is shown in Figure 11.

8.10 Actuation Model

The PD position control equation is computed through:

$$\boldsymbol{\tau}_m = \mathbf{k}_p(\hat{\mathbf{q}} - \mathbf{q}) - (\mathbf{k}_d^{\min} + \mathbf{k}_d)\dot{\mathbf{q}}, \tag{8}$$

where $\mathbf{k}_p$ and $\mathbf{k}_d$ are the gains, $\hat{\mathbf{q}}$ is the joint setpoint, $\mathbf{q}$ is the joint position, and $\dot{\mathbf{q}}$ is the joint velocity.

During motor testing, we observed significant additional damping when the motor was powered on, even with $\mathrm{k}_d = 0$. We modeled this effect as $\mathrm{k}_d^{\min}$, separate from passive damping, as it only occurs when the motor is active and should respect torque limits. Additionally, we found the conversion factor from Dynamixel's unitless k_p to physical k_p used in simulators to be approximately 150.

Inspired by Grandia et al. [42], we design our actuation model as follows. The motor torque limit $\boldsymbol{\tau}_{\text{limit}}$ varies with velocity:

$$\boldsymbol{\tau}_{\text{limit}} = \begin{cases} \boldsymbol{\tau}_{\max}, & |\dot{\mathbf{q}}| \le \dot{\mathbf{q}}_{\boldsymbol{\tau}_{\max}} \\ \dfrac{\dot{\mathbf{q}}_{\max} - |\dot{\mathbf{q}}|}{\dot{\mathbf{q}}_{\max} - \dot{\mathbf{q}}_{\boldsymbol{\tau}_{\max}}} \cdot \boldsymbol{\tau}_{\max}, & \dot{\mathbf{q}}_{\boldsymbol{\tau}_{\max}} < |\dot{\mathbf{q}}| \le \dot{\mathbf{q}}_{\max} \\ 0, & |\dot{\mathbf{q}}| > \dot{\mathbf{q}}_{\max} \end{cases} \tag{9}$$

$\boldsymbol{\tau}_{\text{limit}}$ comprises a constant torque limit $\boldsymbol{\tau}_{\max} > 0$ for low velocities, and a linear reduction in available torque beyond a specific velocity $\dot{\mathbf{q}}_{\boldsymbol{\tau}_{\max}}$. This linear limit reaches zero torque at the velocity $\dot{\mathbf{q}}_{\max}$. Note that $\boldsymbol{\tau}_{\text{limit}}$ is the maximum acceleration torque, but maximum deceleration torque is assumed to be always a constant $\boldsymbol{\tau}_{\text{brake}}$. We separate the braking torque limit rather than using the

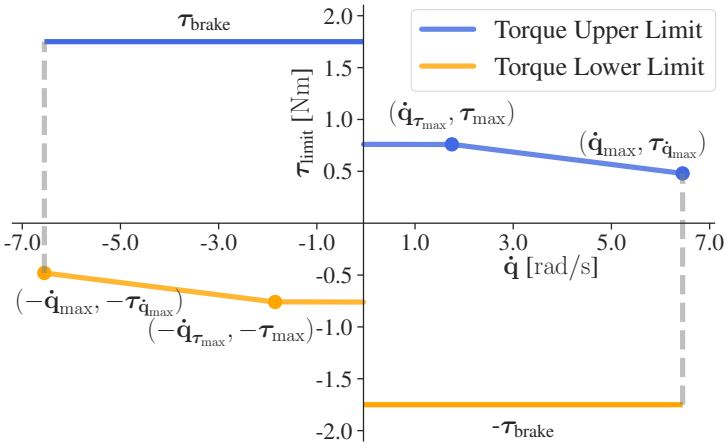


Figure 12: The relationship between torque limit τ_{limit} and joint velocity $\dot{\mathrm{q}}$ for Dynamixel XC330.

Table 3: SysIDed Parameters for Dynamixel Motors.

