Visual Whole-Body Control for Legged Loco-Manipulation

Minghuan Liu*[1,2], Zixuan Chen*[1,3], Xuxin Cheng[1], Yandong Ji[1], Rizhao Qiu[1], Ruihan Yang[1], Xiaolong Wang[1]

[1]UC San Diego, [2] Shanghai Jiao Tong University, [3] Fudan University

https://wholebody-b1.github.io

***Abstract*—We study the problem of mobile manipulation using legged robots equipped with an arm, namely legged loco-manipulation. The robot legs, while usually utilized for mobility, offer an opportunity to amplify the manipulation capabilities by conducting whole-body control. That is, the robot can control the legs and the arm at the same time to extend its workspace. We propose a framework that can conduct the whole-body control autonomously with visual observations. Our approach, namely Visual Whole-Body Control (VBC), is composed of a low-level policy using all degrees of freedom to track the body velocities along with the end-effector position, and a high-level policy proposing the velocities and end-effector position based on visual inputs. We train both levels of policies in simulation and perform Sim2Real transfer for real robot deployment. We perform extensive experiments and show clear advantages over baselines in picking up diverse objects in different configurations (heights, locations, orientations) and environments.**

I. Introduction

The study of mobile manipulation has achieved large advancements with the progress of better manipulation controllers and navigation systems. While installing wheels for a manipulator can help solve most household tasks [23, 7], it is very challenging to adopt these robots outdoors with challenging terrains. Imagine going camping; having a robot picking up trash and wood for us would be very helpful. To achieve such flexibility and applications in the wild, we study mobile manipulation with legged robots equipped with an arm, i.e., legged loco-manipulation. Specifically, we are interested in how the robot can conduct tasks based on its visual observation autonomously.

While this is appealing, it is a very challenging control problem to coordinate all the joints simultaneously with large degrees of freedom (19 DoF). The robot will need to exploit the contact with its surroundings and the objects and maintain stability and robustness to external disturbances all at the same time. Besides, acquiring a mobile robot platform to do precise manipulation jobs requires a steady robot body, which is much harder for legged robots compared to wheeled robots. An even larger challenge comes from achieving all these in diverse environments autonomously, given only the observation from an egocentric camera. Recent learning-based approaches have shown promising results on legged robots avoiding obstacles, climbing stairs, and jumping over stages robustly [13, 17, 1, 22, 26, 4, 27, 5, 16, 14], some with visual inputs. However, all these efforts still focus only on locomotion without manipulation, which requires more precise control. The extra complexity and more challenging tasks make direct end-to-end learning infeasible.

In this paper, we conduct loco-manipulation by introducing a two-level framework with a high-level policy proposing end-effector pose and robot body velocity commands based on visual observations and a low-level policy tracking these commands. Such a hierarchical design is effective as the low-level controller can be effectively shared when interacting with diverse environments and objects. We named our framework **V**isual Whole-**B**ody **C**ontrol (VBC). Specifically, VBC is a hierarchical solution that contains three stages of training: first, we train a universal low-level policy using reinforcement learning (RL) to track any given goals and achieve whole-body behaviors; then, we train a privileged teacher policy by RL to provide a set of appropriate goals and guide the low-level policy to accomplish specific tasks (e.g., pick-up); finally, to deploy the policy into the real robot, we distill the teacher policy into a depth-image-based visuomotor student policy via online imitation learning. All this training is done in simulation, and we perform direct Sim2Real transfer for deployment in the real robot.

Our hardware platform is built on a Unitree B1 quadruped robot equipped with a Unitree Z1 robotic arm. We conducted the pickup task with 14 different objects under three height configurations, all achieved high success rates.

We summarize the contribution of this paper as:

- We develop an autonomous vision-based loco-manipulation system that utilizes model-free RL and sim-to-real techniques to learn whole-body control policies for legged robots.
- Our proposed method emerges retrying behavior and generalizes well to complicated terrains, various objects, and heights.

