# Wheeled Lab: Modern Sim2Real for Low-cost, Open-source Wheeled Robotics

Anonymous Authors

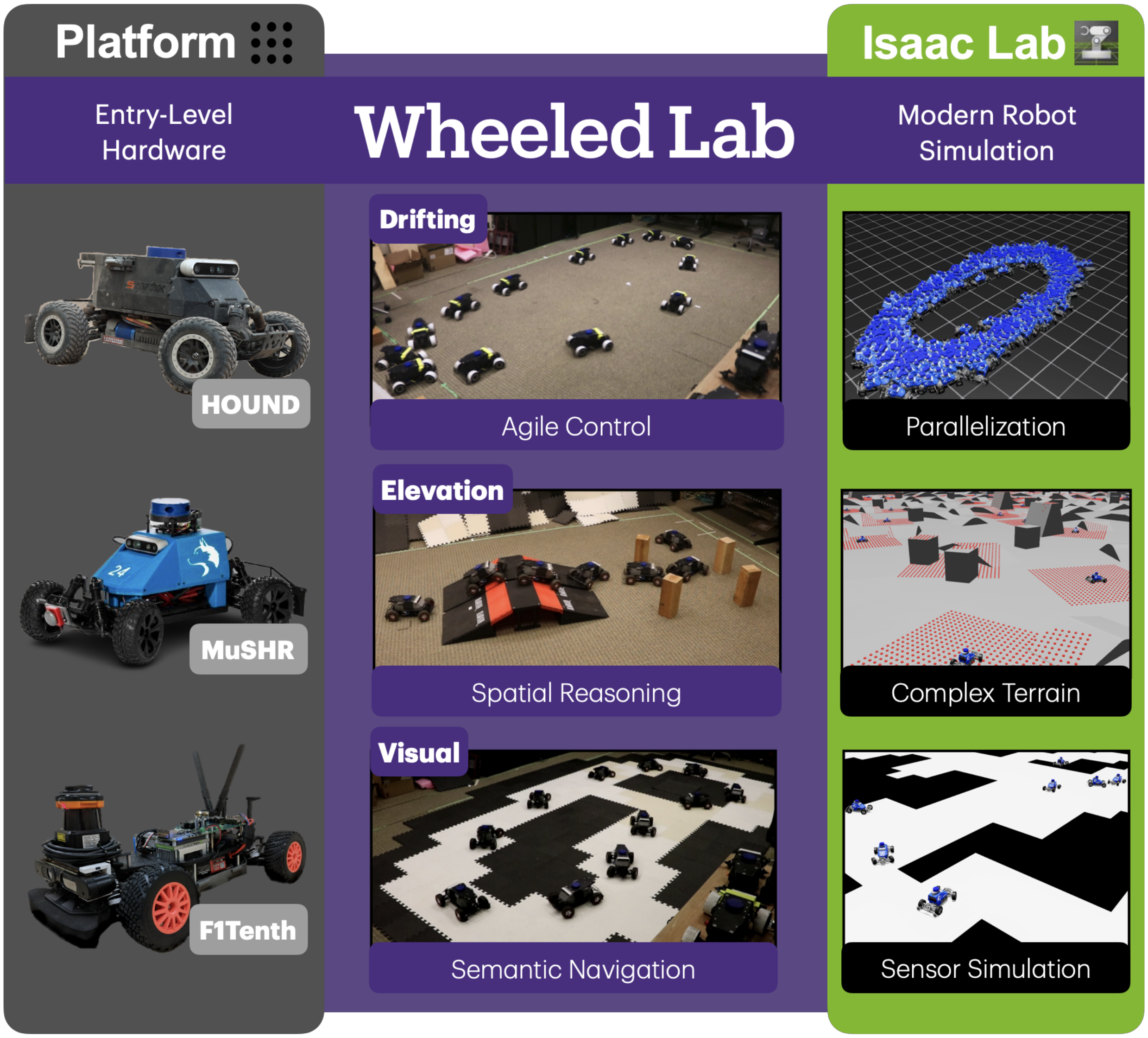


Fig. 1: Wheeled Lab bridges popular low-cost open-source wheeled platforms [45, 43, 38] with the research-backed robotics ecosystem Isaac Lab [36]. This integration helps introduce modern Sim2Real methods to more accessible platforms for a broader audience. Three complete Sim2Real pipelines for state-of-the-art policies are developed in this work to demonstrate advantages of modern methods and help kickstart further education and research.

***Abstract*—Simulation has been pivotal in recent robotics milestones and is poised to play a prominent role in the field's future. However, recent robotic advances often rely on expensive and high-maintenance platforms, limiting access to broader robotics audiences. This work introduces Wheeled Lab, a framework for integrating the low-cost, open-source wheeled platforms that are already widely established in education and research with Isaac Lab, an open-source, widely adopted, and rapidly growing simulation framework for robotics research. Wheeled Lab thus introduces to new user communities modern techniques in Sim2Real, such as domain randomization, sensor simulation, and end-to-end learning. To kickstart educational uses and demonstrate the framework's capabilities, we develop three state-of-the-art policies for small-scale RC cars: controlled drifting, elevation traversal, and visual navigation, each trained in simulation and deployed in the real world. By bridging the gap between advanced Sim2Real methods and affordable, available robotics, Wheeled Lab aims to democratize access to cutting-edge tools, fostering innovation and education in a broader robotics context. The full stack, from hardware to software, is low cost and open-source.**

## I. Introduction

The robotics community has made remarkable strides in recent years, boasting advances that enable quadrupeds to hike mountain trails and perform parkour [35, 26], autonomous drones to race at champion-level ability [31], and robot hands to solve Rubik's Cubes [37]. Shared across these feats is the process of simulated policy learning and direct deployment in the real world, termed *Sim2Real*.

However, these remarkable Sim2Real innovations now target expensive, space-intensive, and high-maintenance systems, including quadrupeds, humanoids, drones, and dexterous manipulators. Such pricey, high-end platforms make it impossible for the general population of robotics builders and users to access the state-of-the-art techniques they employ, such as domain randomization and sensor simulation [31, 28, 33, 26].

In contrast, small-scale autonomous RC cars offer low-cost[1] yet performative platforms that can leverage modern robotics methods, as we demonstate in this paper. Used by classrooms, national racing competitions, and enthusiasts, platforms like F1Tenth [38] have established themselves as go-to, entry-level platforms for beginning roboticists and researchers alike. Crucially, these platforms still provide sufficient onboard real estate to house components like the battery and compute necessary for modern algorithms as well as sufficient mobility to exercise complex dynamics and environment interactions like drifting or off-roading [48, 24, 25, 14]. In short, small-scale wheeled robots have high algorithmic potential at relatively low cost and hardware complexity.

A survey of existing ecosystems for low-cost wheeled platforms indicates that current simulation capabilities are isolated and outdated (see Table I). As a result, education [27] and research opportunities [19] currently limit practitioners to model-based or model-free methods with low-fidelity simulation. With the advent of high-fidelity, open-source simulators like Isaac Lab [36], which are geared toward modern robotic learning, these limited ecosystems no longer need to be the status quo.

This work aims to support education and research by making modern robotics methods available to the broader community. To accomplish this, *we contribute a Sim2Real framework for open-source wheeled robots that we have integrated with a state-of-the-art, research-grounded simulation ecosystem, Isaac Lab.* Modern methods we implement and evaluate include: massive parallelization, domain randomization, sensor simulation, and end-to-end learning. We demonstrate these methods using three policy types trained in simulation and deployed on low-cost, open-source platforms:

1) The **Drifting Policy** ($\pi_{\text{drift}}$) performs a controlled drift through extensive domain randomization, an approach in modern Sim2Real for tasks with uncertain and unstable dynamics. This is the first work to demonstrate direct Sim2Real transfer for drifting without online fine-tuning.
2) The **Elevation Policy** ($\pi_{\text{elev}}$) traverses three-dimensional features with spatial reasoning. This demonstration highlights end-to-end training for modern tasks that tightly couples perception and control. Wheeled Lab is the first to provide integration of an accessible elevation-based Sim2Real pipeline with massive parallelization.
3) The **Visual Policy** ($\pi_{\text{vis}}$) traverses visual features with semantic understanding. This demonstration highlights simulation, end-to-end training, and deployment with cameras, a sensor modality gaining substantial research attention due to its pre-training potential. Wheeled Lab provides a lower cost alternative for visual Sim2Real compared to existing integrations.

All three policies that we demonstrate also exhibit generalization to robustness and recovery behaviors during testing.