Parameter	**2XL430**	**XC330**	**XC430**	**2XC430**	**XM430-W210**
Damping[(a)]	0.0010	0.0036	0.0066	0.0028	0.0056
Armature	0.0083	0.0040	0.0042	0.0044	0.0022
Friction Loss	0.078	0.036	0.024	0.060	0.025
$\tau_{\max}$	0.94	0.76	1.32	1.09	1.61
$\dot{\mathrm{q}}_{\tau_{\max}}$	2.00	1.80	1.60	2.00	0.10
$\dot{\mathrm{q}}_{\max}$	5.97	6.50	7.00	6.78	7.63
$\tau_{\dot{\mathrm{q}}_{\max}}$	0.10	0.48	0.21	0.23	0.47
$\mathrm{k}_d^{\min}$	0.161	0.384	0.170	0.185	0.203
τ_{brake}	1.40	1.75	3.00	2.20	3.70

[(a)] The units of damping and $\mathrm{k}_d^{\min}$ are $\mathrm{Nms/rad}$, the unit of armature is kgm^2, the units of frictionloss, $\tau_{\max}$ and τ_{brake} are Nm, and the units of $\dot{\mathrm{q}}_{\tau_{\max}}$ and $\dot{\mathrm{q}}_{\max}$ are $\mathrm{rad/s}$.

acceleration torque limit as in Grandia et al. [42], since the motor typically provides higher braking torque due to passive resistance and gearbox inefficiencies. Additionally, the gearbox efficiency has a large effect when the motor is back-driven by external torque. Concretely, when the motor is back-driven, it shows significantly higher k_p, as the external torque is consumed by the gearbox inefficiency. We model this by a parameter called passive-active ratio, which is equivalent to $\frac{1}{\eta^2}$, where η is gearbox efficiency. We have empirically determined the passive-active ratio to be 3, equating gearbox torque efficiency around 58%. The resulting sysID relationship for Dynamixel XC330 is illustrated in Figure 12.

The joint torque is calculated by applying torque limits to $\boldsymbol{\tau}_m$ and combining the resistance force $\boldsymbol{\tau}_r$:

$$\boldsymbol{\tau} = \begin{cases} \text{clamp}_{[-\boldsymbol{\tau}_{\max}, \boldsymbol{\tau}_{\text{brake}}]}(\boldsymbol{\tau}_m) - \boldsymbol{\tau}_r, & \dot{\mathbf{q}} \geq 0 \\ \text{clamp}_{[-\boldsymbol{\tau}_{\text{brake}}, \boldsymbol{\tau}_{\max}]}(\boldsymbol{\tau}_m) + \boldsymbol{\tau}_r, & \dot{\mathbf{q}} < 0 \end{cases} \tag{10}$$

$\boldsymbol{\tau}_r$ follows the joint passive force model in MuJoCo [45], which is characterized by three parameters: damping, armature, and friction loss. The sysIDed parameters for various Dynamixel motors are presented in Table 3.

Using this actuator model, we jointly optimize all parameters to minimize the gap between simulated and real-world tracking. The model includes 9 parameters: damping, frictionloss, armature, $\boldsymbol{\tau}_{\max}$, $\dot{\mathbf{q}}_{\boldsymbol{\tau}_{\max}}$, $\boldsymbol{\tau}_{\dot{\mathbf{q}}_{\max}}$, $\dot{\mathbf{q}}_{\max}$, $\mathbf{k}_d^{\min}$, and $\boldsymbol{\tau}_{\text{brake}}$. Due to the complexity of this optimization, constraining parameter ranges is crucial for convergence to a valid local minimum. The final simulation achieves an average tracking error of 1.3° with the optimized parameters (Table 3).