II. Visual Whole-Body Control

In this section, we introduce our Visual Whole-Body Control (VBC) framework (as described in Fig. 1). Our VBC framework consists of a low-level goal-reaching policy and a high-level task-planning policy. Our low-level goal-reaching policy tracks a root velocity command and a target end-effector pose. Our vision-based high-level task-planning policy provides velocity and end-effector pose command to the low-level policy, given the segmented depth images and proprioception of the robot as inputs. When combined, our robot

* The first two authors contributed equally.

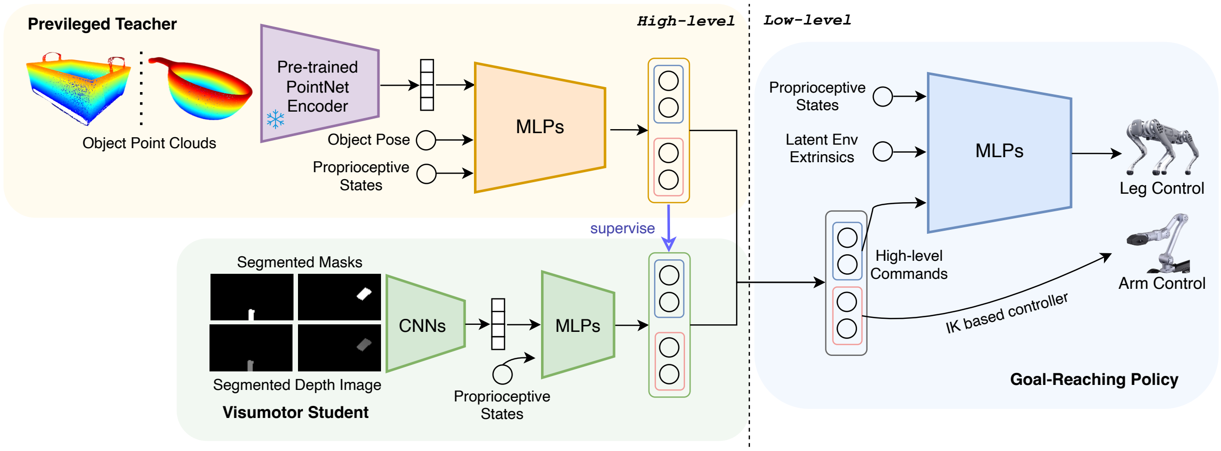


Fig. 1: **Training a vision-based whole-body policy for loco-manipulation.** Our VBC framework trains a low-level control policy and a high-level planning policy using reinforcement learning and imitation learning.

could operate fully autonomously using only visual input and its proprioception. We start with an overview of our robot setup, followed by the training pipeline of the low-level policy and the high-level policy, and our robot system design.

A. Low-Level Policy for Whole-Body Goal-Reaching

The low-level control policy that takes over the whole-body locomotion control π^{low} as described in the blue part in Fig. 1, is trained to track any given end-effector poses and body velocities across various terrains. In this work, we turn to the help of RL and utilize ROA (Regularized Online Adaptation) [6, 3, 11] to realize such a universal and robust behavior that can become the foundation of any high-level tasks. The network architecture used for low-level policy is MLP.

Commands. The command $\mathbf{b}_t$ for our low-level policy is defined as:

$$\mathbf{b}_t = [\mathbf{p}^{\text{cmd}}, \mathbf{o}^{\text{cmd}}, v_{\text{lin}}^{\text{cmd}}, \omega_{\text{yaw}}^{\text{cmd}}]$$

where $\mathbf{p}^{\text{cmd}} \in \mathbb{R}^3, \mathbf{o}^{\text{cmd}} \in \mathbb{R}^3$ are end-effector position and orientation command; $v_{\text{lin}}^{\text{cmd}} \in \mathbb{R}, \omega_{\text{yaw}}^{\text{cmd}} \in \mathbb{R}$ are the desired forward linear and angular velocity, respectively.