By integrating low-cost, open-source platforms with Isaac Lab, this work addresses a current need in robotics to extend modern principles in Sim2Real to resource-constrained audiences.

## II. Related Work

Considerable prior work in Sim2Real has addressed open-source wheeled robots, including [14, 39, 6, 40, 49, 13, 48, 18]. Some of these works produce one of $\pi_{\text{drift}}$, $\pi_{\text{elev}}$, or $\pi_{\text{vis}}$ with partial or full Sim2Real. The individual accomplishments of this research both motivate our choice of demonstrations and help us highlight that *varied state-of-the-art approaches can be coherently integrated under the framework presented here*.

**Drifting Policy**. Drifting is a challenging task in both optimal control and reinforcement learning (RL) [22, 9, 16]. It is a fundamentally unstable maneuver whose control solutions are traditionally determined by vehicle and ground parameters [11]. In many cases (ours included), researchers are unable to demonstrate a controlled drift for imitation-based methods [13].

These challenges make drifting an appealing task for RL. RL-based approaches have thus far been achievable only through further real-world fine-tuning [13, 48]. *In this work, we show that aggressive domain randomization, perturbation*

[1] About 3000 USD or less. See [41] for a comparative analysis.

| | Sensor Simulation | | | Agent Physics | | Reinforcement Learning | | | |
|---|---|---|---|---|---|---|---|---|---|
| Ecosystem | Elevation | LiDAR | Camera | Dynamics | Parallelization | Perturbation | Corruption | Community | Platforms |
| Wheeled Lab | ✓ | 3D | Depth, RGB | 16 DOF + Collisions | High | ✓ | ✓ | RSL, SB3 | HOUND, MuSHR, F1Tenth |
| AutoDrive [41] | ✗ | 2D | ✗ | 16 DOF | Low | ✗ | ✗ | Gym | F1Tenth, Nigel |
| F1Tenth [38] | ✗ | 2D | ✗ | Kinematic | ✗ | ✗ | ✗ | Gym | F1Tenth |
| CARLA [17] | ✗ | 3D | Depth, RGB | 16 DOF + Collisions | ✗ | Steering Only | ✓ | SB3 | ✗ |
| Brunnbauer *et al.* [8] (PyBullet) | ✗ | 2D | RGB | 12 DOF | ✗ | ✗ | ✗ | ✗ | F1Tenth |
| Hamilton *et al.* [23] (Gazebo) | ✗ | 2D | ✗ | Kinematic | ✗ | ✗ | ✗ | ✗ | F1Tenth |
| *PIETRA* [10] (Chrono) | ✓ | ✗ | ✗ | 12 DOF | ✗ | ✗ | ✗ | ✗ | F1Tenth |
| RoboTHOR [15] | ✗ | ✗ | RGB | Kinematic | ✗ | ✗ | ✓ | AllenAct | Household |

TABLE I: **Comparisons of existing ecosystems on their capabilities.** Various simulation, learning, and deployment ecosystems have been integrated with accessible wheeled platforms for Sim2Real. However, these ecosystems are noticeably isolated from the research community and, where implemented, have outdated features.

*simulation, and massive parallelization can enable direct transfer of a drifting policy.* To reduce the barrier-to-entry, we deliberately distance our methods from techniques such as gain tuning and extensive system identification that require additional domain expertise and tooling.

**Elevation Policy.** It has become standard practice for locomotion policies on outdoor terrain to be trained with elevation maps [26]. In parallel, autonomous off-road vehicles have also demonstrated the importance of incorporating elevation in model-based methods [25, 20, 21]. Generally, these elevation-based tasks address challenges that arise at the interface of agent morphology and perception, which are often difficult to model. Therefore, lower-cost wheeled platforms have traditionally been oriented toward flat terrain [38, 6], revealing a gap in methodology between the broader community and the state-of-the-art. Notable exceptions include [14, 44, 50]. However, *no previous work integrates these platforms with RL-facing simulators.*

**Visual Policy.** Improvements in camera and photorealistic scene simulation have advanced policies capable of semantic reasoning, bridging the gap between pre-trained visual models and embodied actions [30, 53]. Demonstrations of small-scale platforms with pre-trained visual models have also been investigated [44]. In general, cameras provide an approachable and interpretable sensor modality for modern robotics education and research. Unfortunately, similar to elevation policies, visual policy pipelines on low-cost mobile platforms are not readily available even though the success of proprietary camera-based Sim2Real platforms suggests a clear demand [6]. *This work demonstrates simulation, end-to-end training, and deployment with cameras on a low-cost platform.*

**Isaac Lab Simulation Framework.** This work builds on Isaac Lab, an open-source, widely adopted, and rapidly growing simulation framework for robotics research [36, 33] that is heavily supported by industry and academia alike [47, 1, 2, 4, 3, 33, 34, 26, 46, 51].

## III. Towards Accessibly Reproducible Sim2Real

Reinforcement Learning (RL) and Sim2Real, especially on full autonomy stacks, presents an extremely challenging frontier for reproducible and approachable robotics. Isaac Lab takes steps towards RL transparency by collecting parameters in a hierarchical configuration framework, exposing relevant simulation, algorithmic, architectural, and computational parameters while abstracting away consistently re-implemented RL behaviors and infrastructure. Wheeled Lab aims to take this a step further by introducing run configurations to improve transparency of the full Sim2Real evaluation cycle.

Figure 1 illustrates this objective of Wheeled Lab, portable and reproducible training and development for Sim2Real. To achieve this, Wheeled Lab bridges low-cost wheeled platforms with Isaac Lab and helps support experiments across these platforms for a variety of tasks. Some of the highest-level abstractions are illustrated in Figure 2 (e.g., Scene, Observation, Reward). Through iterative testing of the Sim2Real pipelines in this work, we have developed reliable and empirically validated baseline configurations of these abstractions, which are amenable specifically to low-cost wheeled platforms. See also Section B for further details about environment and articulation definitions and tuned rewards.

In addition to training, Wheeled Lab itself further provides infrastructure for the evaluation and model selection, such as visualization, benchmarks, and reporting. The configuration structure is extended to wholly include run parameters, architecture, and hyperparameters in addition to the environment configuration itself. These additions help ensure reproducible, accessible, and portable experimentation settings and results. These components are important aspects of the Sim2Real development process, which helps practitioners diagnose and iterate on algorithmic challenges. We find that this aspect of the development cycle can be unfamiliar to those beginning with real-world RL, shifting engineering labor from real-time debugging to iterative model analysis and evaluation.

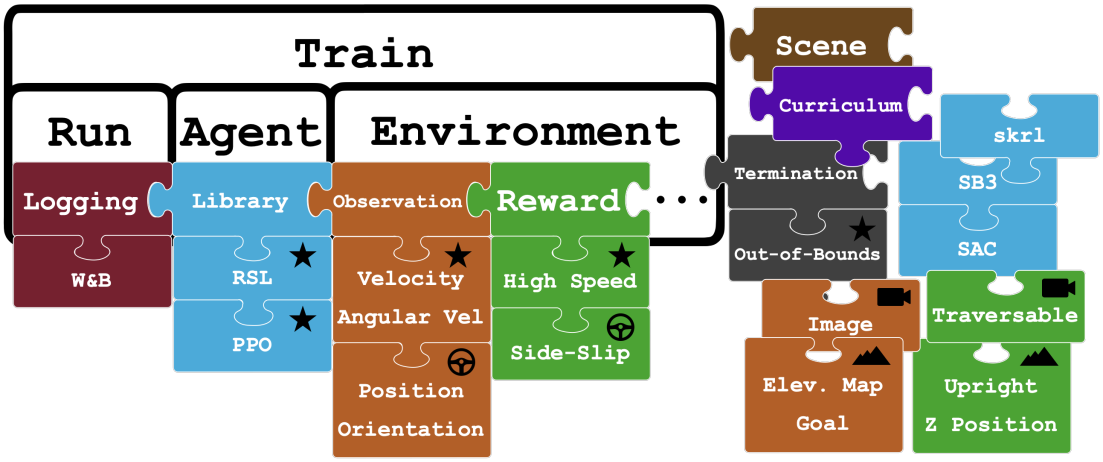


Fig. 2: **Modular training framework imagined as the assembly of a puzzle.** Three main components comprise training: `Run`, `Agent`, `Environment`. Outlined puzzle pieces (e.g., `Observation`, `Reward`) are sub-components whose behaviors are defined by the pieces below them. For instance, `High Speed` is a reward setting in all of our RL environments, as denoted by the star (★) icon. Wheel (⊛), camera (■), and mountain (▲) icons indicate specific design choices for $\pi_{\text{drift}}$, $\pi_{\text{vis}}$, and $\pi_{\text{elev}}$, respectively. Components shown are non-exhaustive.