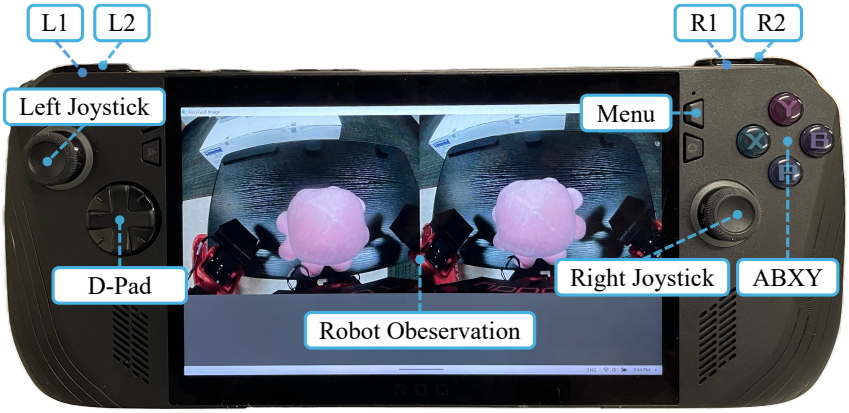


Figure 13: **Remote Controller Layout**. We show the button and axis layout on ROG Ally X.

Table 4: Pupperteering Button and Axis Mapping

Button or Axis[(a)]	**Effect**
Menu	Toggle teleoperation / Mark episode start or end during data collection
Left Joystick ↕	Walk forward or backward along the x-axis
Left Joystick ↔	Walk leftward or rightward along the y-axis
Right Joystick ↕	Stand up or squat down
Right Joystick ↔	Turn clockwise or counterclockwise around the z-axis
D-Pad ↕	Lean with the waist roll joint
D-Pad ↔	Twist with the waist yaw joint
Y and A	Look up or down with the neck pitch joint
X and B	Look left or right with the neck yaw joint
L1	Hold to run the bimanual DP and release to end.
R1	Hold to run the full-body DP and run to end.
L2	Hold to run the wagon pushing policy and release to end.
R2	Hold to run the cuddling policy and release to end.

[(a)] This mapping is compatible with various remote controllers and has been tested on the ROG Ally X and Steam Deck. The remaining buttons can be assigned to additional skills based on user preference.

8.11 Pupperteering Mapping

Figure 13 illustrates the remote controller layout to teleoperate ToddlerBot, with button and axis mappings detailed in Table 4. During teleoperation, the human operator sends velocity commands to the walking policy and determines the timing for skill transitions. The same mapping was used for Steam Deck and ROG Ally X.

8.12 Reinforcement Learning Details

The RL implementation leverages MJX [45] and Brax [50]. We train the policy using PPO [46] with hyperparameters listed in Table 5. Inspired by prior work [51, 52], our reward function is shaped by three categories of reward terms as detailed in Table 6. Full implementation details are available in our open-source codebase. During inference, the RL policy runs on the CPU of Jetson Orin NX 16GB, achieving a 50 Hz control loop.

Table 5: Hyperparameters for PPO Training.

Parameter	Value
Policy hidden layer sizes	$(512, 256, 128)$
Value hidden layer sizes	$(512, 256, 128)$
Number of timesteps	3×10^8
Number of environments	1024
Episode length	1000
Unroll length	20
Batch size	256
Number of minibatches	4
Number of updates per batch	4
Discounting factor	0.97
Learning rate	0.0001
Entropy cost	0.0005
Clipping epsilon	0.2

Table 6: Reward Shaping for PPO Training.

Imitation Term	Value
Torso quaternion	1.0
Linear velocity (XY)	5.0
Linear velocity (Z)	1.0
Angular velocity (XY)	2.0
Angular velocity (Z)	5.0
Leg motor position	5.0
Feet contact	1.0
Regularization Term	
Feet air time	500.0
Feet clearance	0.05
Feet distance	1.0
Feet slip	0.05
Align with the ground	1.0
Stand still	1.0
Torso roll	0.5
Torso pitch	0.5
Collision	0.1
Leg action rate	0.05
Leg action acceleration	0.05
Motor torque	0.01
Energy	0.05
Survival Term	
Survival	10.0

8.13 Diffusion Policy Details

The diffusion policy processes a cropped and downsampled 96×96 RGB image, which is encoded by a ResNet [53] pretrained on ImageNet [54] to extract visual features. Both leader and follower joint angles are downsampled to 10 Hz for training, where the leader joint angles serve as actions and the follower as observations. To prevent motor overload during data collection, the upper body motors use low proportional gains, allowing modulation of the manipulation force. This behavior is embedded in the discrepancy between leader and follower joint angles, which the policy learns.