Observations. The observation of our policy is defined as a 90-dimensional vector $\mathbf{o}_t$:

$$\mathbf{o}_t = [\mathbf{s}_t^{\text{base}}, \mathbf{s}_t^{\text{arm}}, \mathbf{s}_t^{\text{leg}}, \mathbf{a}_{t-1}, \mathbf{z}_t, \mathbf{t}_t, \mathbf{b}_t].$$

Among them, $\mathbf{s}_t^{\text{base}} \in \mathbb{R}^5$ is the current quadruped base state including roll, pitch, and quadruped base angular velocities, $\mathbf{s}_t^{\text{arm}} \in \mathbb{R}^{12}$ is arm state (position and velocity of each arm joint except the end-effector), $\mathbf{s}_t^{\text{leg}} \in \mathbb{R}^{28}$ is leg state (joint position and velocity of each leg joint, and foot contact patterns that are 0 or 1), $\mathbf{a}_{t-1} \in \mathbb{R}^{12}$ is the last output of policy network, $\mathbf{z}_t \in \mathbb{R}^{20}$ is the environment extrinsic vector. Besides, Agarwal et al. [1], we provide some extra timing reference variables $\mathbf{t}_t$ to the observation to learn a steady walking behavior.

Actions. Our low-level policy outputs the target joint angles for all 12 joints of the robot.

Arm control. Fu et al. [6] has revealed that whole-body RL policy is good at controlling the end-effector position but struggles when controlling the orientation, which are critical for manipulation. Therefore, we use inverse kinematics (IK) to convert the end-effector pose command into target joint angles for arm control.

Domain randomization. When training, we randomize the type of terrains, including flat plane and rough ones. We also randomized the friction between robots and terrains. Besides, we randomized the quadruped's mass and center of mass. These randomizations, together with our ROA module, contribute to a robust low-level policy.

B. High-Level Policy for Task Planning

When deploying our policy into the real world, the robot can only perceive the object through depth images to achieve high-frequency control. However, RL with visual observation is quite challenging and requires particular techniques to be efficient [10, 25, 9, 8, 21]. To achieve simple and stable training, we choose to train a privileged state-based policy that can access the shape information of the object (as described in the orange part of Fig. 1) and distill a visuomotor student (as described in the green part of Fig. 1). The low-level policy is fixed during this training process.

C.1. Privileged Teacher Policy

Privileged observations. The privileged observations include the encoded shape feature and object poses. We formally define it as a 1094-dim vector:

$$\mathbf{o}_t = [\mathbf{z}^{\text{shape}}, \mathbf{s}_t^{\text{obj}}, \mathbf{s}_t^{\text{proprio}}, \mathbf{v}_t^{\text{base vel}}, \mathbf{a}_{t-1}],$$

where $\mathbf{z}^{\text{shape}} \in \mathbb{R}^{1024}$ is the latent shape feature vector encoded from the object point clouds using a pre-trained PointNet++ [18]. The pre-trained PointNet++ is fixed, so $\mathbf{z}^{\text{shape}}$ remains invariant during training. $\mathbf{s}_t^{\text{obj}} \in \mathbb{R}^6$ is the object pose in local observation w.r.t the robot arm base. $\mathbf{s}_t^{\text{proprio}} \in \mathbb{R}^{53}$ consists of joint positions including gripper joint position $\mathbf{q}_t \in \mathbb{R}^{19}$, joint velocity not including gripper joint velocity $\dot{\mathbf{q}}_t \in \mathbb{R}^{18}$, and end-effector position and orientation $\mathbf{s}_t^{\text{ee}} \in \mathbb{R}^6$. $\mathbf{v}_t^{\text{base vel}} \in \mathbb{R}^3$ is the base velocity of robot and $\mathbf{a}_{t-1} \in \mathbb{R}^9$ is the last high-level action.

High-level Actions. The high-level policy π^{high} outputs actions that determine the velocity of the robot, the target end-effector pose and whether the gripper is open.