Fig. 3: Family of HOUND platforms used for a robotics class at the University of Washington.

| Policy | Corruption | Perturbation | DR | # Envs | Epochs |
|---|---|---|---|---|---|
| $\overline{\pi}_{\text{drift}}$ | ✓ | ✗ | ✗ | 64 | 5000 |
| $\pi_{\text{drift}}$ | ✓ | ✓ | ✓ | 1024 | 5000 |
| $\overline{\pi}_{\text{elev}}$ | ✗ | ✗ | ✗ | 128 | 1000 |
| $\pi_{\text{elev}}$ | ✓ | ✓ | ✓ | 1024 | 1000 |

TABLE II: **Training settings**. Baseline policies $\overline{\pi}_{\text{drift}}$ and $\overline{\pi}_{\text{elev}}$ (liberally) reflect capabilities currently available to low-cost, open-source wheeled platforms.

## IV. Demonstrations

For the demonstrations in this work, we use the HOUND [45] for $\pi_{\text{drift}}$ and the MuSHR[2] [43] for $\pi_{\text{elev}}$ and $\pi_{\text{vis}}$. The HOUND's estimated total price is about 3000 USD for a complete autonomy package with sensing (i.e., LiDAR, RGB) and compute (i.e., Jetson Orin NX). The MuSHR is estimated at 930 USD. Within the context of our work, the HOUND platform's hardware is nearly equivalent to that of the F1tenth platform (3800 USD). A single NVIDIA RTX 3080 GPU is used for training.

To demonstrate the capabilities of the framework, Wheeled Lab implements, deploys, and evaluates three state-of-the-art policy types: $\pi_{\text{drift}}$, $\pi_{\text{elev}}$, and $\pi_{\text{vis}}$ on the HOUND platform.

[2]This work's MuSHR uses the same casing as the HOUND but is otherwise a MuSHR in terms of mechanics and electronics.

In each task, we compare the policy trained with modern Sim2Real techniques (e.g., domain randomization, perturbations, etc) against a baseline policy trained without them to demonstrate their effectiveness in Sim2Real transfer in the presented domains. Settings for drifting and elevation tasks are shown in Table II. All other configuration settings not mentioned (e.g., rewards, actuator parameters, articulations) can be assumed to be the same between $\pi$ and $\overline{\pi}$. Each baseline is a liberal representation of what training ecosystems currently provide to low-cost wheeled platforms, as shown in Table I. We use $\overline{(\cdot)}$ to denote the baseline version of a policy (e.g., $\overline{\pi}_{\text{drift}}$). Across all policies, we use Proximal Policy Optimization (PPO) [42] as implemented in the RSL RL library to train the agent in simulation [32].

### A. Drifting

**Background.** Drifting is characterized by a vehicle's *side-slip angle*, i.e., the angle between the heading and the velocity, as it maneuvers through a turn. In high-performance driving, drifting allows the vehicle to take much sharper turns at higher speeds than would be otherwise allowable by the no-slip turn radius [5]. The total mass, center-of-mass, friction properties of the surface and the wheels, actuator properties, state estimation, steering linkages, and suspension are all

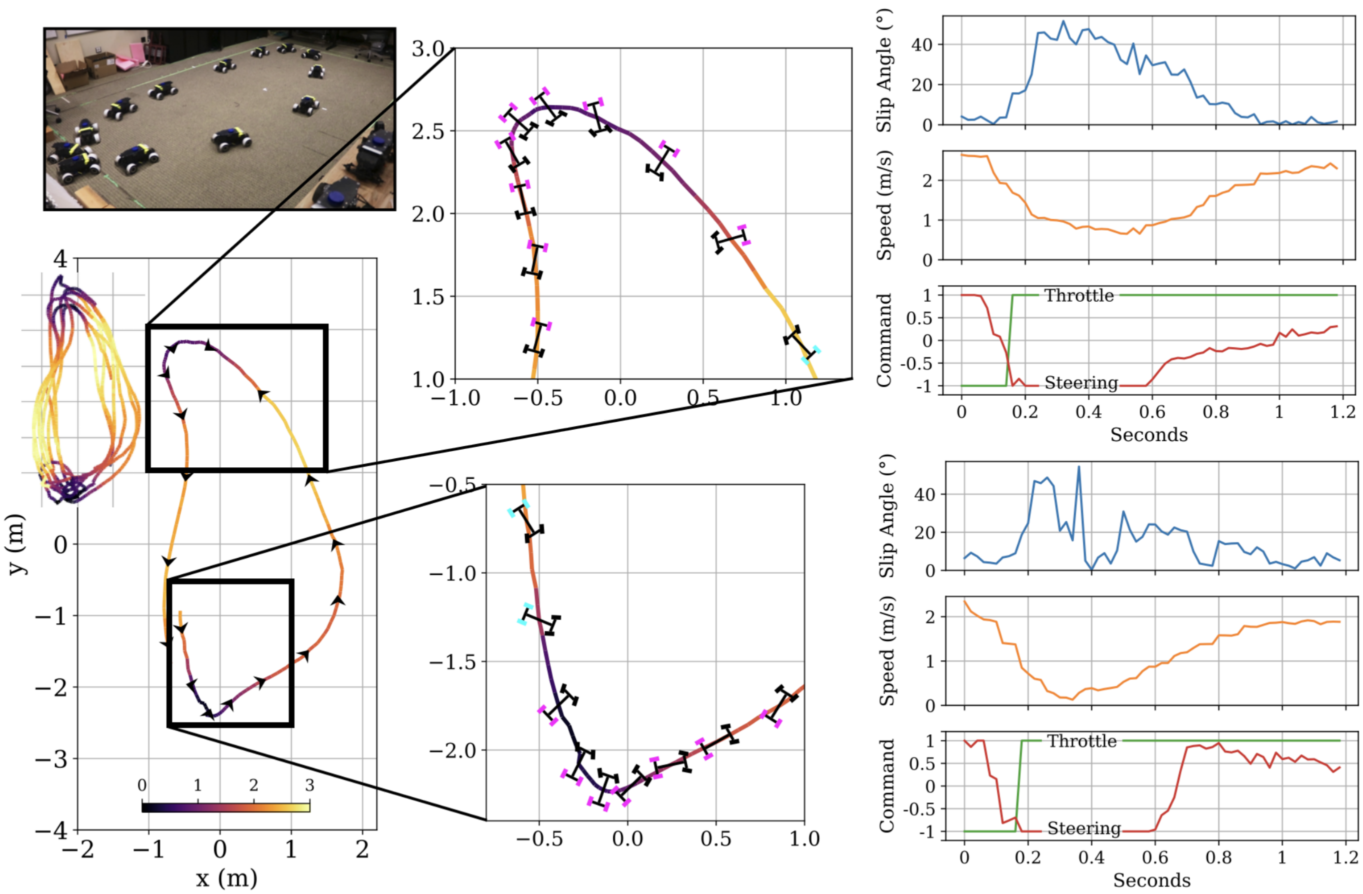


Fig. 4: **A control strategy arises for drifting through turns. Left:** (Overlay: colored trajectories over multiple runs.) One run is spotlighted for this visualization. Arrows indicate the direction of travel, and colorbar denotes speeds (in m/s). **Middle:** Magnification of drifted turns with markers visualizing vehicle orientation, steering (front wheel direction), and throttle (rear wheels; blue: $-1$, pink: $+1$). **Right:** Slip angle, speed, steering, and throttle are plotted for each turn. To initiate the drift, the platform quickly cuts the throttle to destabilize the rear wheels and then steers inwards sharply to throw them outwards. With the platform now facing the track center, it throttles through the remainder of the turn while counter-steering to control its residual angular momentum from the initial maneuver. The entire sequence occurs in just over 1 second. Visualization inspired by [16].

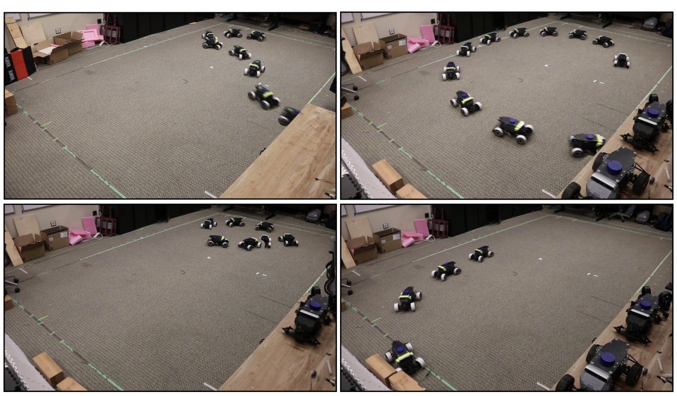

Fig. 5: **Captured trajectories from the baseline drift policy in real experiments.** Baseline is unable to complete a valid turn without crashing or spinning out.

relevant factors in the narrow dynamics of a controlled drift.