The model is trained with 100 diffusion steps. During inference, the trained model runs directly on the Jetson Orin NX 16GB with 3 DDPM steps, which are sufficient for satisfactory results.

With 300M parameters, the inference latency remains under 0.1 s on the GPU, ensuring smooth execution at 10 Hz without stuttering. Each inference yields a 16-step prediction; the first 3 actions are discarded to compensate for latency [55], and the next 5 actions are executed.

8.14 Velocity Tracking Details

Table 7 shows the quantitative results of the walking velocity tracking experiment.

Table 7: Tracking Errors in Simulation versus the Real World

Tracking Errors	Simulation	Real-World
Position [m]	0.082	0.133 ± 0.018
Linear Velocity [m/s]	0.016	0.032 ± 0.002
Angular Velocity [rad/s]	0.056	0.113 ± 0.010

8.15 Skill Chaining Details

Figure 14 shows the results of the skill chaining experiment.

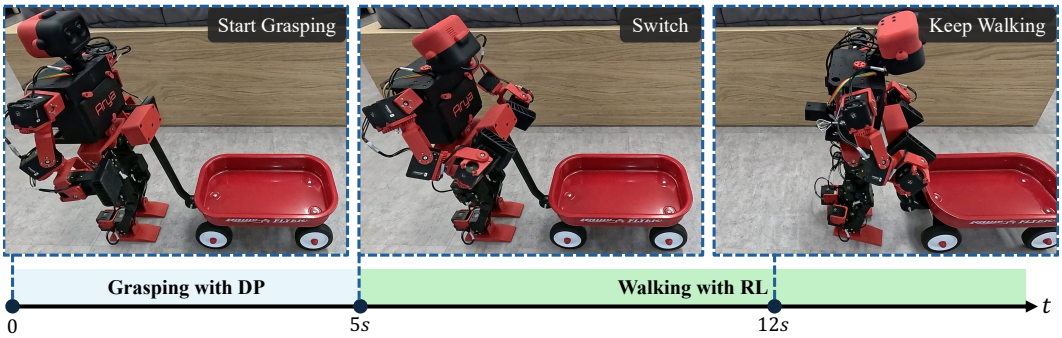


Figure 14: **Skill Chaining.** We demonstrate that ToddlerBot can seamlessly transition from a DP-based grasping skill to an RL-trained walking policy, enabling it to grasp the wagon handle and push it forward while maintaining its grip.

8.16 Reproduction Results

Figure 15 shows some successful reproductions of ToddlerBot by teams around the world.

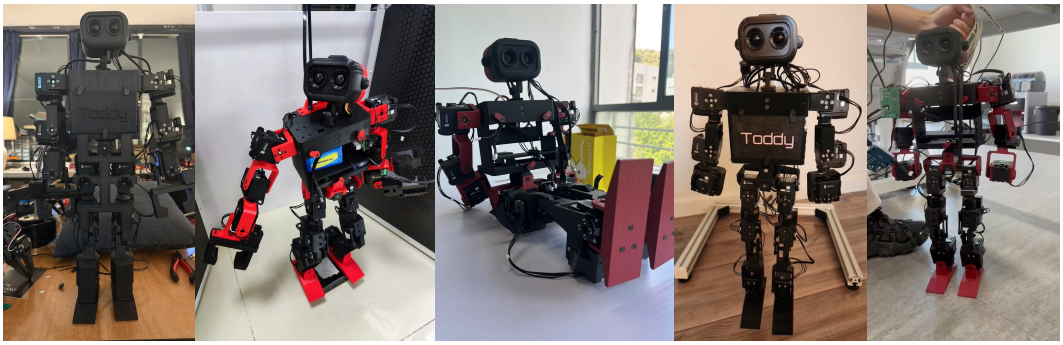


Figure 15: **Successful Reproductions.** We showcase five independent replications of ToddlerBot by teams across the globe. On average, each team completed the replication in about a week.

Figure 16 shows our open-source assembly manual.