Reward functions. The reward function for training our state-based policy comprises task-specific rewards and assistant rewards. For the task-specific rewards, we design three stages. The first stage is approaching, where the corresponding reward r_{approach} encourages the gripper and quadruped to get close to the target object. The second stage is task progress, where the reward r_{progress} leads the robot to the desired final state, e.g., lifting the object in the pickup task. The third stage is task completion, where reward $r_{\text{completion}}$ bonus when the agent completes the task. The assistant rewards are designed to smooth the robot's behavior and prevent deviation and sampling useless data.

Domain randomization. During training, we randomized the friction between the robot and the terrain, the mass of the robot, the center of mass of the robot, the transmission frequency between low-level and high-level policy, and the motor strengths of the arm.

C.2. Visuomotor Student Policy

Observations and actions. To mitigate the sim2real gap, we use the segmented depth images and the corresponding segmentation masks. Other observations include proprioceptive information and last action defined above. The output actions are the same as those of state-based policy.

Imitation learning for student distillation. The vision-based policy is distilled from state-based policy using DAgger [20]. Specifically, transitions are sampled using the student policy and supervised by the teacher. Augmentations like RandomErasing, GaussianBlur, GaussianNoise, and RandomRotation are applied to the depth images.

System design. During deployment, the low-level policy that controls the quadruped works at a frequency of 50Hz; in the meantime, the high-level policy runs at 10Hz, which is five times slower than the low-level policy.

Vision masks. Note that our system works with only limited human intervention, i.e., we need to annotate and segment objects from environment backgrounds. To annotate and segment objects while keeping tracking them, we develop TrackingSAM [19], a tool that combines the tracking model AOT (Associating Objects with Transformers) [24] and SAM (Segment Anything) [12]. During real-world experiments, we annotated the objects at the beginning, then the TrackingSAM will keep tracking the segmented object and providing the mask. We carefully implemented the TrackingSAM so that it can run at around 10Hz during deployment on Nvidia Jetson Orin.

III. Experiments

We conduct a set of experiments on pick-up tasks to compare VBC with baselines both in simulation and the real world, showing effectiveness from the following perspectives:

- VBC solves the mobile manipulation (pickup) task over varying objects with different heights;
- VBC has the advantage over easy baseline methods, such as decoupling mobile manipulation problems into separate moving and arm-control;
- The object shape feature produced by pre-trained PointNet++ boosts learned policy grasp varying objects;

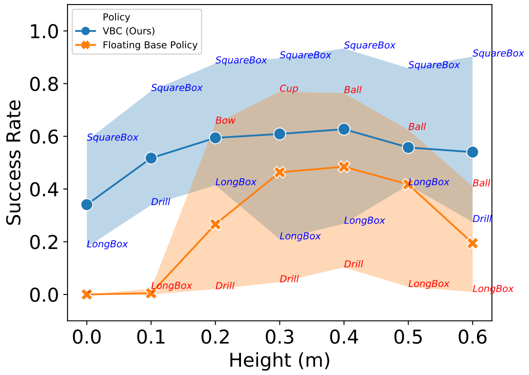


Fig. 2: **Success rates** of different student policies at different heights, tested in the *simulator*.

We utilize IsaacGym [15] for massive parallel training and simulation evaluation.

A. Simulation Results

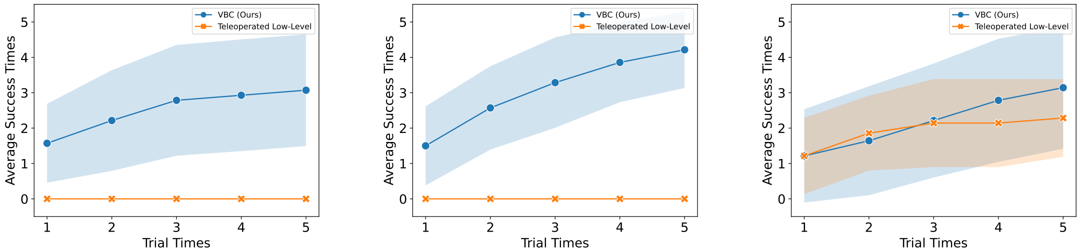
Objects. We included 34 objects in total, mainly adapted from the YCB dataset [2]. According to the object shapes, we roughly divided them into 7 categories: *ball*, *long box*, *square box*, *bottle*, *cup*, *bowl* and *drill*.