The sensitivity of these dynamics becomes especially obvious when we (researchers) attempt the task manually despite the deceptively simple action space (see website for videos). When the maneuver is not executed correctly, the vehicle spins uncontrollably off track, losing traction (also known as a *spin out*) and usually requiring a reset.

**Task.** For our evaluation, the task involves maintaining speed while minimizing the cross-track distance from an oval racing line without spinning out.

**Setup.** State estimation is provided primarily by onboard VIO. We use a motion capture system for evaluation and occasional (1 hz) integrator-drift correction due to space limitations of our indoor testing environment. Successful VIO-only runs not used for data collection are shown on our website. To enable the platform to be physically capable of drifting, we wrapped tape around the platform's rubber wheels to reduce friction and converted the base to a rear-wheel drive.

**Results.** We find that the baseline is unable to make a stable turn, much less complete a full lap (Figure 5). However,

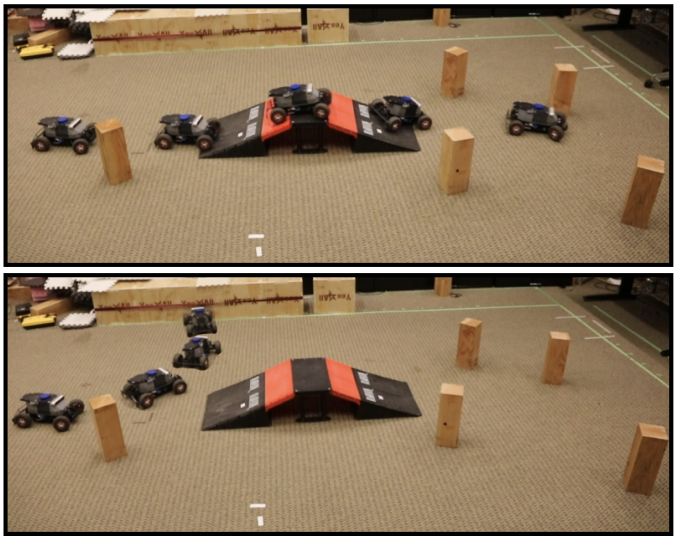

Fig. 6: **Captured trajectories of elevation policy in real experiments. Top:** $\pi_{\text{elev}}$ demonstrates spatial reasoning through ramp traversal and obstacle avoidance in one sequence using only a local elevation map. **Bottom:** Nearing a rollover, the Baseline $\overline{\pi}_{\text{elev}}$ falls off the ramp and then crashes into it.

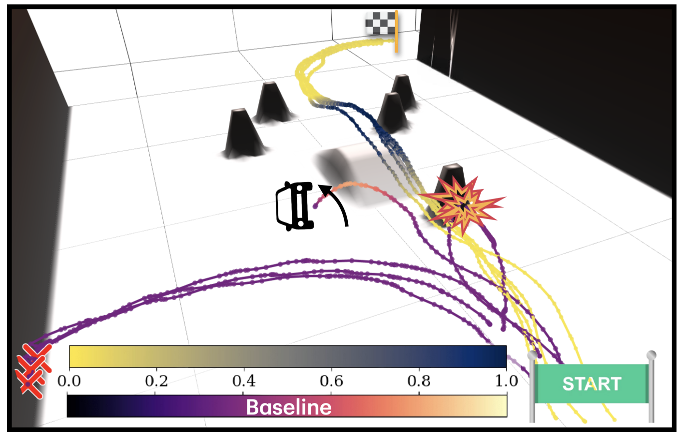


Fig. 7: **Three-dimensional view of global elevation map.** $\pi_{\text{elev}}$ (yellow/blue) and $\overline{\pi}_{\text{elev}}$ (purple/white) trajectories are visualized over the map, with elevation values denoted using color. Note that the policies can only access their local $2.5 \times 2.5$ m maps during evaluation.

| | MLP | | CNN | |
|---|---|---|---|---|
| | No-aug. | Aug. | No-aug. | Aug. |
| # of Success / Trial | 0 / 5 | 3 / 5 | 0 / 5 | 1 / 5 |

TABLE III: **Visual policy results.** Results demonstrate the effectiveness of image augmentations for Sim2Real generalization. While we expect CNN to be more capable of learning, MLP presents better generalization capability. We include videos of each run in the supplementary materials.

it occasionally attempts to counter-steer despite showing no indication during training. Baseline training runs consistently fail to discover the drifting mode. Motor cogging, noisy state estimates, constant slipping, and steering biases cause significant covariate shift.

On the other hand, $\pi_{\text{drift}}$ is able to complete full laps. In addition, we observe evidence of high robustness in $\pi_{\text{drift}}$ not yet seen in previous literature [48, 16]. That is, when the platform does spin out, instead of attempting to regain control by cutting the throttle, the policy maintains its angular momentum for a full spin (or more) before returning to the track. A visualization of the results (Figure 4) gives insight into the policy's precise control. The maximum (controlled) slip angle is $58°$, and the average speed is about 1.6 m/s.

Appendix B-A provides further details about the design and implementation of $\pi_{\text{drift}}$.

### *B. Elevation*

**Background.** On unstructured and uneven terrain, an elevation map is essential for identifying traversable areas in the scene. However, traversability is strongly dependent on agent morphology and dynamic state. Along with the ground geometry, the following affect how to safely traverse uneven terrain: the platform's ground clearance, center-of-mass height, wheel size, maximum torque, suspension, and momentum. Poor traversal can result in catastrophic failures, such as rollovers, and motor (or engine) stalling.

**Task.** To show that the trained elevation-based policy is capable of geometric reasoning, we construct a scene with two types of elevation features that contribute to the overall elevation map, i.e., walls and ramps. Crucially, the incorporation of ramps sets the task apart from sole obstacle avoidance since ramps are features with height but are still traversable with the correct approach strategy. The starting and goal positions are placed on opposite sides of the testing area, with the elevation features placed in between. The platform must safely traverse the scene to arrive at the goal position by avoiding obstacles and ascending ramps when locally feasible.

**Setup.** States provided to the policy are goal-relative position, orientation, and a local ($2.5 \times 2.5$ meters) body-centric elevation map. A motion capture system is used to create the global map from which we sample local maps. Although the platform is capable of creating onboard maps [45], a configurable indoor environment requires higher mapping resolution than is vailable for method evaluation.

**Results.** We find that the baseline primarily deviates from any elevation features to approach the goal. When the baseline does ascend the ramp, it quickly falls off the sides. However, $\pi_{\text{elev}}$ can both traverse the ramp safely and navigate through subsequent obstacles to reach the goal. Like $\overline{\pi}_{\text{drift}}$, the lower number of simulated agents gives the baseline policy significantly fewer opportunities to explore correct trajectories in a narrow range of safe states.

### *C. Visual*

**Background.** Visual navigation traditionally relies heavily on a semantic understanding of the environment through visual feedback [29]. Despite its importance, embedding information