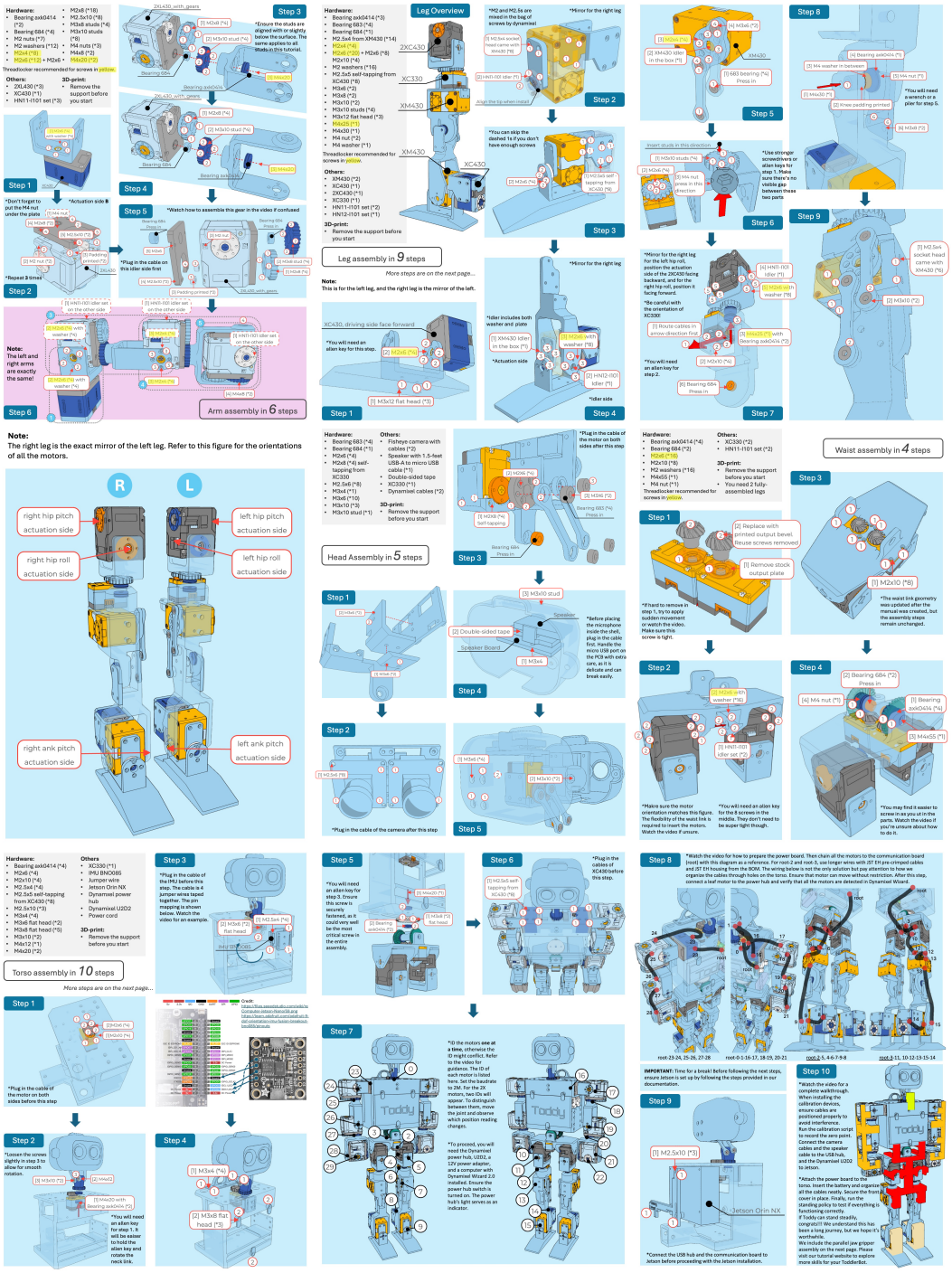


Figure 16: **Assembly Manual.** We present the open-source assembly manual for ToddlerBot.