Evaluation principles and baselines. In simulation, we compare both the privileged teacher policies of VBC against baselines, and their visuomotor student policies over 34 kinds of different objects: 1) VBC w.o. shape feature: a visuomotor policy that is trained the same as VBC, but does not access the pre-trained object shape features during the teacher training stage. 2) Floating base: a floating base policy with a perfect low-level navigation ability but without whole-body behavior; in other words, the robot can always follow the given velocity commands and body height commands ranging from 0.4 to 0.55 meters (we set 0.4 meters as the lower bound as the default controller in real-world only allows 0.47 meters in minimum). 3) Non-hierarchical: a unified policy trained with low-level policy and visuomotor high-level policy jointly in an end-to-end style. This policy takes the observation of our low-level and high-level policies together. It outputs all the target positions of 12 robot joint angles and the target pose of the gripper. We failed in training such a policy, indicating the effectiveness and necessity of the training pipeline of VBC. We design these baselines to provide a valuable analysis of the design of each module and part.

Picking up different objects. We test the pick-up *success rate* on each distinct object for more than 300 episodes and collect the results over every category. During test time, all object heights are randomly set from 0.0m to 0.6m for each trial. As shown in Tab. I, VBC achieves the best performance on 4/7 categories of objects. It is also interesting that, with 3D

TABLE I: **Success rates** of VBC compared to baseline method on seven categories of objects, tested in the *simulator*. Detailed descriptions of each method are stated in Section III-A.

	Policy Type	Ball	Long Box	Square Box	Bottle	Cup	Bowl	Drill
Privleged Teacher	**Floating Base**	63.25%	20.51%	72.66%	38.07%	69.55%	71.88%	4.17%
	VBC w.o. Shape Feature	**96.18%**	74.26%	**96.80%**	**82.46%**	84.38%	55.84%	73.77%
	VBC (Ours)	89.08%	**92.83%**	81.87%	76.34%	**89.12%**	**78.14%**	**84.00%**
Visuomotor Student	**Non-Hierarchical**	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Floating Base	42.45%	0.0%	37.36%	8.33%	41.41%	43.57%	3.70%
	VBC (Ours)	**55.40%**	**28.57%**	**80.00%**	**56.57%**	**68.01%**	**58.96%**	**53.33%**



(a) Objects on the ground (0.0m). (b) Objects on a box (∼0.3m). (c) Objects on a table (∼0.5m)

Fig. 3: **Average success times w.r.t. the trial times** of picking up 14 objects at different heights, tested in the *real world*.

features, VBC becomes worse at regular objects (compared to *VBC w.o. Shape Feature*), but is much better on irregular objects with complicated shapes, i.e., *long box*, *cup*, *bowl* and *drill*. One possible reason is that, without pre-trained features, the policy targets the object's center for grasping, which is effective enough on simpler objects yet inapplicable for objects with more complicated shapes. In addition, the performance of the student policy of VBC behaves the worst on *long boxes*. This is because without shape features, it is much harder to determine the pose of the object, and some of the initial poses of *long boxes* are extremely hard to grasp for the gripper we used at some height (also as shown in Fig. 2).

Picking up over various heights. We test the mean *success rate* on each object category on a fixed height, where we also list the object with the maximum/minimum success rate on each height. We compare VBC and the floating base baseline on 7 different heights, and illustrate the results in Fig. 2. From the significant improvement of VBC on almost every height, compared to the floating base policy, we highlight the advantage of VBC that is able to achieve flexible whole-body behaviors in achieving high-level tasks.

B. Real-World Experiments

Objects. In the real world, we choose 14 objects, including four irregular objects, six common objects, and four regular shapes. It is worth noting that there are no same objects compared with simulation during real-world experiments.