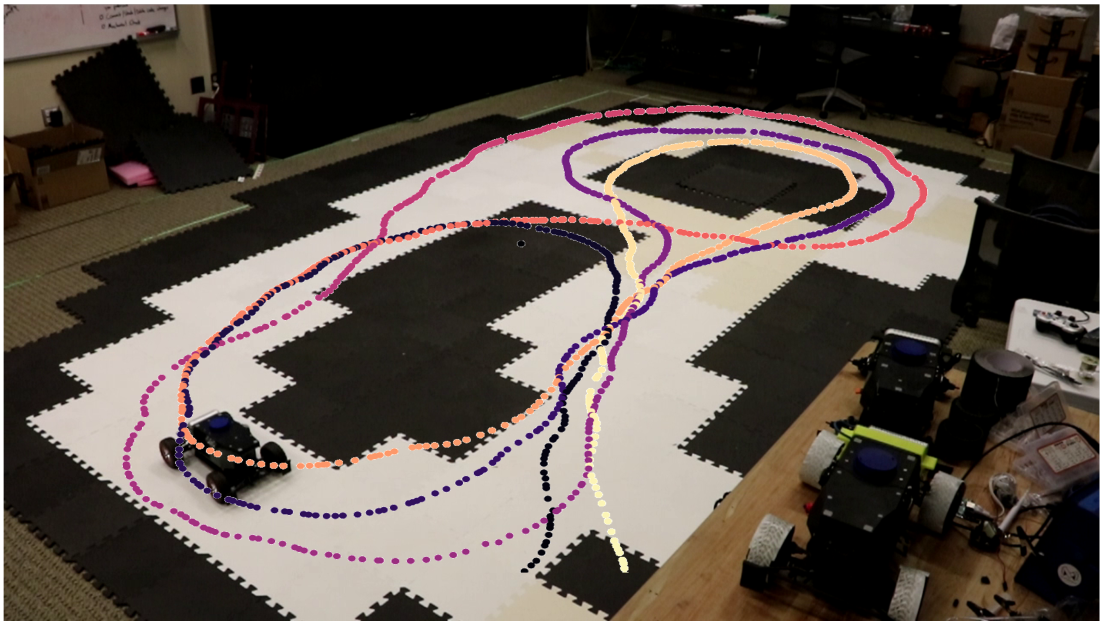

Fig. 8: **Captured trajectories of visual policy in real experiments**. The trajectories are obtained from the motion capture system. We include videos of each setting and trial in the supplementary materials.

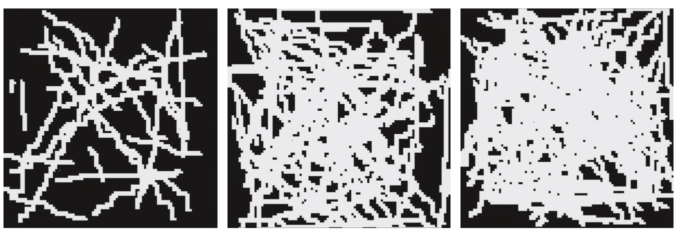


Fig. 9: **Examples of randomly generated environments to train $\pi_{\mathbf{vis}}$.** Black and white represent penalizing and safe regions, respectively. We adjust the number of walkers to train the policy across different difficulties. Using a smaller number of walkers will make the training environments more challenging.

such as traversability through this modality has its unique challenges [52]. First, visual observations (e.g., RGB images) from robots are represented in high-dimensional tensors, requiring exponentially more training data. Also, small changes in environments that result from illumination, weather, or time of day can result in catastrophic failure of machine learning models [12].

These challenges can be effectively addressed through the simulation of visual information. In addition to generating large amounts of task-relevant data, domain randomization can help improve generalization during transfer.

**Task.** We construct our task scene using black and white foam tiles (see Fig. 8). These tiles are inexpensive, safe, and widely available. Surrounded by black tiles, a “figure-8” path is laid using white tiles. The platform must remain on the white path while avoiding the surrounding black tiles. A dynamic version of this task is also used in evaluation; black “barricade” tiles are dynamically removed and placed to demonstrate robustness and real-time navigation reasoning.

We generate simulation environments with black and white areas, as we show in Fig. 9 to train the visual policy. Each example in the figure represents each sub-environment, and our entire map is composed of sub-environments. This allows us to assign different difficulties for each sub-environment. See Appendix B-C for additional details about map generation.

**Evaluation.** We measure success or failure over five trials. We define success if the robot finishes at least one lap without staying in one place for more than 5 seconds.

**Setup.** Our input states include grayscale observation (resolution of $40 \times 60$), linear and angular velocity, and last action. Linear and angular velocities are obtained from the motion capture system. We design experiments to demonstrate the effectiveness of image augmentations (i.e., color jittering and Gaussian blur) and different architectures (i.e., MLP vs CNN) in Sim2Real generalization.

**Results.** For quantitative evaluation, we run experiments five times and count the number of successes for each setting. As reported in Table III, image augmentation plays an essential role in Sim2Real generalization. The policies without augmentation fail at generalization. Further, despite the better learning capability of CNNs, they result in worse generalization to real; we also found that CNN training often collapses to a trivial solution, with the reward designs remaining the same as MLP trainings. Such results imply that CNNs may overfit to simulated data and require more careful design in rewards and loss functions for stable training.

In the supplementary materials, we qualitatively evaluate the results using recorded videos. Although both policies with augmentation predict meaningful actions, the MLP-based policy generates more smooth and precise actions, while the CNN-based policy often generates noisy actions, as shown in fast steering changes and audible motor cogging.

## V. Limitations

Although an objective of Wheeled Lab is to support education, this work has not yet conducted a study to evaluate how comprehensible the pipeline is to robotics students. Like F1Tenth, we aim to construct an open curriculum around Wheeled Lab to help introduce difficult or interdisciplinary topics that may be sources of confusion. This study will help reveal what steps in the Sim2Real process might prove most challenging or unfamiliar.

While iterative testing for unique conditions can serve pedagogical purposes, the appeal of building, training, and deploying a platform that "just works" can be inspirational for beginner roboticists. However, the policies presented here in their current open-source state must be met with patience. Thus, custom low-cost wheeled platforms invite yet another frontier of modern robot learning: adaptation [7]. For instance, the development of a drifting pipeline that adequately adapts to friction on deployment can help lower the barrier-to-entry even further.

## VI. Conclusion

In modern robotics, state-of-the-art methods often focus on complex, costly, and proprietary platforms and ecosystems, leaving behind more resource-constrained audiences. Wheeled platforms, on the other hand, already have deep roots in the broader community of educators, students, and enthusiasts, often acting as the entry-level platform of choice for beginning roboticists.

We present this work, Wheeled Lab, to act as a bridge between the broader community and state-of-the-art research in robot learning through low-cost, open-source wheeled platforms. Atop Isaac Lab, Wheeled Lab provides a modular and self-contained design framework to lower the barrier-to-entry while being fully integrated with Isaac Lab's extensive research community and ecosystem.

To kickstart this integration, we demonstrate the entire Sim2Real pipeline on three state-of-the-art policy types. Each policy and task is motivated by their fundamental relevance to general robot learning and Sim2Real. *Drifting* is a complex control problem that highlights accurate, randomized, and massive physical simulation. Through Wheeled Lab, we demonstrate the first zero-shot transfer of a drifting policy thus far in literature. *Elevation traversal* defines safe behavior at the interface between perception and control, highlighting physical scene generation and massive interaction. *Visual navigation* integrates a sensor modality with substantial research attention and pre-training potential while showcasing challenges in data diversity and transferability.

Wheeled Lab is entirely low-cost and open-source. Please refer to our websitefor experiment videos, code, and more information.

## Acknowledgments

TH is supported by the NSF GRFP under Grant No. DGE 2140004.

## References

[1] Crossing the Sim2Real Gap With NVIDIA Isaac Lab, . URL https://agilityrobotics.com/content/crossing-sim2real-gap-with-isaaclab.

[2] Get Started with Reinforcement Learning for Spot, . URL https://support.bostondynamics.com/s/article/Get-Started-with-Reinforcement-Learning-for-Spot-49966.

[3] MenteeBot, . URL https://menteebot.com/blog/#shopping-companion-2024.

[4] NVIDIA Isaac Lab Blog, . URL https://fieldai.com/news/field-ai-nvidia-partnership.

[5] Manuel Acosta, Stratis Kanarachos, and Michael E. Fitzpatrick. A Hybrid Hierarchical Rally Driver Model for Autonomous Vehicle Agile Maneuvering on Loose Surfaces:. In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics*, pages 216–225, Madrid, Spain, 2017. SCITEPRESS - Science and Technology Publications. ISBN 978-989-758-263-9 978-989-758-264-6. doi: 10.5220/0006393002160225. URL http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006393002160225.

[6] Bharathan Balaji, Sunil Mallya, Sahika Genc, Saurabh Gupta, Leo Dirac, Vineet Khare, Gourav Roy, Tao Sun, Yunzhe Tao, Brian Townsend, Eddie Calleja, Sunil Muralidhara, and Dhanasekar Karuppasamy. DeepRacer: Autonomous Racing Platform for Experimentation with Sim2Real Reinforcement Learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2746–2754, May 2020. doi: 10.1109/ICRA40945.2020.9197465. URL https://ieeexplore.ieee.org/abstract/document/9197465?casa_token=P1_UodV4wbEAAAAA:6EV0TBoDKvtpQailXF-0QMLLebuYMajCFlRjbSzrKQB__Q2khTqumEjWZKAM4C3Xrg7WgicvRg. ISSN: 2577-087X.

[7] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250, May 2018. doi: 10.1109/ICRA.2018.8460875. URL https://ieeexplore.ieee.org/document/8460875/?arnumber=8460875. ISSN: 2577-087X.

[8] Axel Brunnbauer, Luigi Berducci, Andreas Brandstätter, Mathias Lechner, Ramin Hasani, Daniela Rus, and Radu Grosu. Latent Imagination Facilitates Zero-Shot Transfer in Autonomous Racing, February 2022. URL http://arxiv.org/abs/2103.04909. arXiv:2103.04909 [cs].

[9] Peide Cai, Xiaodong Mei, Lei Tai, Yuxiang Sun, and Ming Liu. High-Speed Autonomous Drifting With Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, 5(2):1247–1254, April 2020. ISSN 2377-3766. doi: 10.1109/LRA.2020.2967299. URL https://ieeexplore.ieee.org/document/8961997/?arnumber=8961997. Conference Name: IEEE Robotics and Automation Letters.

[10] Xiaoyi Cai, James Queeney, Tong Xu, Aniket Datar, Chenhui Pan, Max Miller, Ashton Flather, Philip R. Osteen, Nicholas Roy, Xuesu Xiao, and Jonathan P. How. PIETRA: Physics-Informed Evidential Learning for Traversing Out-of-Distribution Terrain. *IEEE Robotics and Automation Letters*, pages 1–8, 2025. ISSN 2377-3766. doi: 10.1109/LRA.2025.3527285. URL https://ieeexplore.ieee.org/document/10833878/?arnumber=10833878. Conference Name: IEEE Robotics and Automation Letters.

[11] Guoying Chen, Xuanming Zhao, Zhenhai Gao, and Min Hua. Dynamic Drifting Control for General Path Tracking of Autonomous Vehicles. *IEEE Transactions on Intelligent Vehicles*, 8(3):2527–2537, March 2023. ISSN 2379-8904. doi: 10.1109/TIV.2023.3235007. URL https://ieeexplore.ieee.org/document/10011537/?arnumber=10011537. Conference Name: IEEE Transactions on Intelligent Vehicles.

[12] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T. Kim, Seungryong Kim, and Jaegul Choo. RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening. pages 11580–11590, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Choi_RobustNet_Improving_Domain_Generalization_in_Urban-Scene_Segmentation_via_Instance_Selective_CVPR_2021_paper.html.

[13] Mark Cutler and Jonathan P. How. Autonomous drifting using simulation-aided reinforcement learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5442–5448, Stockholm, Sweden, May 2016. IEEE. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487756. URL http://ieeexplore.ieee.org/document/7487756/.

[14] Aniket Datar, Chenhui Pan, Mohammad Nazeri, and Xuesu Xiao. Toward Wheeled Mobility on Vertically Challenging Terrain: Platforms, Datasets, and Algorithms. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16322–16329, May 2024. doi: 10.1109/ICRA57147.2024.10610079. URL https://ieeexplore.ieee.org/document/10610079.

[15] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3161–3171, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.00323. URL https://ieeexplore.ieee.org/document/9157346/.

[16] Franck Djeumou, Michael Thompson, Makoto Suminaka, and John Subosits. Reference-Free Formula Drift with Reinforcement Learning: From Driving Data to Tire Energy-Inspired, Real-World Policies, October 2024. URL http://arxiv.org/abs/2410.20990. arXiv:2410.20990 [cs].

[17] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16. PMLR, October 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html. ISSN: 2640-3498.

[18] Benjamin David Evans, Hendrik Willem Jordaan, and Herman Arnold Engelbrecht. Comparing deep reinforcement learning architectures for autonomous racing. *Machine Learning with Applications*, 14:100496, December 2023. ISSN 26668270. doi: 10.1016/j.mlwa.2023.100496. URL https://linkinghub.elsevier.com/retrieve/pii/S266682702300049X.

[19] Benjamin David Evans, Raphael Trumpp, Marco Caccamo, Felix Jahncke, Johannes Betz, Hendrik Willem Jordaan, and Herman Arnold Engelbrecht. Unifying F1TENTH Autonomous Racing: Survey, Methods and Benchmarks, April 2024. URL http://arxiv.org/abs/2402.18558. arXiv:2402.18558 [cs].

[20] Jonas Frey, Manthan Patel, Deegan Atha, Julian Nubert, David Fan, Ali Agha, Curtis Padgett, Patrick Spieler, Marco Hutter, and Shehryar Khattak. RoadRunner – Learning Traversability Estimation for Autonomous Off-road Driving, August 2024. URL http://arxiv.org/abs/2402.19341. arXiv:2402.19341 [cs].

[21] Jason Gibson, Bogdan Vlahov, David Fan, Patrick Spieler, Daniel Pastor, Ali-akbar Agha-mohammadi, and Evangelos A. Theodorou. A Multi-step Dynamics Modeling Framework For Autonomous Driving In Multiple Environments, May 2023. URL http://arxiv.org/abs/2305.02241. arXiv:2305.02241 [cs].

[22] J. Gonzales, F. Zhang, K. Li, and F. Borrelli. Autonomous drifting with onboard sensors. In *Advanced Vehicle Control*. CRC Press, 2016. ISBN 978-1-315-26528-5. Num Pages: 6.

[23] Nathaniel Hamilton, Patrick Musau, Diego Manzanas Lopez, and Taylor T. Johnson. Zero-Shot Policy Transfer in Autonomous Racing: Reinforcement Learning vs Imitation Learning. In *2022 IEEE International Conference on Assured Autonomy (ICAA)*, pages 11–20, March 2022. doi: 10.1109/ICAA52185.2022.00011. URL https://ieeexplore.ieee.org/document/9763640/?arnumber=9763640.

[24] Tyler Han, Alex Liu, Anqi Li, Alex Spitzer, Guanya Shi, and Byron Boots. Model Predictive Control for Aggressive Driving Over Uneven Terrain, June 2024. URL http://arxiv.org/abs/2311.12284. arXiv:2311.12284

[cs].
[25] Tyler Han, Sidharth Talia, Rohan Panicker, Preet Shah, Neel Jawale, and Byron Boots. Dynamics Models in the Aggressive Off-Road Driving Regime, May 2024. URL http://arxiv.org/abs/2405.16487. arXiv:2405.16487 [cs].
[26] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. ANYmal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88): eadi7566, March 2024. ISSN 2470-9476. doi: 10.1126/scirobotics.adi7566. URL https://www.science.org/doi/10.1126/scirobotics.adi7566.
[27] Ernő Horváth, Gergő Ignéczi, Norbert Markó, Rudolf Krecht, and Miklós Unger. Teaching Aspects of ROS 2 and Autonomous Vehicles. *Engineering Proceedings*, 79(1):49, 2024. ISSN 2673-4591. doi: 10.3390/engproc2024079049. URL https://www.mdpi.com/2673-4591/79/1/49. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
[28] Kevin Huang, Rwik Rana, Alexander Spitzer, Guanya Shi, and Byron Boots. DATT: Deep Adaptive Trajectory Tracking for Quadrotor Control, December 2023. URL http://arxiv.org/abs/2310.09053. arXiv:2310.09053 [cs].
[29] Sanghun Jung, JoonHo Lee, Xiangyun Meng, Byron Boots, and Alexander Lambert. V-STRONG: Visual Self-Supervised Traversability Learning for Off-road Navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1766–1773, May 2024. doi: 10.1109/ICRA57147.2024.10611227. URL https://ieeexplore.ieee.org/abstract/document/10611227.
[30] Katie Kang, Suneel Belkhale, Gregory Kahn, Pieter Abbeel, and Sergey Levine. Generalization through Simulation: Integrating Simulated and Real Data into Deep Reinforcement Learning for Vision-Based Autonomous Flight. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6008–6014, May 2019. doi: 10.1109/ICRA.2019.8793735. URL https://ieeexplore.ieee.org/document/8793735/?arnumber=8793735. ISSN: 2577-087X.
[31] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06419-4. URL https://www.nature.com/articles/s41586-023-06419-4. Publisher: Nature Publishing Group.
[32] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5 (47):eabc5986, October 2020. doi: 10.1126/scirobotics.abc5986. URL https://www.science.org/doi/10.1126/scirobotics.abc5986. Publisher: American Association for the Advancement of Science.
[33] Qiayuan Liao, Bike Zhang, Xuanyu Huang, Xiaoyu Huang, Zhongyu Li, and Koushil Sreenath. Berkeley Humanoid: A Research Platform for Learning-based Control, July 2024. URL http://arxiv.org/abs/2407.21781. arXiv:2407.21781 [cs].
[34] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-Level Reward Design via Coding Large Language Models, April 2024. URL http://arxiv.org/abs/2310.12931. arXiv:2310.12931 [cs].
[35] 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, January 2022. doi: 10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822. Publisher: American Association for the Advancement of Science.
[36] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, June 2023. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2023.3270034. URL http://arxiv.org/abs/2301.04195. arXiv:2301.04195 [cs].
[37] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving Rubik's Cube with a Robot Hand, October 2019. URL http://arxiv.org/abs/1910.07113. arXiv:1910.07113 [cs].
[38] Matthew O'Kelly, Hongrui Zheng, Dhruv Karthik, and Rahul Mangharam. F1TENTH: An Open-source Evaluation Environment for Continuous Control and Reinforcement Learning. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, pages 77–89. PMLR, August 2020. URL https://proceedings.mlr.press/v123/o-kelly20a.html. ISSN: 2640-3498.
[39] Sahar Salimpour, Jorge Peña-Queralta, Diego Paez-Granados, Jukka Heikkonen, and Tomi Westerlund. Sim-to-Real Transfer for Mobile Robots with Reinforcement Learning: from NVIDIA Isaac Sim to Gazebo and Real ROS 2 Robots, January 2025. URL http://arxiv.org/abs/2501.02902. arXiv:2501.02902 [cs].
[40] Chinmay Samak, Tanmay Samak, and Venkat Krovi. Towards Sim2Real Transfer of Autonomy Algorithms using AutoDRIVE Ecosystem. *IFAC-PapersOnLine*, 56(3): 277–282, January 2023. ISSN 2405-8963. doi: 10.1016/j.ifacol.2023.12.037. URL https://www.sciencedirect.com/science/article/pii/S2405896323023704.
[41] Tanmay Samak, Chinmay Samak, Sivanathan Kandhasamy, Venkat Krovi, and Ming Xie. AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin

Ecosystem for Autonomous Driving Research & Education. *Robotics*, 12(3):77, June 2023. ISSN 2218-6581. doi: 10.3390/robotics12030077. URL https://www.mdpi.com/2218-6581/12/3/77. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347.

[43] Siddhartha S. Srinivasa, Patrick Lancaster, Johan Michalove, Matt Schmittle, Colin Summers, Matthew Rockett, Rosario Scalise, Joshua R. Smith, Sanjiban Choudhury, Christoforos Mavrogiannis, and Fereshteh Sadeghi. MuSHR: A Low-Cost, Open-Source Robotic Racecar for Education and Research, December 2023. URL http://arxiv.org/abs/1908.08031. arXiv:1908.08031 [cs].

[44] Kyle Stachowicz, Dhruv Shah, Arjun Bhorkar, Ilya Kostrikov, and Sergey Levine. FastRLAP: A System for Learning High-Speed Driving via Deep RL and Autonomous Practicing. In *Proceedings of The 7th Conference on Robot Learning*, pages 3100–3111. PMLR, December 2023. URL https://proceedings.mlr.press/v229/stachowicz23a.html. ISSN: 2640-3498.

[45] Sidharth Talia, Matt Schmittle, Alexander Lambert, Alexander Spitzer, Christoforos Mavrogiannis, and Siddhartha S. Srinivasa. Demonstrating HOUND: A Low-cost Research Platform for High-speed Off-road Underactuated Nonholonomic Driving, July 2024. URL http://arxiv.org/abs/2311.11199. arXiv:2311.11199 [cs].

[46] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: GPU Parallelized Robotics Simulation and Rendering for Generalizable Embodied AI, October 2024. URL http://arxiv.org/abs/2410.00425. arXiv:2410.00425 [cs].

[47] 1X Technologies. 1X Technologies | Safe, Intelligent Humanoids. URL https://www.1x.tech/.

[48] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, May 2017. doi: 10.1109/ICRA.2017.7989202. URL https://ieeexplore.ieee.org/document/7989202/?arnumber=7989202.

[49] Wenli Xiao, Haoru Xue, Tony Tao, Dvij Kalaria, John M. Dolan, and Guanya Shi. AnyCar to Anywhere: Learning Universal Dynamics Model for Agile and Adaptive Mobility, September 2024. URL http://arxiv.org/abs/2409.15783. arXiv:2409.15783 [cs].

[50] Tong Xu, Chenhui Pan, and Xuesu Xiao. Reinforcement Learning for Wheeled Mobility on Vertically Challenging Terrain. In *2024 IEEE International Symposium on Safety Security Rescue Robotics (SSRR)*, pages 125–130, November 2024. doi: 10.1109/SSRR62954.2024.10770034. URL https://ieeexplore.ieee.org/document/10770034/?arnumber=10770034. ISSN: 2475-8426.

[51] Yuxiang Yang, Guanya Shi, Changyi Lin, Xiangyun Meng, Rosario Scalise, Mateo Guaman Castro, Wenhao Yu, Tingnan Zhang, Ding Zhao, Jie Tan, and Byron Boots. Agile Continuous Jumping in Discontinuous Terrains, September 2024. URL http://arxiv.org/abs/2409.10923. arXiv:2409.10923 [cs].

[52] Yuri D. V. Yasuda, Luiz Eduardo G. Martins, and Fabio A. M. Cappabianco. Autonomous Visual Navigation for Mobile Robots: A Systematic Literature Review. *ACM Computing Surveys*, 53(1):1–34, January 2021. ISSN 0360-0300, 1557-7341. doi: 10.1145/3368961. URL https://dl.acm.org/doi/10.1145/3368961.

[53] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to Manipulate Anywhere: A Visual Generalizable Framework For Reinforcement Learning, October 2024. URL http://arxiv.org/abs/2407.15815. arXiv:2407.15815 [cs].

## Appendix A
## Supplementary Material

Code can be found on the project Github repository:

`redacted for review`

Videos and more information can be found on the project website:

`redacted for review`

## Appendix B
## Design and Implementation Details

The typical ranges on friction, mechanical tolerances, damages, etc. for low-cost platforms are far from ideal. However, iterating and developing on individual platforms in unique environments is practically and pedagogically critical, especially for real-world robotics. We hope that this work allows practitioners to focus their efforts on precisely this development cycle rather than infrastructure. The following implementation details should be regarded as a starting point and potential reference for troubleshooting inevitable issues.

We recommend interested readers view the configuration files for each task found in GitHub (`redacted for review`) for a detailed overview. Isaac Lab environment configurations primarily include parameter settings and reward functions and can be quickly parsed while efficiently communicating environment behavior. We highlight some notable details in this section.

### A. Drifting

We found that drifting was an exemplary task to demonstrate the physical Sim2Real gap. Much of the development cycle was focused on closing this gap, as supported by countless other seminal works [37, 26].

*1) Actuators:* In many related works for agile tasks, data is typically collected to assess actuator response and estimate gains. However, this process can be unfamiliar to beginning roboticists or non-traditional enthusiasts. *Instead, all actuator parameters are extracted exactly from their technical specification sheet online, and gains are domain-randomized across environments.* Achieving the task in this manner helps to convey that, while domain expertise can certainly improve performance (domain randomized policies are known to find conservative solutions [33]), it is not strictly necessary through modern methods.

It is worth noting that a few "Sim2Real2Sim" (or Sim2Sim) cycles were spent narrowing the randomized gain ranges for the throttle actuator settings. Due to neural network policy tendencies to exhibit "bang-bang" (on-off) behavior, gain values can become extremely important to the actual forces exerted on the vehicle.

Establishing and improving pipelines for this process can be an accessible yet effective future contribution for use by potential practitioners.

*2) Friction:* Initially, the domain randomization over friction was used to cover a wide range (0.2 to 0.8) of potential friction values between the tape-covered tires and the carpeted ground. However, resulting policies would perform poorly and inconsistently on deployment. A long thread of Sim2Sim cycles was used to search for friction parameters but still resulting policies were not reliably transferring.

In the end, a cheap (less than 5 USD) spring scale was used to estimate the coefficient of friction, measured by dragging the vehicle with the scale along the carpet. This value (about 0.4) was then used as the midpoint of the friction randomization during training.

The project website also contains videos of deployments on different surfaces (e.g. finished concrete). Lower friction values appear to cause the vehicle to take wider turns (due to lack of turning friction) and spin out quickly during drift maneuvers. Enabling the platform to drift over multiple surfaces is also of fundamental interest to the robotics community as it is a form of domain adaptation.