Evaluation principles and baselines. We deploy the trained visuomotor student policy of VBC directly into the real world. In the real world, we test how many times the robot can pick up the assigned object in 5 trials (i.e., the robot is allowed to continuously retry four times when it fails to grasp without resetting), as we find that the robot emerges to have retrying behavior, as will be described below. We choose this as a measurement since we find that the emergent retrying behaviors improve the robustness, reliability, and success rate of finishing a job. We consider it a success when the object is picked more than 0.1m higher than the plane where it is placed and a failure when the robot tries more than five times or the object falls from its surface. Each object is evaluated over three different heights, as stated before, and five resets are permitted for each height. Regarding the baseline, we compared VBC with the default Unitree base controller (allowing the heights of the arm base to vary from 0.47m to 0.55m) combined with a similar high-level policy trained on stationary legs, in which the robot has no whole-body behaviors but also retains the retrying. We teleoperated the robot to be closed enough to the object before the policy started to grasp.

Emergent retrying behaviors. It is worth noting that during simulation training, the robot emerges to learn a retrying behavior. In other words, when the robot fails to grasp an object, it will automatically retry to grasp without any human intervention. This shows an advantage and a clear difference compared to model-based methods.

Pickup performance. We collect all results and conclude the averaged performance over all objects on each height setting and show in Fig. 3. It is worth noting that our method allows for autonomous retrying behaviors when one pick-up trial fails. Obviously, VBC surpasses the baseline method on all settings. To highlight, the baseline methods fail at 0.0m and 0.3m, as the default controller keeps a fixed robot height, similar to the floating base baseline. Even on the 0.5m setting, where the teleoperated baseline without whole-body behavior is equivalent to a static robotics arm, VBC is generally better. Besides, VBC shows great generalization ability on unseen shapes, thanks to the training strategy and observation design in the training pipeline.

IV. Conclusions, Limitations and Future Works

In this paper, we proposed a fully autonomous mobile manipulation system based on quadruped robots and a hierarchical training pipeline, Visual Whole-Body Control (VBC). We believe such system contribution shows the feasibility of using hierarchical models for legged loco-manipulation. The limitations can be generalized as: 1) inadequate gripper design, 2) imprecise depth estimation and 3) compounding error of hierarchical models during sim2real.

Acknowledgments

We would like to thank Unitree for their technical support. We thank Chengzhe Jia for the great help on 3D modeling. We also thank Haichuan Che, Jun Wang, Shiqi Yang, and Jialong Li for their kind help in recording demos.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.

[2] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36 (3):261–268, 2017.

[3] Xuxin Cheng, Ashish Kumar, and Deepak Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[4] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[5] Zipeng Fu, Ashish Kumar, Ananye Agarwal, Haozhi Qi, Jitendra Malik, and Deepak Pathak. Coupling vision and proprioception for navigation of legged robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17273–17283, 2022.

[6] 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.

[7] 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.

[8] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2023.

[9] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control, 2023.

[10] Tairan He, Yuge Zhang, Kan Ren, Minghuan Liu, Che Wang, Weinan Zhang, Yuqing Yang, and Dongsheng Li. Reinforcement learning with automated auxiliary loss search. *Advances in Neural Information Processing Systems*, 35:1820–1834, 2022.

[11] 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.

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[13] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[14] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[15] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[16] Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[17] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[19] Rizhao Qiu. *Towards real-time robotics perception with continual adaptation*. PhD thesis, University of Illinois at Urbana-Champaign, 2023.

[20] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[21] Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, Shuzhen Li, Yanjie Ze, Hal Daumé III au2, Furong Huang, and Huazhe Xu. Drm: Mastering visual reinforcement learning through dormant ratio minimization, 2023.

[22] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1430–1440, 2023.

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

[24] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems*, 34:2491–2502, 2021.

[25] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_SJ-_yyes8.

[26] Wenhao Yu, Deepali Jain, Alejandro Escontrela, Atil Iscen, Peng Xu, Erwin Coumans, Sehoon Ha, Jie Tan, and Tingnan Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*, 2021.

[27] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.