*3) Articulations:* While we initially used the MuSHR [43] articulations defined by the open-source URDF file, we found that its modeling inaccuracies affected the precise control expectations of the platform. As a result, we improved the articulations: (1) by moving the steering linkages from the wheel axis to the actual steering joint and (2) by articulating the suspension joints of the vehicle.

Steering inaccuracies were noticeable due to the gap between turning radii in real and sim. The moment arm on the linkage can also adversely affect the tracking of the steering joint during high-velocity maneuvers. This can even be observed in simulation when steering joints behave strangely asymmetrically with unsuitable actuator settings.

Suspension helps close the gap on load shifting during turns. With a suspension travel of 4 cm, these shifts have consequences in the total friction and "thrust" produced while throttling through turns. Interestingly, suspension articulation also helped to stabilize the simulation. At a wheel speed of 3 m/s, the rotation rate of the wheels reaches 60 rad/s, occasionally resulting in collision penetrations during simulation which cause unrealistic "jumping" through drifted turns. Damped suspension joints helped to stabilize these collisions without requiring smaller simulation time steps which would significantly lengthen training times.

*4) Rewards:* Six rewards were designed for this task:

i.) **Cross-Track Distance**. The lateral distance of the vehicle from an oval track line. This is defined without time-dependent reference trajectories as often done in model-based approaches or even in deep RL methods [9]. This is highly weighted otherwise the agent tends to take larger, safer turns.

ii.) **Velocity**. Reward for high speeds.

iii.) **Side-Slip**. Only stable side-slip angles rewarded. Stable side-slip angles are manageable through the steering limits (0-30°).

iv.) **Progress**. Reward for making progress along an arbitrary direction (counter-clockwise, in this work).

v.) **Turn-Energy**. Shaping term for increasing velocity rewards specifically when inside turn regions.

vi.) **Turn-Left-Go-Right**. Shaping term for encouraging exploration to discover an alternate turning mode (e.g. drifting) by counter-steering. Strictly positive when angular velocity and steering command are opposite to each other. 0 otherwise.

Penalties were also given for going out-of-bounds.

### B. Elevation

*1) Training:* Training elevation in simulation proved to be a challenging task. Training would often collapse into one of: (1) seeking goals while avoiding obstacles or (2) climbing ramps. Striking a balance between the two often required re-designing the scene or task configuration in simulation.

As observed for model-based methods [25], evaluating uneven terrain traversal for vehicles can be challenging due to the reasonable alternate mode of avoiding risky terrain altogether, which may often be faster. However, the primary goal of this work is to better enable the broader community to engage with state-of-the-art methods in elevation-based robotics. Thus, the reward and scene were constructed to incentivize the agent to actively seek out risky terrain while penalizing failures.

Supported by machine learning literature, noise injection becomes less effective for generalization at higher dimensions. For states such as elevation maps, perceived noise is likely more structured than the additive Gaussians injected through observation corruption. While debugging, we found that the policy would improve if the elevation features were brought closer in shape to the features seen in the simulation. To resolve this without "training on test", the scene was augmented with slopes of various gradations and heights.

*2) Suspension:* As mentioned in Appendix B-A, suspension joints were added to the open-source MuSHR articulations. This is critical for elevation where maintaining momentum up a ramp helps training stability. Without suspension articulation, climbing ramps have an abrupt and destabilizing impact at higher speeds.

*3) Rewards:*

i.) **Goal Velocity**. Rewards velocity projected onto goal heading vector.

ii.) **Z-Position**. Rewards larger z-positions (gained through ascending ramps).

iii.) **Falling Penalty**. Penalizes negative body-frame z velocities.

iv.) **Roll on elevation**. Penalizes roll angles while on elevation features.

### C. Visual

While visual information is essential in modern robotics to solve complex real-world problems, simulating such information requires state-of-the-art rendering techniques to close the Sim2Real gap. Despite such computationally heavy efforts, machine learning models trained with simulated data often fail at real-world generalization due to diverse types of environmental changes, e.g., illuminations, weather, etc. Instead, in this work, we "simplify" visual information from both simulation and real by compressing RGB pixels to grayscale, closing the Sim2Real gap. Still, we observed that such compressed visual information exhibits a non-trivial amount of Sim2Real gap. This section describes how we tackle this problem by providing details of our environment, training augmentations, architectures, and reward designs.

*1) Visual map generation:* We designed the problem by encoding traversability into color information. Black tiles are not-traversable regions while white indicates safe to travel. We divided the whole map into multiple sub-environments that are individually configurable. Then, we randomized white paths over the black area in each sub-environment. Specifically, each sub-environment comprises 9 cells, and we sample a start point in each cell and generate paths toward random endpoints in the sub-environment. We made the number of random paths from a single start point configurable, allowing the users to adjust the difficulty of the training (see Fig. 9). This can later be used for curriculum learning by assigning different difficulties for sub-environments. We tested the generated map with more than 2,000 agents in parallel running seamlessly with a single GPU. The algorithm overview for sub-environment generation is presented in Alg. 1.

**Algorithm 1** Visual Policy Sub-environment Generation

**Input:**
$(H_{\text{env}}, W_{\text{env}}) \leftarrow$ # of rows and # of columns for each sub environment
$(H_{\text{cell}}, W_{\text{cell}}) \leftarrow$ # of rows and # of columns for each cell
$N_{\text{walkers}} \leftarrow$ # of walkers for path generation

**Pseudo-code:**
▷ Initialize traversability map; 0 - not traversable / 1 - traversable
$\mathbf{T} \in \{0,1\}^{H_{\text{env}} \times W_{\text{env}}} \leftarrow \mathbf{0}$

▷ Sample start points from each cell
$\mathbf{P}^{\text{start}} \leftarrow [\,]$ ▷ Initialize an array to save start points
▷ Sample random start points for each cell
**for** row $r = 0, 1, \ldots, H_{\text{env}}/H_{\text{cell}}$ **do**
  **for** col $c = 0, 1, \ldots, W_{\text{env}}/W_{\text{cell}}$ **do**
    $p_{\text{row}} \leftarrow \text{UniformSampling}(0, H_{\text{cell}}) + r \cdot H_{\text{cell}}$
    $p_{\text{col}} \leftarrow \text{UniformSampling}(0, W_{\text{cell}}) + c \cdot W_{\text{cell}}$
    $\mathbf{P}^{\text{start}} \mathrel{+}= [(p_{\text{row}}, p_{\text{col}})]$ ▷ Append the start point to $\mathbf{P}^{\text{start}}$
  **end for**
**end for**

**for** $(p_{\text{row}}, p_{\text{col}})$ in $\mathbf{P}^{\text{start}}$ **do**
  **for** iter$= 0, 1, \ldots, N_{\text{walkers}}$ **do**
    ▷ Sample random end points
    **Do**
      $q_{\text{row}} \leftarrow \text{UniformSampling}(0, H_{\text{env}})$
      $q_{\text{col}} \leftarrow \text{UniformSampling}(0, W_{\text{env}})$
    **do While** $\mathbf{T}(q_{\text{row}}, q_{\text{col}}) == 1$
    ▷ Generate random paths and update traversability map
    GenerateRandomPaths($\mathbf{T}$, $(p_{\text{row}}, p_{\text{col}})$, $(q_{\text{row}}, q_{\text{col}})$)
  **end for**
**end for**
**Return:** $\mathbf{T}$

*2) Model Architectures:* We adopt multi-layer perceptron (MLP) as our basic policy network. For ablations, we add a simple 3-layer convolutional neural network (CNN) that serves as an image feature extractor before the policy networks. The extracted features are concatenated to the rest of the observation terms such as linear/angular velocities and last action and fed into the MLP policy networks. We observe that inducing CNN takes longer training time to learn optimal actions and often collapses into a trivial solution - agents decide to spin in a circle, only maximizing the velocity rewards.

*3) Image Augmentations:* To close the sim-to-real gap, we apply image augmentations for policy training. This includes aggressive color jittering (*i.e.,* brightness, contrast, saturation, and hue adjustments) and Gaussian blur. As demonstrated in both qualitative and quantitative evaluations, image augmentations play a crucial role in closing sim-to-real gaps. Further augmentations or randomization can be easily applied to our code design.

*4) Rewards:* Two rewards were designed for this task

i.) **Velocity Reward**. Rewards higher velocities

ii.) **Traversability Reward and Penalty**. Rewards being on white regions, and penalizes it for being on black regions

*5) Terminal Conditions:* We experimented with different terminations. In addition to the standard time-out and out-of-bounds terminal conditions, we added a terminal condition when the robot is on black. However, this often resulted in a lack of exploration, and policies that produced “jerky” or undesirable actions, such as idling at the initial location.